

Руководство пользователя

Motiware
Белгород

2015

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

ГЛАВА 1. Общие сведения

- 1.1. Описание настоящего Руководства
- 1.2. Назначение платформы Motiware Melody One
- 1.3. Преимущества использования платформы Motiware Melody One для организации
 - 1.3.1. Преимущества для руководителей
 - 1.3.2. Преимущества для сотрудников
- 1.4. Описание платформы Motiware Melody One
 - 1.4.1. Загрузка данных в Motiware Melody One
 - 1.4.2. Консолидация данных в Motiware Melody One
 - 1.4.3. Визуализация данных в Motiware Melody One
 - 1.4.4. Хранение и защита данных в Motiware Melody One
- 1.5. Возможности платформы Motiware Melody One
- 1.6. Сайт Motiware Melody One
- 1.7. Требования к пользовательскому программному и аппаратному обеспечению
 - 1.7.1. Программно - аппаратные требования к серверной части
 - 1.7.2. Программно-аппаратные требования к клиентской части
- 1.8. Необходимый уровень подготовки администратора системы
- 1.9. Сопровождение систем на платформе Motiware Melody One
 - 1.9.1. Разработка архитектуры информационной системы
 - 1.9.2. Проектирование и разработка функционала
 - 1.9.3. Обучение
 - 1.9.4. Консультирование
 - 1.9.5. Техническое сопровождение
- 1.10. Термины и определения (принятые обозначения)

ГЛАВА 2. Архитектура платформы Motiware Melody One

- 2.1. Прикладная архитектура платформы Motiware Melody One
 - 2.1.1. Трехуровневая архитектура платформы Motiware Melody One
 - 2.1.2. Функциональная схема платформы Motiware Melody One
 - 2.1.3. Структурная схема платформы Motiware Melody One
- 2.2. Сетевая архитектура платформы Motiware Melody One
- 2.3. Архитектура данных
- 2.4. Принципы разработки архитектурных решений
- 2.5. Взаимодействие с бизнес-архитектурой

ГЛАВА 3. Интерфейс пользователя платформы Motiware Melody One

- 3.1. Общие сведения о пользовательском интерфейсе
- 3.2. Главное меню
 - 3.2.1. Структура главного меню

- [3.2.2. Пункты и подпункты](#)
 - [3.3. Системные пункты главного меню](#)
 - [3.3.1. Пункт "Настройка"](#)
 - [3.3.1.1. Подпункт "Мои реквизиты"](#)
 - [3.3.1.2. Подпункт "Группы пользователей"](#)
 - [3.3.2. Пункт "Администрирование"](#)
 - [3.3.2.1. Подпункт "Подразделения и пользователи"](#)
 - [3.3.2.2. Подпункт "Роли"](#)
 - [3.3.2.3. Подпункт "Экспорт"](#)
 - [3.3.2.4. Подпункт "Типы данных"](#)
 - [3.3.2.5. Подпункт "Статистика"](#)
 - [3.3.2.6. Подпункт "Кэш"](#)
 - [3.3.2.7. Подпункт "Пользователи"](#)
 - [3.3.2.8. Подпункт "Меню"](#)
 - [3.3.2.9. Подпункт "Периоды рабочего времени"](#)
 - [3.3.2.10. Подпункт "Календарь рабочего времени"](#)
 - [3.3.2.11. Подпункт "Действия пользователей"](#)
 - [3.3.2.12. Подпункт "Импорт таблицы из Excel"](#)
 - [3.3.2.13. Подпункт "Шаблоны"](#)
 - [3.3.2.14. Подпункт "Планировщик"](#)
 - [3.3.3. "Пункт "Подразделения, должности и пользователи"](#)
 - [3.3.4. Пункт "Процессы"](#)
 - [3.3.4.1. Подпункт "JBPM"](#)
 - [3.3.4.2. Подпункт "ETL"](#)
 - [3.3.5. Пункт "Отчеты"](#)
 - [3.3.6. Кнопка "Выход"](#)
 - [3.4. Пункты пользователей](#)
 - [3.5. Рабочая область](#)
 - [3.5.1. Кнопки для вызова системных операций](#)
 - [3.5.2. Панель навигации](#)
 - [3.5.3. Кнопки для настройки фильтров](#)
- [ГЛАВА 4. Информационные объекты платформы Motiware Melody One](#)
- [4.1. Типы данных](#)
 - [4.1.1. Системные типы данных](#)
 - [4.1.2. Пользовательские типы данных](#)
 - [4.1.3. Свойства типов данных](#)
 - [4.2. Типы полей](#)
 - [4.2.1. Простые типы полей](#)
 - [4.2.2. Сложные \(составные\) типы полей](#)
 - [4.2.3. Атрибуты полей](#)
 - [4.3. Метки и Ячейки](#)
 - [4.4. Таблицы \(экземпляры типов данных\)](#)
 - [4.5. Поставщики данных](#)

[4.6. Операции над типами данных](#)

[4.6.1. Системные операции](#)

[4.6.2. Пользовательские операции](#)

[4.6.3 Триггеры](#)

[4.7. Создание и настройка типов данных](#)

[4.7.1. Создание типа данных.](#)

[4.7.2. Настройка типа данных](#)

[4.7.2.1. Вкладка "Свойства"](#)

[4.7.2.2. Вкладка "Поля"](#)

[4.7.2.3. Вкладка "Метки"](#)

[4.7.2.4. Вкладка "Ячейки"](#)

[4.7.2.5. Вкладка "Экземпляры"](#)

[4.7.2.6. Вкладка "Дополнительно"](#)

[4.7.2.7. Вкладка "Ссылки"](#)

[4.8. Общие принципы создания моделей типов данных](#)

[4.9. Использование диаграмм классов UML](#)

[ПРИЛОЖЕНИЯ](#)

[Приложение 1. Кнопки для вызова системных операций](#)

[Приложение 2. Системные типы данных](#)

[Приложение 3. Типы триггеров](#)

[Приложение 4. Описание функций](#)

[Приложение 5. Виды таблиц](#)

[ГЛОССАРИЙ](#)

ВВЕДЕНИЕ

Настоящее руководство включает в себя описание принципов настройки информационных систем, созданных на базе платформы Motiware Melody One, и предназначено для пользователей, проводящих обслуживание и поддержку работоспособности систем, созданных на базе платформы Motiware Melody One.

Данное руководство содержит общие сведения о платформе Motiware Melody One, описание ее архитектуры, интерфейса пользователя, информационных объектов, основных модулей и настроек платформы Motiware Melody One.

В настоящем руководстве приведены описания последовательности действий пользователя системы при выполнении основных типовых операций технологического процесса настройки, назначения прав доступа пользователям, поддержания работоспособности систем, обеспечения сохранности информации в процессе работы.

ГЛАВА 1. Общие сведения

1.1. Описание настоящего Руководства

ГЛАВА 1 включает в себя общие сведения о платформе Motiware Melody One, назначение и преимущества использования Motiware Melody One в компаниях и организациях, требования к программному и аппаратному обеспечению, к уровню подготовки администратора системы.

В ГЛАВЕ 2 описана архитектура платформы Motiware Melody One, рассмотрены функциональная и структурная схемы, архитектурные решения и принципы их разработки.

ГЛАВА 3. содержит информацию об интерфейсе пользователя платформы Motiware Melody One. Здесь подробно рассмотрено главное меню, его пункты и подпункты, организация рабочей области, основные функции и расположение панелей с кнопками для вызова системных операций.

ГЛАВА 4. знакомит с описаниями основных информационных объектов платформы Motiware Melody One, прежде всего с типами данных и их свойствами. В этой главе размещена информация о создании и настройке типа данных.

ГЛОССАРИЙ содержит краткие толкования основных терминов и понятий, используемых в настоящем руководстве.

В ПРИЛОЖЕНИЯХ размещена различная вспомогательная информация, которая может понадобится при решении задач практического характера.

1.2. Назначение платформы Motiware Melody One

Платформа Motiware Melody One – это гибкая, масштабируемая, надёжная и высокопроизводительная система, предназначенная для разработки и внедрения в короткие сроки корпоративных веб-приложений, которые ориентированны на использование веб-браузеров в качестве клиентских мест для организации доступа к базам данных во внутрикорпоративных сетях, в сети Интернет.

Создание законченного решения на платформе Motiware Melody One занимает немного времени. При этом первые результаты пользователь получает практически сразу, кроме того у него остается возможность легко наращивать мощность решения, добавляя новые объекты и новые связи.

Реализованные в Motiware Melody One технологии позволяют на базе единой архитектуры в сжатые сроки пройти все этапы построения законченной прикладной

системы – от создания хранилища данных и проектирования входных форм до визуализации аналитики на информационных панелях.

Платформа Motiware Melody One предназначена для разработки решений в организациях любой формы и сферы деятельности, которым необходима информационная поддержка как управленческих решений, так и производственных процессов. Используя возможности платформы Motiware Melody One, пользователь может создавать удобные приложения для эффективного решения бизнес-задач.

1.3. Преимущества использования платформы Motiware Melody One для организации

Применение платформы Motiware Melody One предоставляет организациям следующие преимущества (выгоды):

- легкое и быстрое создание приложений корпоративного уровня при невысокой стоимости владения;
- мгновенное внесение изменений в решения, исходя из потребностей бизнеса;
- интеграция в существующую инфраструктуру;
- удобный механизм сбора и обработки информации;
- надёжная система хранения данных и доступа к ним;
- наглядные способы представления данных и получения отчётности;
- быстрое внедрение решений.

1.3.1. Преимущества для руководителей

Используя платформу Motiware Melody One, руководители приобретают возможность:

- получать актуальные отчёты в режиме реального времени,
- проводить анализ информации для принятия управленческих решений,
- прогнозировать развитие ситуации,
- контролировать работу своих сотрудников в разных режимах,
- ставить задачи сотрудникам, которые находятся в различных местах.

Кроме того, в лице Motiware Melody One руководители обретают удобное средство для организации командной работы в своей компании.

1.3.2. Преимущества для сотрудников

Применяя системы, построенные на платформе Motiware Melody One в своей работе, сотрудники организации легко решают задачи:

- ввода большого количества данных, их проверки и сохранности,

- соблюдения сроков предоставления отчётности и выполненных работ,
- организации любых удалённых взаимодействий.

1.4. Описание платформы Motiware Melody One

Платформа Motiware Melody One включает средства, позволяющие максимально сокращать время разработки, быстро создавать новые прикладные решения и в короткие сроки адаптировать их в соответствии с изменяющимися требованиями предприятий. Возможности платформы обеспечивают не только высокую скорость первоначальной разработки продукта, но и его быструю адаптацию в дальнейшем.

Motiware Melody One - это многопользовательская среда, позволяющая одновременно работать, вносить и обрабатывать данные, получать отчётную информацию любому количеству пользователей; в качестве ограничений здесь могут выступать лишь пропускная способность канала связи и мощность серверного оборудования.

Основой платформы Motiware Melody One служит набор системных типов данных, которые могут использоваться в дальнейшей разработке как непосредственно, так и для создания новых (пользовательских) типов данных, применяемых для моделирования поведения сущностей предметной области. Под типом данных здесь и далее понимается набор полей, значений, триггеров, операций и экземпляров. Тип данных описывает поведение и представление записей с полями, триггерами, операциями, значениями и т.д.

Построена на технологии java servlet. Может запускаться на любом контейнере сервлетов. Поиск на базе lucene. ORM - hibernate, используется spring. Поддержка etl процессов. реализация bpmn 2.0 на платформе activiti. Клиентская часть построена на технологии gwt, связь с серверной частью построена на технологиях gwt rpc, jsonrpc, web service

Серверные операционные системы – любые, на которых работает контролер java servlet'ов: Windows, Linux, MacOS X, FreeBSD. Обеспечивается поддержка браузеров от разных производителей.

1.4.1. Загрузка данных в Motiware Melody One

Платформа Motiware Melody One, как и любое решение, построенное на ней, способно получать данные из различных источников - будь то непосредственный ввод, загрузка из других систем или приложений при помощи настроенного ETL - процесса, загрузка данных из электронной почты, выгрузка из XML файла и др. При непосредственном вводе данных в заранее настроенные формы в Motiware Melody One

можно организовать процесс автоматических вычислений, который может быть чрезвычайно полезен при обмене информацией между различными подразделениями организации (компании), ее филиалами.

Гибкие настройки прав доступа, реализованные в Motiware Melody One в виде системы ролей, помогают организовать групповую работу с данными, разграничить обязанности сотрудников, предоставив каждому из них доступ только к тем элементам вебинтерфейса и инструментам управления данными, которые ему необходимы.

Настраиваемый ETL процесс даёт возможность получать информацию из других баз данных и других приложений. Стандарт XML позволяет загружать данные различных систем, таких как 1С и др. Платформа Motiware Melody One позволяет настраивать автоматическую проверку вводимой информации, заполнение обязательных полей, проверку данных при сохранении с учётом правил валидации, заданных при настройке.

1.4.2. Консолидация данных в Motiware Melody One

Система способна моментально обработать данные, анализируя сотни тысяч показателей по запросу и автоматически сформировать отчёт нужного вида. Сводные отчёты могут содержать не только консолидированные данные по всей организации, но и результаты анализа определенных показателей ее деятельности.

1.4.3. Визуализация данных в Motiware Melody One

Платформа Motiware Melody One позволяет осуществлять мониторинг различных информационных событий, необходимых пользователю, обеспечивая тем самым эффективный анализ информации.

Периодичность и сроки ввода данных могут рассчитываться с учетом рабочего и нерабочего времени, выходных и праздничных дней. Используемая система индикации позволит не пропустить сроки предоставления отчетности, внесения данных, выполнения работ, завершения задач.

Мощная встроенная подсистема отчетов позволяет собирать информацию в различных разрезах и с любой периодичностью. Отображаться данные могут на информационных панелях в виде графиков, схем, диаграмм. Кроме того, предусмотрена выгрузка в файлы популярных форматов.

1.4.4. Хранение и защита данных в Motiware Melody One

Motiware Melody One умеет работать с широким спектром СУБД, таких как PostgreSQL, Oracle, MS SQL, Firebird, MySQL и других, имеющих JDBC-коннекторы, что,

в свою очередь, говорит об универсальности используемых подходов, надёжности хранения данных, масштабируемости. Хранящиеся в СУБД данные надёжно защищены от несанкционированного доступа за счёт использования устойчивых к взлому логинов и паролей, кроме того для обеспечения аутентичности и целостности данных можно использовать ЭЦП.

Доступ к хранящейся информации осуществляется исходя из существующей системы доступа, которая позволяет предоставить доступ к одной ячейке, одной строке, либо запретить такой доступ как к объекту целиком, так и к отдельным элементам типа данных.

1.5. Возможности платформы Motiware Melody One

Платформа Motiware Melody One обладает широкими возможностями, позволяющими пользователям решать задачи разного характера, а именно:

- в сжатые сроки разработать макет формы для сбора информации, используя для этого привычные инструменты – табличные редакторы;
- задать правила и ограничения на вводимые данные, сведя тем самым возможность ввода некорректных данных к минимуму;
- быстро, гарантированно и контролируемо доставить адресатам (поставщикам информации) электронные формы для заполнения;
- обеспечить сотрудников простым и действенным средством, позволяющим организовать ввод данных, исключив при этом случайные и преднамеренные ошибки;
- предоставить возможность наблюдения за ходом процесса ввода данных и заполнения форм;
- качественно отфильтровать информацию, поступающую на этапе ввода операторами или при импорте из различных информационных систем;
- запланировать периодичность сбора данных и довести эти планы до исполнителей в простой и понятной форме, включающей предупреждения о неактуальности информации, средствами электронной почты;
- получить наглядное визуальное представление собранной информации в виде понятных отчётных форм как табличного, так и графического вида с индикацией критических значений, позволяющих быстро и качественно выделять ключевые данные в большом потоке информации;
- автоматизировать процедуры проведения сравнительного анализа данных для быстрого и своевременного принятия управленческих решений;

- предоставить сотрудникам удобные средства для совместной и одновременной работы над сбором и обработкой данных, подготовкой отчетов, без проведения личных встреч и совещаний;
- обеспечить планирование и вести учет реальных затрат времени сотрудников по сбору и обработке информации, отправке необходимых донесений.

1.6. Сайт Motiware Melody One

Сайт с информацией о платформе Motiware Melody One расположен по ссылке: <http://www.motiw.ru/>

1.7. Требования к пользовательскому программному и аппаратному обеспечению

Минимальная конфигурация аппаратных (технических) и общесистемных программных средств должна соответствовать указанным далее параметрам.

1.7.1. Программно - аппаратные требования к серверной части

Для предотвращения потери данных и обеспечения надежной работы системы рекомендуется разделять установку на:

1. Основной Web-сервер.
2. Дополнительный Web-сервер (при значительном количестве пользователей (свыше 50) и/или объеме данных (размер базы данных и хранимых файлов свыше 20ГБ).
3. Систему хранения резервных копий данных.
4. Сервер баз данных (при значительном количестве пользователей (свыше 50) и/или объеме данных (размер базы данных и хранимых файлов свыше 5 Гб).

Рекомендуется ориентироваться на приведенные ниже требования к аппаратно-программному обеспечению.

Web-сервер (сервер приложений)

Аппаратные требования:

Объем БД, МБ	Количество пользователей	Рекомендуемые характеристики процессора	Минимальный объем ОЗУ, ГБ	Дисковая подсистема

500-1000	До 10	4-ядерный, 2.1 ГГц	8	Software RAID1 (SATA HDD)
1000-5000	До 20	4-ядерный, 3.6 ГГц	8	Software RAID1 (SATA HDD)
5000-10000	До 50	4-ядерный, 3,6 ГГц или 8-ядерный, 2,6 ГГц	16	Hardware RAID1 (SATA/SSD HDD)
5000-20000	До 100	8-ядерный, 3,6 ГГц	16	Hardware RAID1 (SAS HDD)
20000 и более	Более 100	16-ядерный, 3,6 ГГц	32	Hardware RAID10 (SAS 15к HDD)

Рекомендации:

- Использование аппаратного RAID1/10 SATA/SAS дисками согласно таблицы;
- Доступ сервера к сети Интернет, от 5 Mbps (рекомендуется 10 Mbps);
- Свободное дисковое пространство, объем которого вычисляется по формуле: “<Размер БД>*4 + <объем пользовательских файлов>*2 + 1ГБ”, но не менее 100 Гб.

Программные требования:

- Операционная система:
 - Linux (Рекомендуется: Дистрибутивы семейства Debian x64 или CentOS x64);
 - FreeBSD;
 - MacOS;
 - Microsoft Windows Server 2012 / 2008 / 2003;
 - Microsoft Windows NT 5.0 / XP / Vista / 7 / 8 / 8.1;
- Tomcat версии 6 и выше;
- Java версии 6 и выше.

Сервер баз данных

Аппаратные требования:

- 8 Гб ОЗУ и выше;
- 8-ядерный процессор с тактовой частотой от 2,5 ГГц и выше;
- Отдельный SSD/SAS-диски со свободным дисковым пространством не менее 120 Гб для размещения базы данных;

При выборе SDD/SAS - дисков для сервера баз данных следует учитывать возможное увеличение объемов БД в процессе эксплуатации системы.

- Использование аппаратного RAID1/10 ;
- Доступ к сети Интернет от 5 Mbps (рекомендуется 10 Mbps и выше);
- Доступ к локальной сети 1 Gbps/switched;

Программные требования:

- Операционная система:
 - Linux (Рекомендуется: Дистрибутивы семейства Debian x64 или CentOS x64)
 - FreeBSD.
 - MacOS.
 - Microsoft Windows Server 2012 / 2008 / 2003.
 - Microsoft Windows NT 5.0 / XP / Vista / 7 / 8 / 8.1.
- СУБД:
 - Firebird версии 2.5 и выше.
 - PostgreSQL версии 9.1 и выше

Система хранения резервных копий данных

Аппаратные требования:

- Свободное дисковое пространство не менее 4 ТБ.
- Доступ к локальной сети 1 Gbps/switched.

Желательно расположить устройство в отдельном от места размещения основного серверного оборудования помещении.

1.7.2. Программно-аппаратные требования к клиентской части

Аппаратные требования:

- 2048 МБ ОЗУ.
- 2-ядерный процессор с тактовой частотой от 2.1 ГГц и выше;
- Свободное дисковое пространство от 1 Гб.
- Доступ к сети Интернет, от 2 Mbps (рекомендуется 10 Mbps).

Программные требования:

- Операционная система Windows, Linux, Mac OS, Android, равным образом и любая другая в которой можно установить web-браузер с поддержкой стандарта html5;
- Браузер Firefox версии 3.1 и выше или Chrome версии 3.6 и выше
Последнюю версию веб-браузеров можно скачать со следующих сайтов:
- Google Chrome - с сайта корпорации [Google](#).
- Mozilla Firefox - с сайта [Mozilla Россия](#).

1.8. Необходимый уровень подготовки администратора системы

Администратор системы, построенной на платформе Motiware Melody One, должен соответствовать следующим требованиям:

- знать основные принципы построения локальных сетей (протоколы, сетевое оборудование и т.п.);
- владеть навыком администрирования локальных сетей под управлением различных операционных систем (Windows, Linux, Unix);
- иметь представление об организации доступа к веб-серверу;
- уметь настраивать клиентские приложения;
- владеть навыком управления правами доступа к объектам информационных систем.
- обладать практическим опытом выполнения работ по установке, настройке и 100% администрированию различных СУБД;
- иметь представление об основах обеспечения информационной безопасности;

1.9. Сопровождение систем на платформе Motiware Melody One

Сопровождение систем и решений, разработанных на платформе Motiware Melody One, может осуществляться специалистами компании ООО "Мотивэа" и включать в себя все перечисленные ниже виды сопровождения.

1.9.1. Разработка архитектуры информационной системы

Разработка архитектуры системы содержит:

- установление требований к проектируемой системе (решению);
- разработка концепции, включающая в себя конструирование концептуальной модели предметной области;
- составление спецификаций: спецификации обработки данных в проектируемой системе, спецификации пользовательского интерфейса системы, спецификации деятельности в предметной области с учетом внедрения системы и т.п.;

1.9.2. Проектирование и разработка функционала

Проектирование и разработка функционала включает в себя следующие мероприятия:

- написание технического задания (ТЗ);
- подготовка технического проекта;
- разработка основного и дополнительного функционалов;
- обновление измененных конфигураций;

1.9.3. Обучение

Обучение может проводиться как для администраторов, так и для пользователей в следующих формах:

- индивидуальное очное обучение;
- групповое очное обучение;
- дистанционное обучение (индивидуальное и групповое);
- предоставление обучающих материалов (учебные пособия и видеолекции);
- создание тестовых площадок для практической работы.

1.9.4. Консультирование

Консультирование может вестись в очном и дистанционном форматах по следующим направлениям:

- аудит и описание бизнес-процессов;
- подготовка рекомендаций для оптимизации бизнес-процессов;
- разработка мероприятий по увеличению производительности информационной системы.

1.9.5. Техническое сопровождение

Техническое сопровождение включает в себя:

- обновление и настройку типовых решений (конфигураций);
- установку и развертывание платформы;
- тестирование баз данных;
- создание резервных копий;
- выполнение регламентных работ.

1.10. Термины и определения (принятые обозначения)

Термин	Описание
--------	----------

Тип данных	Набор полей, значений, триггеров, операций и экземпляров с определенными свойствами. Тип данных описывает поведение и представление записей посредством определенных характеристик, именуемых полями.
Таблица (экземпляр)	Экземпляры типов данных. Виртуальная таблица, проецированная на реальную таблицу базы данных. Экземпляр создается на основе описанных в типе данных полей и меток.
Кэш	Промежуточный буфер с быстрым доступом, содержащий информацию, которая может быть запрошена с наибольшей вероятностью.
Право	Доступ пользователя к определенным функциям или элементам системы.
Роль	Набор различных прав в системе.
Операция	Поведение типов данных, а так же действия, которые могут производиться над типами данных. Типичными примерами операции являются операции “Добавить”, “Добавить к узлу”, “Редактировать” и т. д. Операции задаются наименованием, входными параметрами и алгоритмом выполнения. Причем для системных операций алгоритм заданный (штатный), а для пользовательских - произвольный (разработанный пользователем).
Представление данных	Сущность системы, отвечающая за параметры отображения набора данных.
Фильтр	Способ получения данных по заданным условиям.
Идентификатор	Уникальный код, присвоенный объекту
Поле	Переменная (набор переменных), характеризующая конкретный тип данных. Поля описывают структуру таблицы (экземпляра). По полям создаются столбцы для экземпляров

	(таблиц) и определяются связи между объектами. Кроме того, поле задает в каком виде будут храниться значения для этого поля в системе.
Метка	Значение поля. По умолчанию у каждого поля одна метка (значение).
Ячейка	Пересечение поля и метки
Вычисляемое выражение	Выражение (формула или константа), определяющее способ расчета значения
Шаблон	Преднастроенные записи для разных типов данных.

ГЛАВА 2. Архитектура платформы Motiware Melody One

Системная архитектура Motiware Melody One состоит из следующих взаимосвязанных компонентов:

- прикладной архитектуры;
- сетевой архитектуры.
- архитектуры данных.

2.1. Прикладная архитектура платформы Motiware Melody One

В основу прикладной архитектуры платформы Motiware Melody One положен принцип трехуровневой (трехзвенной) архитектуры. В трехуровневой архитектуре, между окружением клиента (интерфейса пользовательской системы) и окружением сервера управления базой данных добавляется еще один промежуточный уровень, называемый сервером приложений.

2.1.1. Трехуровневая архитектура платформы Motiware Melody One

Трехуровневая архитектура представляет собой архитектурную модель, которая состоит из трех взаимосвязанных уровней:

- уровня представления,
- уровня логики,
- уровня данных.

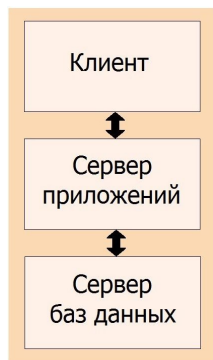


Рис. 2.1.1.1. Трехуровневая архитектура

На рисунке 2.1.1.1. показаны три взаимосвязанных программных приложения трехуровневой архитектуры, а именно:

- клиентское приложение (слой клиента),
- сервер приложений (слой логики, связующий слой),
- сервер баз данных (слой данных).

В отличие от двухуровневой архитектуры клиент-сервер, в трехуровневой архитектуре большая часть бизнес-логики (деловой логики), ранее входившая в клиентские приложения, выделяется в отдельный уровень (звено), которое называется сервером приложений.

При этом на уровне клиента остается практически один пользовательский интерфейс, в котором содержатся алгоритмы авторизации и шифрования, осуществляется проверка вводимых значений на допустимость и соответствие формату, выполняются несложные операции с данными (сортировка, группировка, подсчет значений), уже загруженными на терминал. Взаимодействие клиента осуществляется только с сервером приложений.

На сервере приложений сосредотачивается основная часть бизнес-логики, а за его пределами остаются фрагменты, выполняемые у клиента, как и элементы логики, погруженные в базу данных. Благодаря этому, по сравнению с двухуровневой архитектурой, в трехуровневой архитектуре существенно экономится время полного выполнения запроса с возвратом результата пользователю.

Сервер баз данных обеспечивает хранение данных и реализуется при помощи средств управления базами данных (СУБД). Обращение к серверу баз данных производится только с уровня сервера приложений.

Работа в рамках трехуровневой архитектуры ведется следующим образом.

1. Пользователи запускают клиентское приложение (тонкий клиент), которое инициирует обращение к программному обеспечению, реализующему бизнес-логику и расположенному на сервере приложений.
2. Сервер приложений обрабатывает обращение пользователя и формирует запросы к базе данных при помощи языка запросов (от сервера приложений к серверу баз данных передается лишь текст запроса).
3. При получении запроса от сервера приложений СУБД инкапсулирует внутри себя все сведения о физической структуре базы данных, расположенной на сервере. и инициирует обращения к данным, находящимся на сервере баз данных, результат выполнения запроса копируется на сервер приложений.
4. Сервер приложений производит дополнительную обработку данных, полученных от сервера баз данных, в соответствии с требованиями бизнес-логики и направляет результат в приложение пользователя.
5. Клиентское приложение с помощью пользовательского интерфейса отображает результат исполнения запроса.

2.1.2. Функциональная схема платформы Motiware Melody One

Функциональная схема платформы Motiware Melody One, реализованная на основе трехуровневой архитектуры, изображена на рисунке 2.1.2.1. На данной схеме присутствуют основные компоненты платформы:

- хранилище данных,
- репозитории:
 - типов данных,
 - форм отчетных и информационно-аналитических,
 - процессов,
 - индексов поиска,
 - настроек интеграции со внешними системами,
- генераторы:
 - входных форм,
 - форм пользовательского интерфейса,
 - отчетных форм
- модули выполнения:
 - процессов загрузки и выгрузки данных,
 - поиска,
 - Workflow - процессов,
 - авторизации,

- сканирования,
- редактирования файлов,
- интеграции со внешними системами,
- ЭЦП
- дизайнеры:
 - типов данных,
 - процессов загрузки и выгрузки данных,
 - информационно-аналитических форм,
 - отчетных форм,
 - Workflow - процессов,
- информационно-аналитические порталы и порталы взаимодействия.

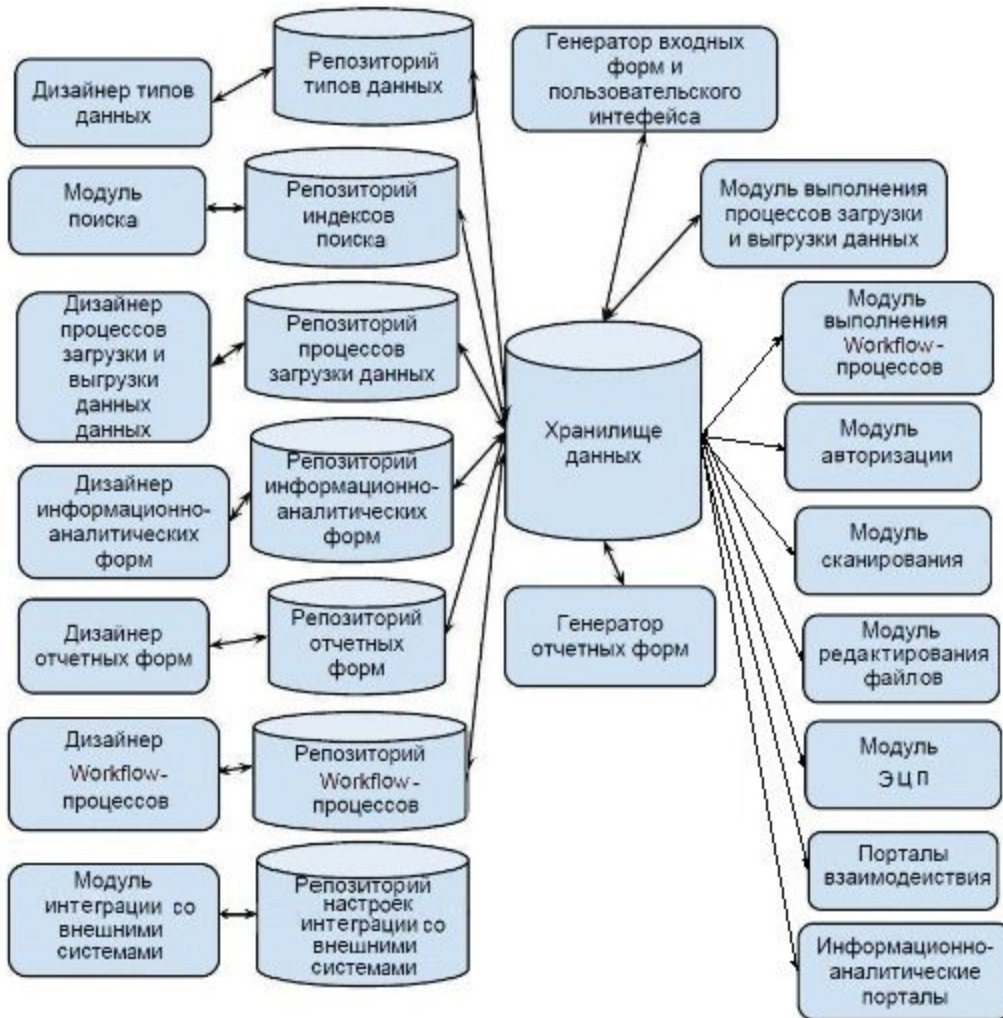


Рис. 2.1.2.1. Функциональная схема платформы Motiware Melody One

2.1.3. Структурная схема платформы Motiware Melody One

Структурная схема платформы Motiware Melody One представлена на рисунке 2.1.3.1. в виде трех частей:

- Рабочее место пользователя (клиент);
- Сервер приложений, включающий в себя ядро системы;
- Сервер баз данных.

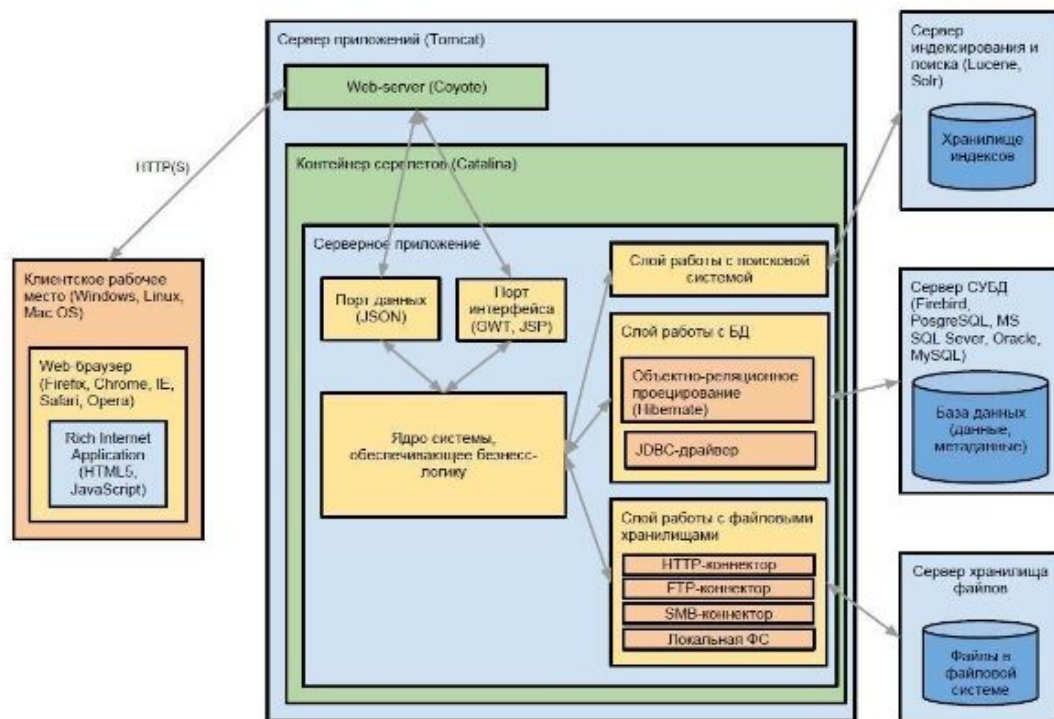


Рис. 2.1.3.1 Структурная схема платформы Motiware Melody One

Клиент. Благодаря размещению продукта на сервере, программное обеспечение пользователя может быть самым разнообразным. Поддерживаются основные типы операционных систем (Windows, Linux, Mac OS), как и все современные браузеры (Firefox, Chrome, IE, Safari, Opera) с разрешённым к использованию JavaScript для корректного отображения в окне браузера.

Сервер приложений. Сервер приложений Apache Tomcat (кроме данного контейнера сервлетов могут быть использованы и другие контейнеры) позволяет запускать веб-приложения. Компонентами сервера являются:

- Web-server Coyote, являющийся компонентом стека HTTP Tomcat, поддерживает протокол HTTP 1.1 для веб серверов или контейнера приложений, который прослушивает входящие соединения на определённом TCP порту сервера и

пересылает запросы в механизм Tomcat для обработки запросов и отправляет ответ назад запрашивающему клиенту;

- Catalina — контейнер сервлетов Tomcat а. Catalina реализует спецификацию сервлетов и JavaServer Pages (JSP).

Ядро системы, обеспечивающее бизнес-логику, выполняет две основные функции:

- При помощи порта данных (для передачи используется текстовый формат JSON) и порта интерфейса (для создания интерфейса используется GWT и JSP) реализует взаимодействие с пользователем;
- Осуществляет взаимодействие с сервером баз данных через три слоя – слой работы с поисковой системой, слой работы с базой данных и слой работы с файловыми хранилищами.

Сервер баз данных. Сервер баз данных включает в себя три основных элемента:

- Сервер индексирования – хранилище индексов, используемых при поиске;
- Сервер СУБД – базы данных и метаданные;
- Сервер хранилища файлов – файлы, расположенные в файловой системе.

2.2. Сетевая архитектура платформы Motiware Melody One

Высокая производительность и отказоустойчивость систем разработанных на базе платформы Motiware Melody One обеспечивается за счет реализации принципа горизонтального масштабирования, т.е. соединении воедино нескольких вычислительных узлов, как это показано на рисунке 2.2.1., при котором нагрузка между узлами перераспределяется.

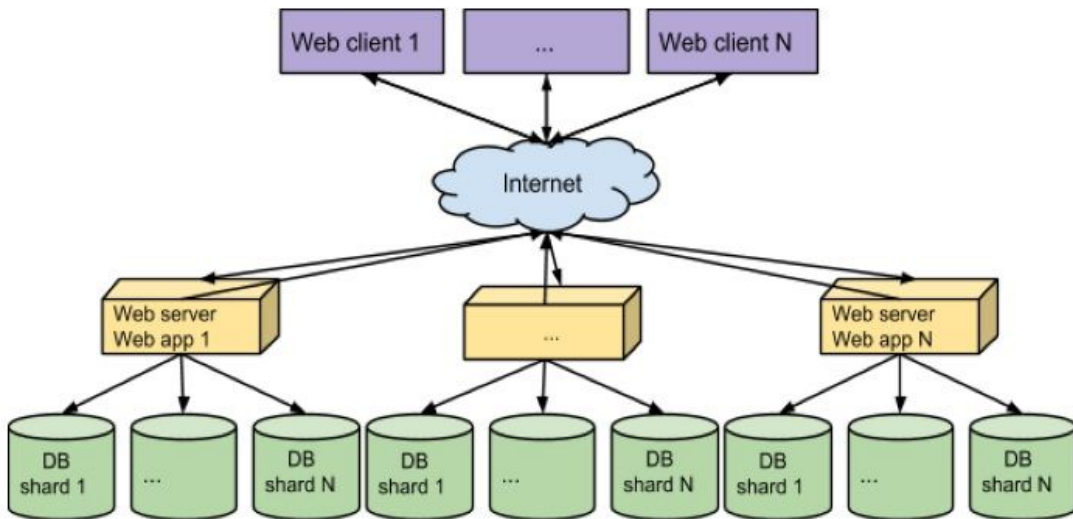


Рис. 2.2.1. Реализация принципа горизонтального масштабирования систем разработанных на платформе Motiware Melody One

Под узлом здесь понимается один или несколько серверов, предназначенных для обеспечения определенных функциональных характеристик системы. Как правило узел состоит из одного или нескольких серверов приложений и серверов баз данных. В свою очередь сервер приложений состоит из Apache, Heartbeat, Apache Tomcat. Причем:

- Heartbeat - обеспечивает запуск apache'а в случае отказа сервера, отвечающего за распределения нагрузки, и перевод балансировщика на новый сервер Apache, с модулем mod_jk - используется в качестве балансировщика;
- Apache Tomcat - контейнер Java-приложений;

Для обеспечения высокой отказоустойчивости и производительности настраивается репликация сессий, при помощи стандартного класса Tomcat'a SimpleTcpCluster. В результате такой настройки отказ одного или даже нескольких приложений не приводит к потере сессии пользователя.

На рисунке 2.2.2. более подробно отображена структура представленного на рисунке 2.2.1. кластера, а именно один из его узлов.

В качестве примера рассмотрим узел, который состоит из сервера(ов) приложений и связанного и ним сервера(ов) баз данных, при этом сервер приложений включает в себя: Apache, Heartbeat, Apache Tomcat, а в применяемой СУБД PostgreSQL для реализации масштабируемости базы данных используется разделение её на части (таблицы) и размещение их на отдельных серверах. В случае образования сверхбольших таблиц, предусмотрена возможность разделения и их на части с последующим размещения на серверах. Имеется возможность задания сервера для хранения Таблиц (Sheet) и определенных типов данных (Metaobjects).

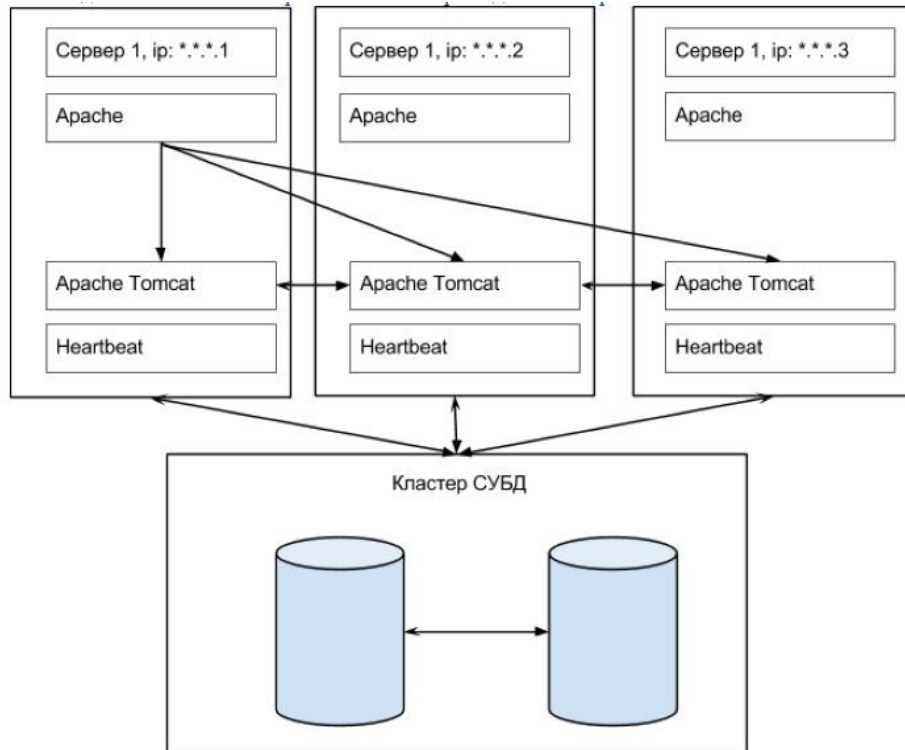


Рис.2.2.2. Структура кластера

Для уменьшения нагрузки на сервера баз данных доступна настройка репликации master-slave, обладающая следующим принципом работы:

- Создаются копии основной базы данных, обладающие ограниченным функционалом – доступна только операция чтения данных.
- Операции по добавлению/изменению/удалению данных производятся в базе данных, размещённой на основном сервере.
- После внесения каких-либо изменений в базу данных на основном сервере, копии базы данных, хранящиеся на других серверах, обновляют своё содержимое в соответствии с основной базой данных.

Таким образом, принимая во внимание тот факт, что основной вклад в нагрузку на сервер вносит процесс считывания данных, в нашем случае нагрузка на основной сервер может быть в значительной степени уменьшена.

Для систем, построенных на платформе Motiware Melody One, взаимодействие при обмене данными между различными инсталляциями системы может происходить на различных коммуникационных шинах. Таким образом реализован шлюз для системы Межведомственного электронного документооборота (МЭДО), рисунок 2.2.3.

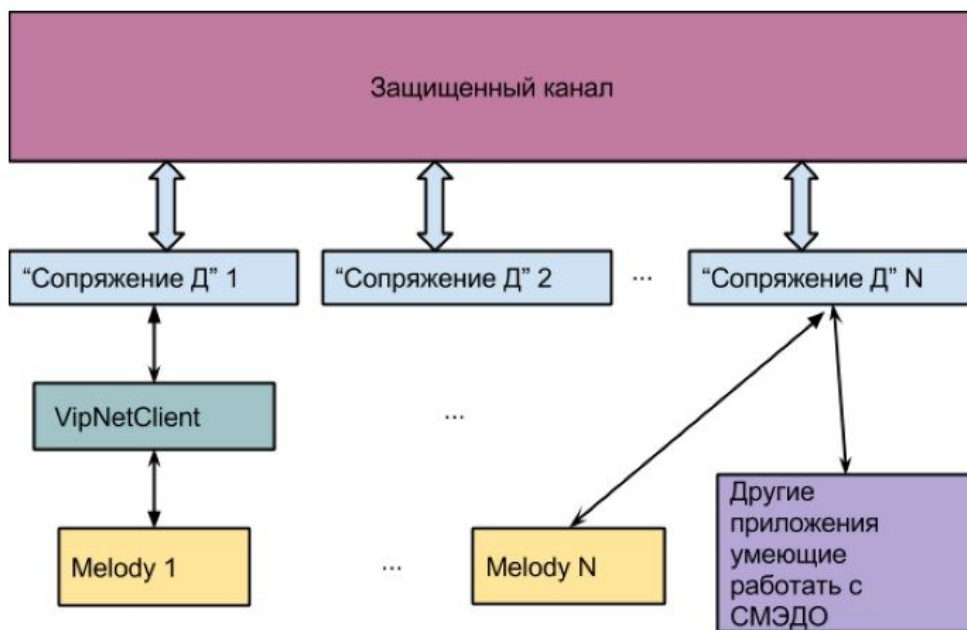


Рис.2.2.3. Обмен данными между различными инсталляциями системы

В этом случае к МЭДО можно подключиться двумя способами: как через VipNet Client, так и через Сопряжение Д, (см. рис. 2.2.3.).

Обмен данными можно осуществлять и непосредственно между различными инсталляциями системы, разработанной на базе платформы Motiware Melody One, при помощи настроенных ETL - процессов (см. подробнее п. 5.7. "Настройка процессов"). Схема такого обмена данными в рамках одного бизнес-процесса представлена на рисунке 2.2.4.



Рис.2.2.4. Обмен данными с помощью настроенных ETL - процессов

2.3. Архитектура данных

Архитектура данных подробно описана в главе 4. "Информационные объекты платформы Motiware Melody One".

2.4. Принципы разработки архитектурных решений

Основные принципы разработки архитектурных решений (систем) на платформе Motiware Melody One включают в себя:

1. **Принцип системности**, который предполагает учет всех взаимосвязей, анализ отдельных частей системы (решения) как ее самостоятельных структурных составляющих и выявление роли каждой из них в функционировании всей системы в целом. Другими словами, при декомпозиции должны быть установлены такие связи между структурными компонентами системы, которые обеспечивают цельность корпоративной системы и ее взаимодействие с другими системами. В этом случае любая бизнес-задача будет рассматриваться комплексно со всеми возможными информационными связями.
2. **Принцип открытости (развития)**, который заключается в том, что внесение изменения в систему (решение), обусловленное самыми различными причинами (внедрением новых информационных технологии, изменением законодательства, организационной перестройкой внутри фирмы и т. п.), осуществляется путем дополнения системы без нарушения ее функционирования.
3. **Принцип своевременности** отражает тот факт, что при разработке системы (решения) используются информационные интерфейсы, благодаря которым она может взаимодействовать с другими системами согласно установленным правилам.
4. **Принцип унификации (стандартизации)** содержит в себе положение о том, что рационально используются типовые, унифицированные и стандартизованные элементы, пакеты прикладных программ, комплексы, компоненты.
5. **Принцип эффективности**, который предусматривает достижение рационального соотношения между затратами на создание системы (разработку решения) и целевыми эффектами, включая конечные результаты, отражающиеся на прибыльности и получаемые по окончании внедрения решения в управленческие процессы.

Таким образом, решения, разработанные на базе платформы Motiware Melody One, использующие изложенные выше принципы, представляют из себя удобный, современный и высокотехнологичный инструмент, позволяющий организациям всех форм собственности максимально эффективно реализовывать стратегию бизнеса и достигать бизнес - цели.

2.5. Взаимодействие с бизнес-архитектурой

В основу архитектурного подхода положена идея тесного, оперативного и эффективного взаимодействия бизнеса и информационных технологий, в котором и бизнес, и информационные технологии рассматриваются как единое целое и объединяются в рамках понятия архитектуры предприятия.

В свою очередь архитектура информационных технологий и бизнес-архитектура организации представляют собой основной механизм интерпретации и реализации корпоративных бизнес-целей и целей бизнес-единиц (структурных подразделений) организации посредством информационных систем.

Информационные системы, созданные на базе платформы Motiware Melody One, предоставляют возможность сотрудникам организаций эффективно и результативно решать бизнес-задачи. Эта цель достигается через разработку и создание определенного количества взаимосвязанных архитектурных представлений на всех уровнях корпоративной архитектуры предприятия:

- уровне информации,
- уровне бизнес-процессов,
- уровне прикладных сервисов,
- уровне инфраструктурных сервисов.

Уровень информации определяет основные типы данных:

- документы;
- основные данные (информация о клиентах, поставщиках, продуктах, заказах, финансовая информация и т.п.);
- отчеты (аналитические и финансовые);
- метаданные (конфигурации, журналы, модели обработки данных и т.д.).

Уровень бизнес-процессов содержит в себе последовательность действий для решения бизнес-задач и включает в себя:

- иерархические модели бизнес-процессов;
- средства для формализации и проектирования бизнес-процессов;
- политики и бизнес-правила.

Уровень прикладных сервисов задает корпоративные приложения и механизмы их интеграции для реализации бизнес-процессов, сервисы для интеграции и управления информацией находящейся на соответствующем уровне.

К уровню инфраструктурных сервисов относятся такие технологические компоненты как: сервисы запросов, обмена и передачи данных, транзакционные сервисы, коммуникационные сервисы и т.п.

Платформа Motiware Melody One располагает широкими возможностями для разработки эффективных, результативных и взаимосвязанных решений на каждом из приведенных выше уровней бизнес-архитектуры предприятия.

ГЛАВА 3. Интерфейс пользователя платформы Motiware Melody One

3.1. Общие сведения о пользовательском интерфейсе

Платформа Motiware Melody One имеет простой в освоении, понятный интуитивно и гибко настраиваемый интерфейс пользователя (рис.3.1.1.), в котором выделены две основные области:

- главное меню (1),
- рабочая область (2).

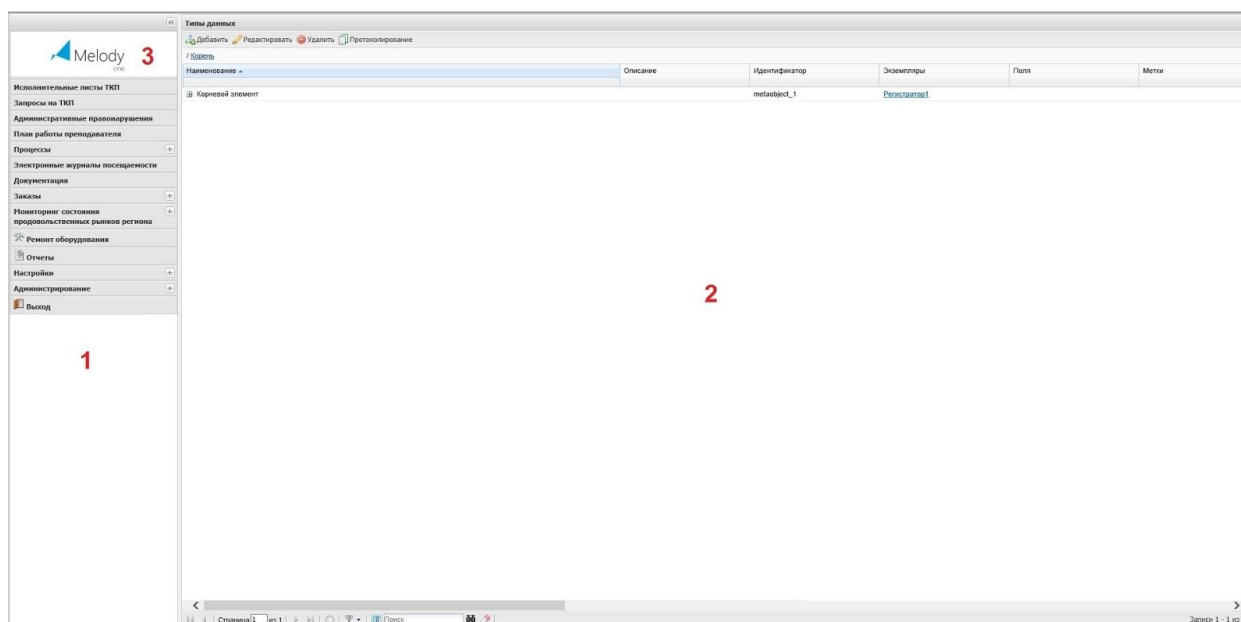


Рис. 3.1.1. Общий вид интерфейса пользователя платформы Motiware Melody One

В определенных случаях к этому виду может быть добавлена правая панель, выполненная в виде, похожем на главное меню, рабочая область может быть разделена на две части (окна) по горизонтали.

Пункты меню, для которых не задано наименование, могут служить для размещения изображений, в частности, логотипа компании. Для этого в качестве иконки пункта меню необходимо выбрать нужное изображение и настроить параметры отображения данного пункта меню. Пункт меню, содержащий логотип, располагается, как правило, в самой верхней позиции. Для подобных пунктов меню чаще всего не задается поставщик данных.

3.2. Главное меню

3.2.1. Структура главного меню

Главное меню платформы Motiware Melody One образует иерархическую структуру, в которой отражены функциональные возможности платформы Motiware Melody One и содержит в себе следующие элементы:

- пункты меню,
- подпункты меню,

Пункты главного меню могут как запускать отдельную операцию приложения (например, отображение данных, выход из системы и т.п.), разработанного на платформе Motiware Melody One, так и содержать в себе группы подпунктов, которые, в свою очередь, запускают определенные операции. Обычно главное меню содержит системные пункты и пункты пользовательские, разработанные для решения различных задач пользователей.

Для пункта меню может быть указано наименование и/или выбрана иконка (графический файл, позволяющий идентифицировать данный пункт меню в списке).

К системным пунктам главного меню относятся пункты: процессы, отчеты, настройки и администрирование. Пункт “администрирование” может включать в себя системные подпункты такие как: статистика, кэш, импорт таблицы из excel, экспорт, поиск, типы данных.

Пользовательские пункты (подпункты) Motiware Melody One могут быть представлены самым широким образом в соответствии с пожеланиями пользователей.

Таким образом, главное меню платформы Motiware Melody One - это набор позиций (команд), структурированный иерархически для доступа к функциональным возможностям системы, которыми пользователь, как предполагается, будет пользоваться чаще всего.

3.2.2. Пункты и подпункты

Главное меню платформы Motiware Melody One, как показано на рисунке 3.2.2.1., содержит пункты (1) и подпункты (2),



Рис. 3.2.2.1. Пункты и подпункты главного меню платформы Motiware Melody One

При разработке решений на базе платформы Motiware Melody One рекомендуется однородные операции объединять в подпункты.

При этом в главном меню системы не допускается создание подпунктов далее второго уровня.

3.3. Системные пункты главного меню

Главное меню содержит операции, сгруппированные в виде иерархической структуры, которые с высокой вероятностью будут использоваться пользователем чаще всего. В настройке интерфейса пользователя по умолчанию присутствуют следующие системные пункты главного меню:

- Процессы;
- Отчеты;
- Настройки;
- Администрирование;
- Выход.

3.3.1. Пункт "Настройка"

Пункт системного меню "Настройка" предназначен для быстрого доступа к операциям, при помощи которых можно внести или отредактировать информацию о конкретном пользователе и назначить ему права доступа. Данный пункт содержит два подпункта:

- Мои реквизиты;
- Группы пользователей;

3.3.1.1. Подпункт "Мои реквизиты"

Подпункт "Мои реквизиты", как показано на рисунке 3.3.1.1.1. - (1), содержит шесть вкладок (2) с полями для ввода конкретной информации (значений) о пользователе, а именно:

- Общее;
- Адрес;
- Контакты;
- Заместители;
- Действия пользователя;

Наименование	Значение
ФИО	Иванов И.И.
КПП	
*Логин	ИвановИИ
*Пароль	
*Подтверждение пароля	
Действительный	<input checked="" type="checkbox"/>
Фамилия	Иванов
Имя	Иван
Отчество	Иванович
Пол	
Почта	
День рождения	
Роли	ROLE_COMMONUSER
Должность	
Муниципальное образование	

Рис.3.3.1.1.1. Подпункт "Мои реквизиты"

Вкладка "Общее" (рис.3.3.1.1.2.) содержит следующие поля:

- ФИО - поле, вычисляемое при помощи функции ifExes (см. Приложение 4), в котором указана фамилия и инициалы пользователя, причем фамилия и инициалы вычисляются на основе полей "Фамилия", "Имя", "Отчество", в том случае, если эти поля не заполнены используется поле "Логин";
- Логин - обязательное поле, в котором указывается логин пользователя, под которым он сможет авторизоваться в системе;
- Пароль - обязательное поле, в котором указывается пароль для входа пользователя в систему;

- Подтверждение пароля - обязательное поле для повторного подтверждения задаваемого пароля;
- Действительный - обязательное поле, с помощью которого активируется учетная запись пользователя.
- Фамилия - необязательное поле для указания реквизитов пользователя;
- Имя - необязательное поле для указания реквизитов пользователя;
- Отчество - необязательное поле для указания реквизитов пользователя;
- Пол - необязательное поле для указания реквизитов пользователя;
- День рождения - необязательное поле для указания реквизитов пользователя;
- Роли - обязательное поле, в котором можно назначить пользователю роли, выбрав их из открываемого по иконке со знаком "+" списка.
- Должность - в данном поле можно задать должность пользователя (поле необязательное для заполнения);
- Муниципальное образование - необязательное поле, в котором можно выбрать муниципальное образование, где находится данный пользователь, из списка.




Пользователи				
Общее	Адрес	Контакты	Заместители	Действия пользователя
Наименование	Значение			
ФИО				
КПП				
*Логин				
*Пароль				
*Подтверждение пароля				
Действительный	<input checked="" type="checkbox"/>			
Фамилия				
Имя				
Отчество				
Пол				
Почта				
День рождения				
Роли				
Должность				
Муниципальное образование				

Рис.3.3.1.1.2. Вкладка “Общее”

Вкладка "Адрес" (рис.3.3.1.1.3.) предназначена для внесения информации о почтовом адресе пользователя. Доступ к форме для внесения информации производится

посредством нажатия на кнопку “добавить”, после чего открывается форма для заполнения, которая содержит поля следующего содержания:

- Страна;
- Регион;
- Почтовый индекс;
- Населенный пункт;
- Улица;
- Дом;
- Квартира.

Пользователи / Адрес/Добавление	
Поле	Значение
Страна	
Регион	
Почтовый индекс	
Населенный пункт	
Улица	
Дом	
Квартира	

Рис.3.3.1.1.3. Вкладка "Адрес"

Во вкладку "Контакты" (рис.3.3.1.1.4.) можно внести иную контактную информацию, например, номер мобильного телефона или адрес в Skype. Как и для вкладки "Адрес", доступ к полям для заполнения осуществляется при нажатии на кнопку “добавить”, при этом открывается форма с полями:

- Тип (данные типы аналогичны описанным в <https://ru.wikipedia.org/wiki/VCard>);
- Контакт.

Пользователи / Контакты/Добавление	
Поле	Значение
Тип	
Контакт	

Рис.3.3.1.1.4. Вкладка "Контакты"

Вкладка "Заместители" (рис.3.3.1.1.5.) содержит два поля:

- Заместитель;
- Включен.


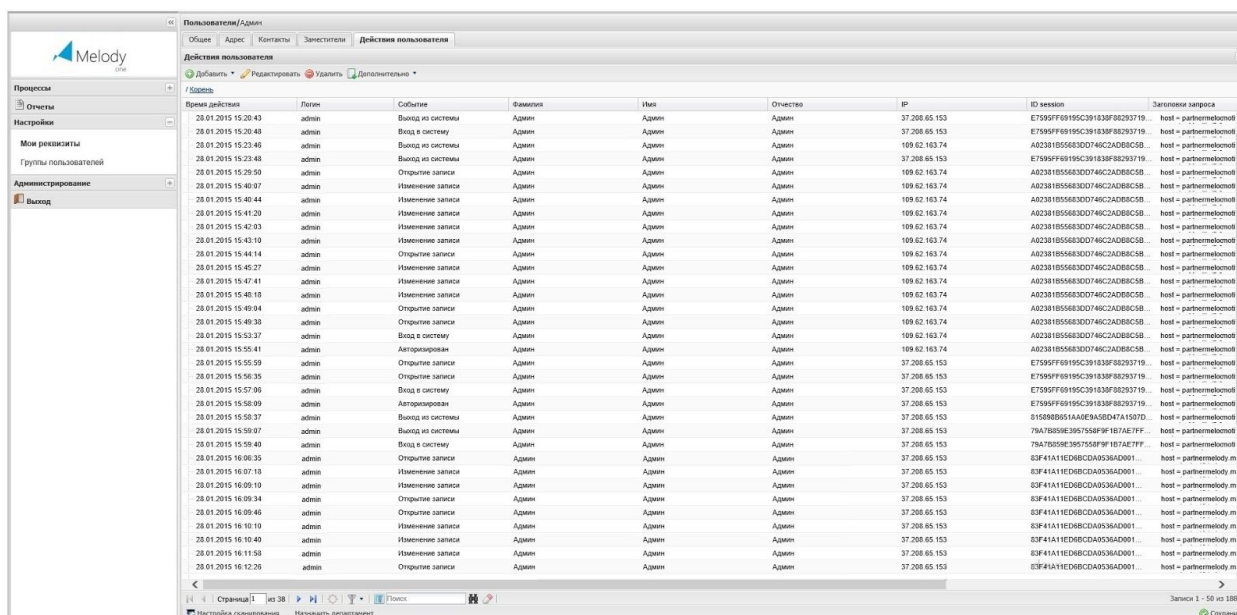
Пользователи / Заместители/Добавление	
Поле	Значение
Заместитель	
Включен	<input type="checkbox"/>

Рис.3.3.1.1.5. Вкладка "Заместители"

Здесь пользователь может выбрать себе заместителя из открывающегося списка (для доступа к списку необходимо кликнуть мышью по иконке со знаком "+") и включить его, проставив флаг в поле "Включен". Заместитель получает доступ ко всем правам и документам данного пользователя, на время отсутствия пользователя.

Во вкладке "Действия пользователя" (рис.3.3.1.1.6.) отображается журнал действий пользователя в данной системе.



Время действия	Логин	Событие	Фамилия	Имя	Отчество	IP	ID session	Заголовок запроса
28.01.2015 15:20:43	admin	Выход из системы	Админ	Админ	Админ	37.208.65.153	E759FF69195C391338F8B293719...	host = partnermelody.n
28.01.2015 15:20:48	admin	Вход в систему	Админ	Админ	Админ	37.208.65.153	E759FF69195C391338F8B293719...	host = partnermelody.n
28.01.2015 15:23:46	admin	Выход из системы	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:22:46	admin	Выход из системы	Админ	Админ	Админ	37.208.65.153	E759FF69195C391338F8B293719...	host = partnermelody.n
28.01.2015 15:29:59	admin	Открытие записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:40:07	admin	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:40:41	admin	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:41:29	admin	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:42:03	admin	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:43:10	admin	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:44:14	admin	Открытие записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:45:27	admin	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:47:41	admin	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:48:10	admin	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:49:04	admin	Открытие записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:49:38	admin	Открытие записи	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:53:37	admin	Вход в систему	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:55:41	admin	Авторизован	Админ	Админ	Админ	109.62.163.74	A02381855683D0746C2AD86C5B...	host = partnermelody.n
28.01.2015 15:55:59	admin	Открытие записи	Админ	Админ	Админ	37.208.65.153	E759FF69195C391338F8B293719...	host = partnermelody.n
28.01.2015 15:56:35	admin	Открытие записи	Админ	Админ	Админ	37.208.65.153	E759FF69195C391338F8B293719...	host = partnermelody.n
28.01.2015 15:57:06	admin	Вход в систему	Админ	Админ	Админ	37.208.65.153	E759FF69195C391338F8B293719...	host = partnermelody.n
28.01.2015 15:58:09	admin	Авторизован	Админ	Админ	Админ	37.208.65.153	E759FF69195C391338F8B293719...	host = partnermelody.n
28.01.2015 15:58:37	admin	Выход из системы	Админ	Админ	Админ	37.208.65.153	0158888061A9E9A80D47A1597D...	host = partnermelody.n
28.01.2015 15:59:07	admin	Выход из системы	Админ	Админ	Админ	37.208.65.153	78A7050E36757558F91674E7FF...	host = partnermelody.n
28.01.2015 15:59:40	admin	Вход в систему	Админ	Админ	Админ	37.208.65.153	78A7050E36757558F91674E7FF...	host = partnermelody.n
28.01.2015 16:06:35	admin	Открытие записи	Админ	Админ	Админ	37.208.65.153	83F41A11ED8BCDA936AD001...	host = partnermelody.n
28.01.2015 16:07:10	admin	Изменение записи	Админ	Админ	Админ	37.208.65.153	83F41A11ED8BCDA936AD001...	host = partnermelody.n
28.01.2015 16:09:10	admin	Изменение записи	Админ	Админ	Админ	37.208.65.153	83F41A11ED8BCDA936AD001...	host = partnermelody.n
28.01.2015 16:09:34	admin	Открытие записи	Админ	Админ	Админ	37.208.65.153	83F41A11ED8BCDA936AD001...	host = partnermelody.n
28.01.2015 16:09:46	admin	Открытие записи	Админ	Админ	Админ	37.208.65.153	83F41A11ED8BCDA936AD001...	host = partnermelody.n
28.01.2015 16:10:10	admin	Изменение записи	Админ	Админ	Админ	37.208.65.153	83F41A11ED8BCDA936AD001...	host = partnermelody.n
28.01.2015 16:10:40	admin	Изменение записи	Админ	Админ	Админ	37.208.65.153	83F41A11ED8BCDA936AD001...	host = partnermelody.n
28.01.2015 16:11:58	admin	Изменение записи	Админ	Админ	Админ	37.208.65.153	83F41A11ED8BCDA936AD001...	host = partnermelody.n
28.01.2015 16:12:26	admin	Открытие записи	Админ	Админ	Админ	37.208.65.153	83F41A11ED8BCDA936AD001...	host = partnermelody.n

Рис.3.3.1.1.6. Вкладка "Действия пользователя"

3.3.1.2. Подпункт "Группы пользователей"

Подпункт "Группы пользователей", представленный на рисунке 3.3.1.2.1. (1), служит для объединения пользователей в группы. Эти группы можно использовать в полях типа "Рабочая группа". Группы могут быть динамическими (можно задать фильтр формирования содержимого папки). Перечень групп пользователей является иерархическим.

Обладая определенными правами, в пункте меню "Группы пользователей" можно произвести добавление, удаление или редактирование записей различных пользователей

(из списка). При этом редактирование записей пользователей будет производиться так же, как описано в пункте 3.3.1.1. Подпункт "Мои реквизиты".

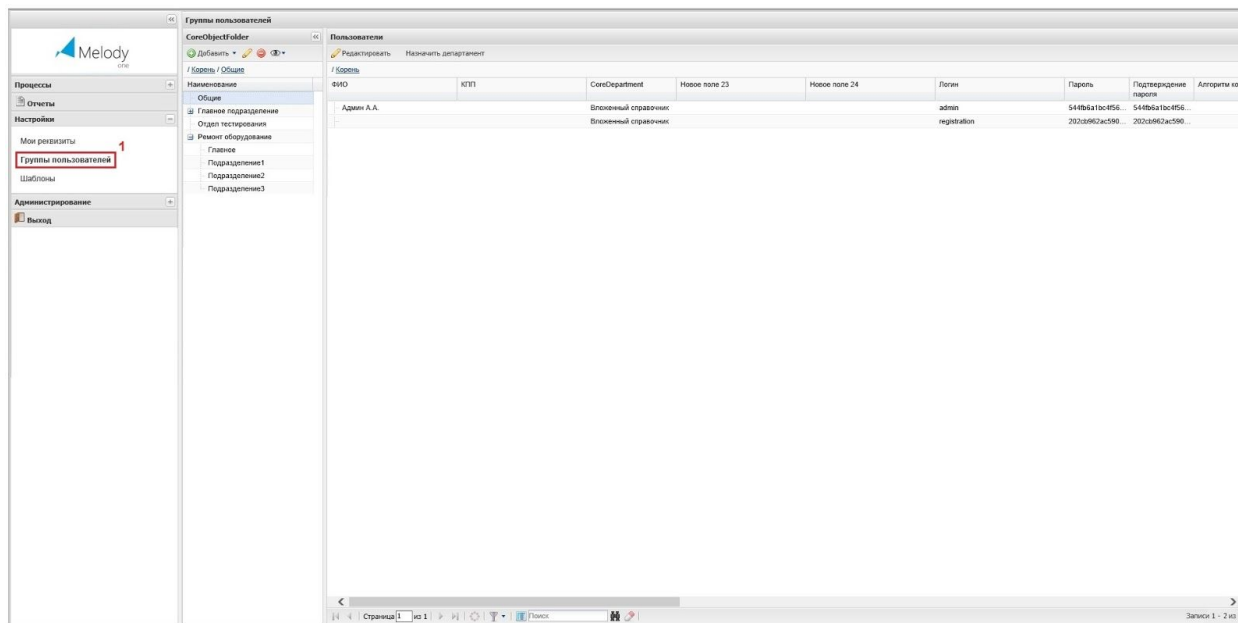


Рис.3.3.1.2.1. Подпункт “Группы пользователей”

3.3.2. Пункт "Администрирование"

Данный пункт системного меню “Администрирование” (рис. 3.3.2.1.) предназначен для администрирования систем, созданных на платформе Motiware Melody One, при этом администрирование этого раздела доступно пользователям, обладающим правами доступа роли (участникам роли) “Системный администратор”. По умолчанию пункт меню “Администрирование” содержит следующие подпункты:

- Роли;
- Типы данных;
- Статистика;
- Кэш.

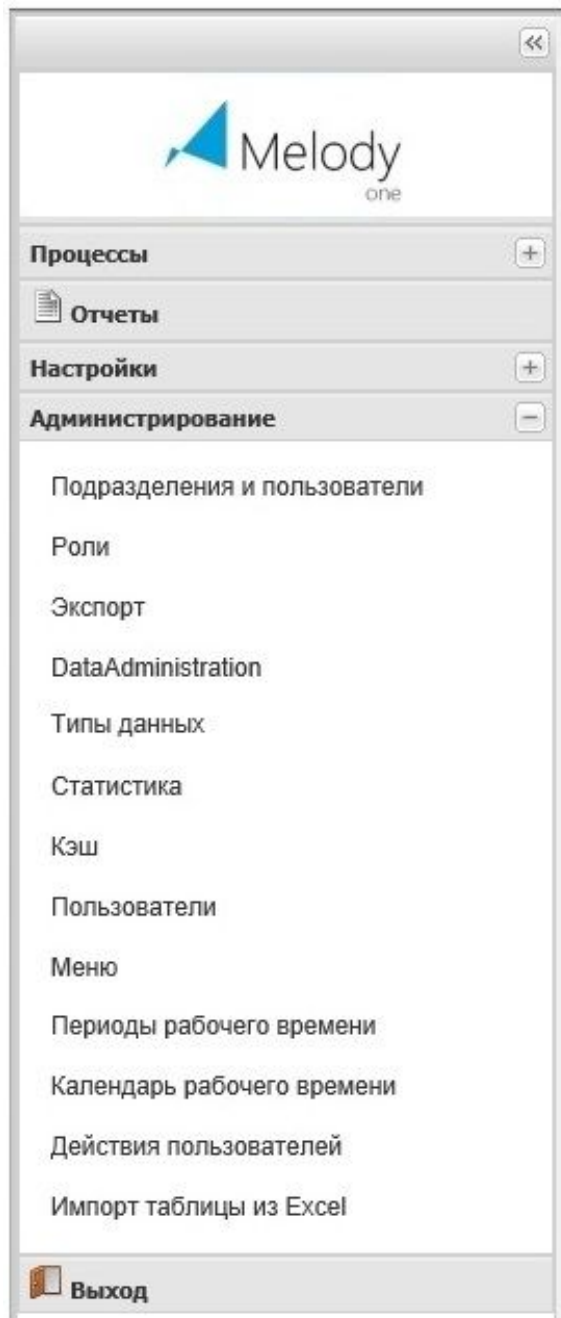


Рис. 3.3.2.1. Пункт "Администрирование"

3.3.2.1. Подпункт "Подразделения и пользователи"

В подпункте "Подразделения и пользователи", как показано на рисунке 3.3.2.1.1. - (1), производится администрирование пользователей и подразделений, здесь же можно добавить подразделение (2).

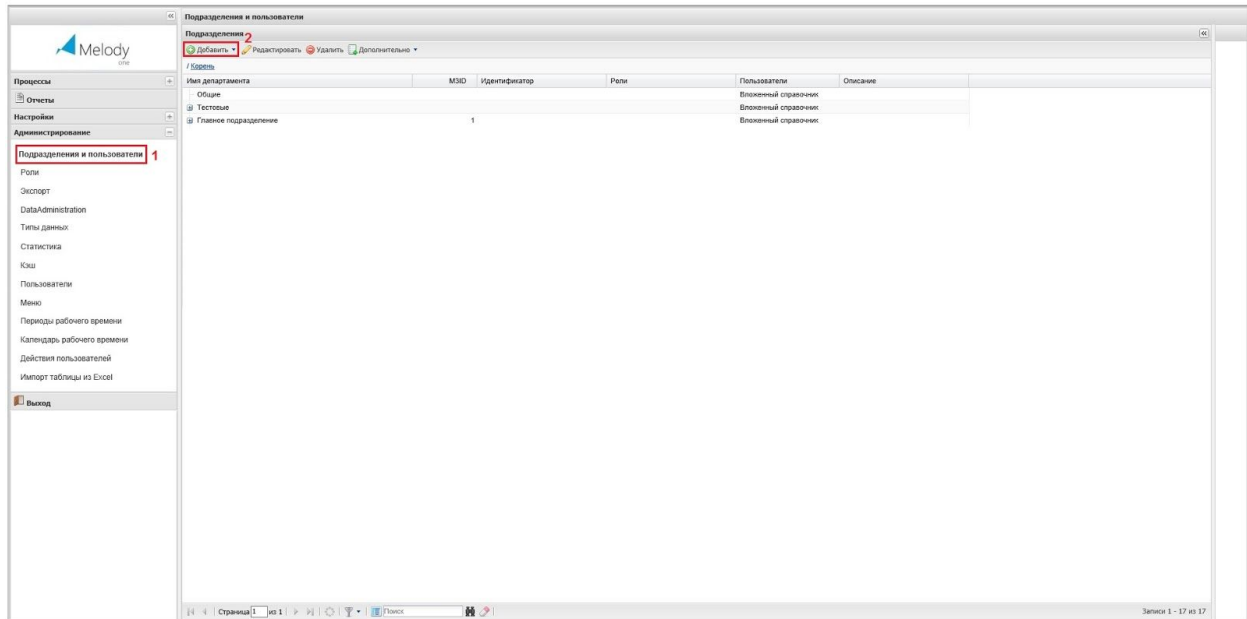


Рис. 3.3.2.1.1. Подпункт "Подразделения и пользователи"

Во вкладке Подразделение/Добавление можно создать подразделение (рис. 3.3.2.1.2.), определив его наименование в поле "Имя департамента" - (1) назначить права группе пользователей этого подразделения, добавив соответствующие роли, выбрав их из списка в поле "Роли" с помощью кнопки со знаком "+" - (2), в свою очередь в текстовом поле "Описание"- (3) можно отразить всю необходимую информацию об этом подразделении или о данной группе пользователей. Здесь же можно определить пользователей внутри данной группы (сотрудников соответствующего подразделения), добавив (4) их с помощью группы вкладок, аналогичных описанным в п. 3.3.1.1. Подпункт "Мои реквизиты"

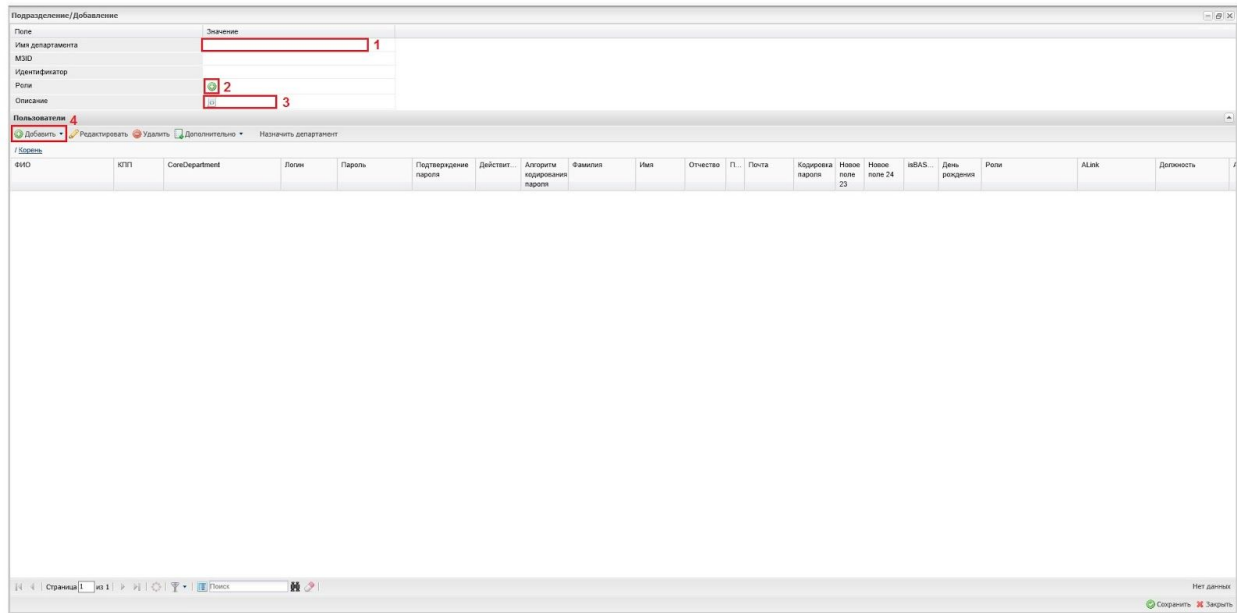


Рис. 3.3.2.1.2. Добавление подразделения

3.3.2.2. Подпункт "Роли"

Подпункт "Роли", как отображено на рисунке 3.3.2.2.1. - (1), предназначен для назначения и администрирования ролей, подробно администрирование ролей будет рассмотрено в разделе 5.4. "Управление ролями" настоящего руководства. В этом подпункте можно добавить роль (2), создать роль по шаблону (3), редактировать роль (4), удалить роль (5) или назначить права (6) для роли используя матрицу прав.

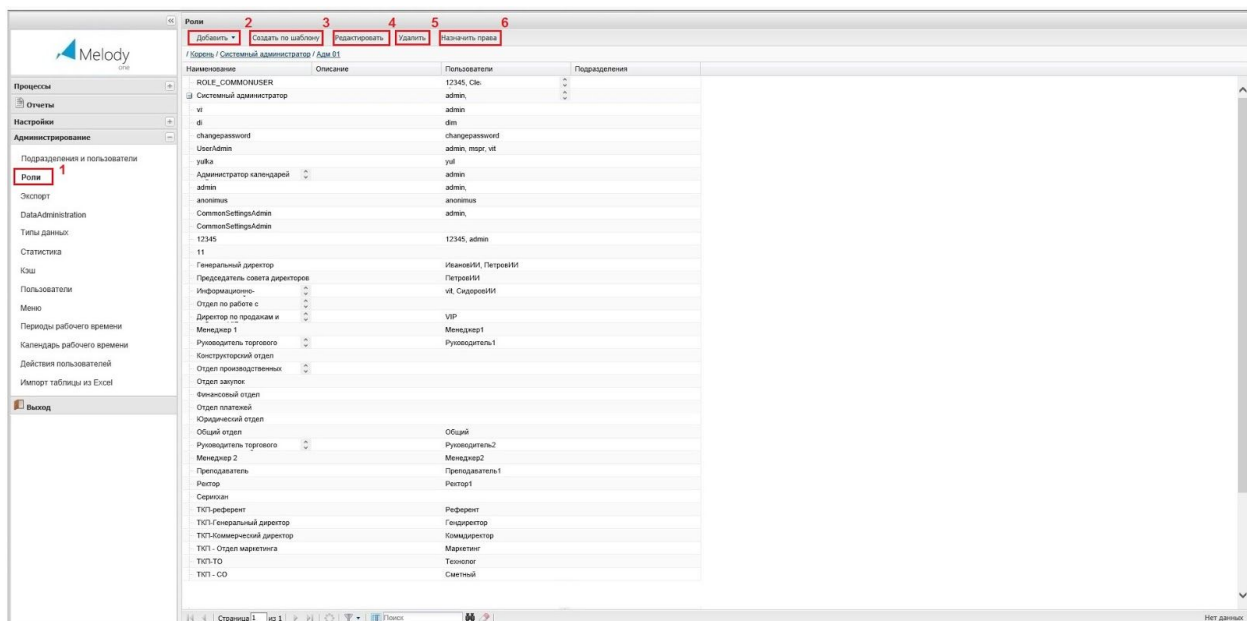


Рис. 3.3.2.2.1. Подпункт "Роли"

3.3.2.3. Подпункт "Экспорт"

Подпункт "Экспорт", как изображено на рисунке 3.3.2.3.1. - (1), предназначен для экспортирования типов данных в формате XML между инсталляциями систем разработанных на платформе Motiware Melody One. Выбор объекта для экспорта производится при помощи отметки в поле, относящемся к экспортируемому объекту (2).

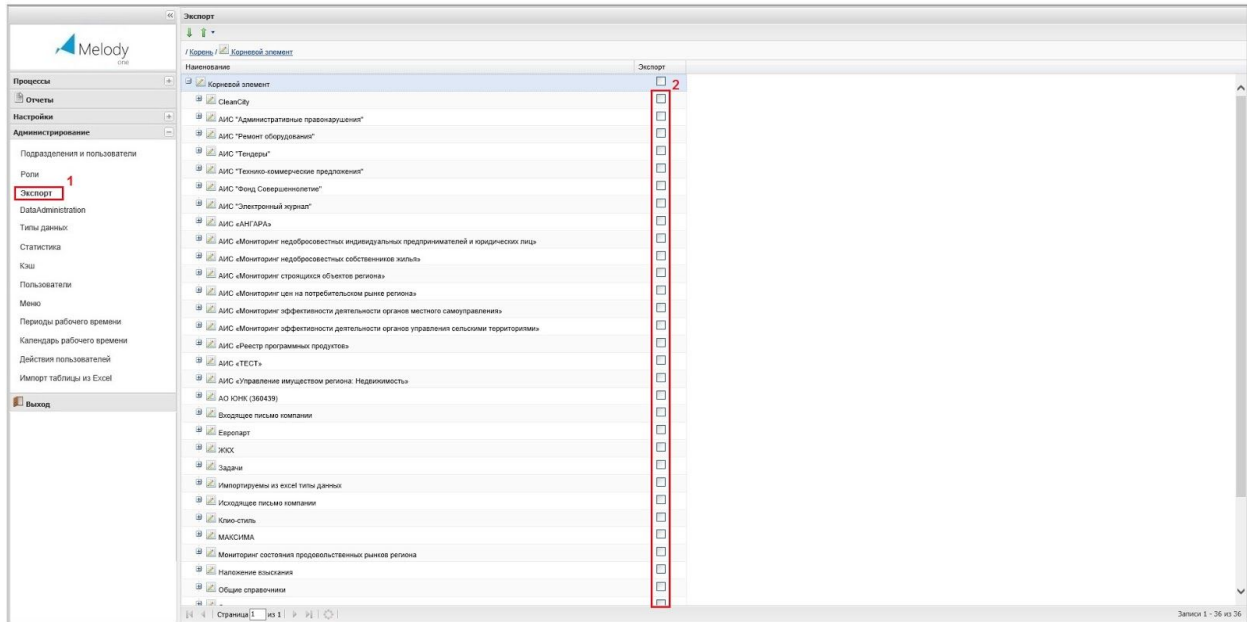


Рис. 3.3.2.3.1. Подпункт "Экспорт"

Можно осуществить выгрузку таблицы из типа данных в таблицу Excel, причем выгрузятся только визуально отображаемые в таблице столбцы.

3.3.2.4. Подпункт "Типы данных"

В подпункте "Типы данных", как представлено на рисунке 3.3.2.4.1. - (1), пользователь может производить администрирование типов данных и таблиц, созданных на основе типов данных. При выборе данного подпункта меню в рабочей области открывается таблица с иерархией типов данных, в которой всегда присутствует корневой элемент - (2), в который, в свою очередь, вложены все остальные элементы, дочерние записи, типы данных, которые относятся к конкретному решению. В корневом элементе всегда присутствует ветка, в которой располагаются системные типы данных.

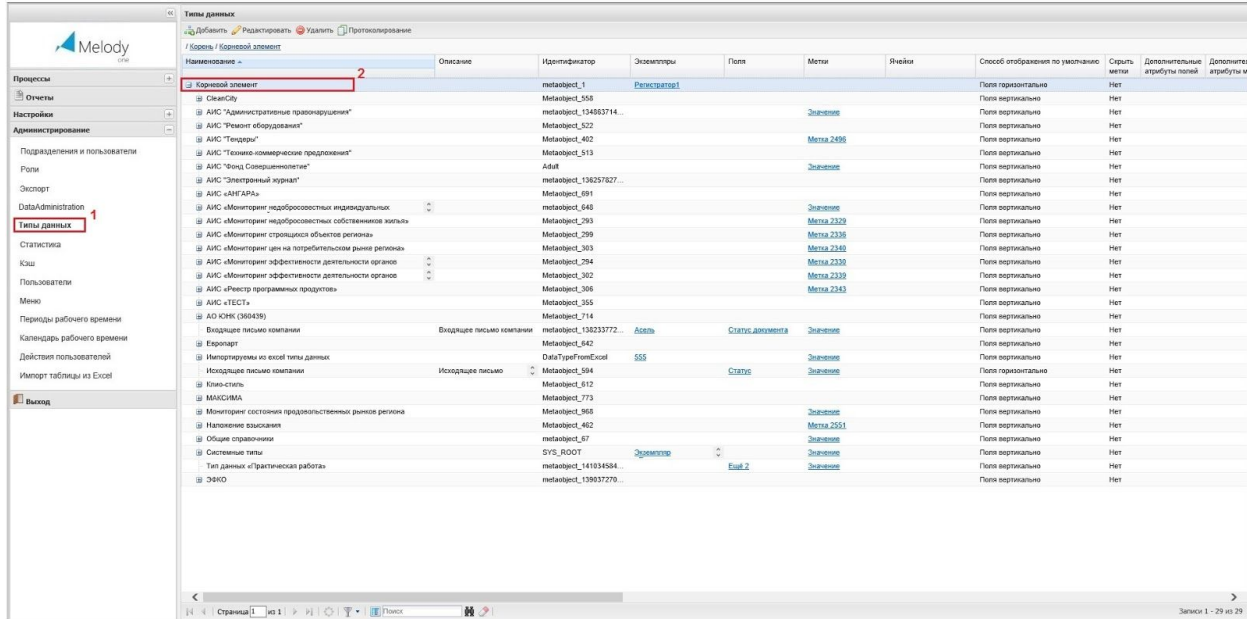


Рис. 3.3.2.4.1. Подпункт "Типы данных"

3.3.2.5. Подпункт "Статистика"

Подпункт меню системы "Статистика" дает возможность получить информацию о работе системы, просмотреть статистику работы приложения: блокировки, состояния памяти, дампы процессов, срабатывание планировщиков, состояние таблиц СУБД и другую системную информацию.

3.3.2.6. Подпункт "Кэш"

Подпункт меню "Кэш", как показано на рисунке 3.3.2.6.1. - (1), позволяет управлять определенной областью памяти (буфером быстрого доступа), в которой отражаются наиболее востребованные объекты. После выполнения серьезных настроек может потребоваться выполнить операцию "Очистить кэш" - (2), кнопка запуска которой расположена слева внизу рабочей области. Справа внизу рабочей области располагается кнопка вызова важной операции "Переподключиться к базе" (3). При этом пользоваться этой операцией нужно очень осторожно, выполняя ее в исключительных случаях, например, при проведении настроек полей данных в рабочей (эксплуатирующейся) системе. Рядом с кнопкой "Переподключиться к базе" находится кнопка "Запустить кэширование метаобъектов" - (4), предназначенная для кэширования типов данных из базы в оперативную память системы. Здесь же размещается кнопка "Запустить кэширование объектов" - (5), при этом выбор данных, для кэширования определяется в настройках экземпляров (sheet). Кроме того, может присутствовать кнопка "Запустить сборку мусора", которая вызывает системную сборку мусора java (garbage collector). В

новых версиях появилась кнопка: "Переиндексировать все", с помощью которой запускается переиндексация в Iucene всех объектов системы. Как правило, данная операция используется после сбоев.

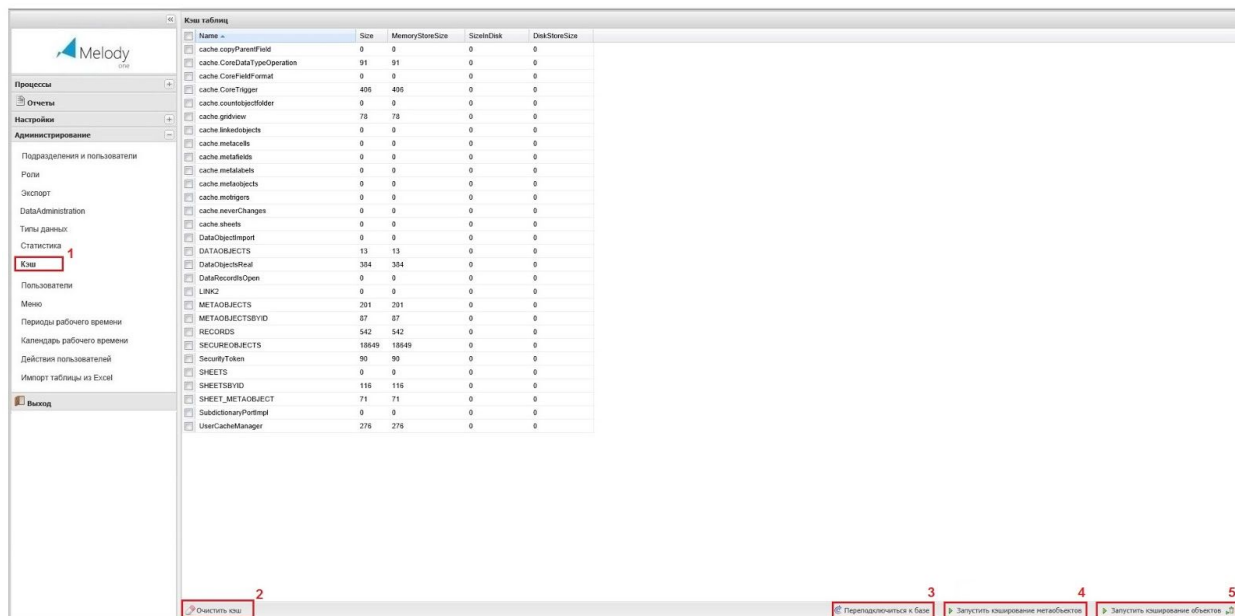


Рис. 3.3.2.6.1. Подпункт "Кэш"

3.3.2.7. Подпункт "Пользователи"

Подпункт меню "Пользователи", как изображено на рисунке 3.3.2.7.1. - (1), необходим для управления учетными записями пользователей системы. Здесь же можно добавить нового пользователя (2). Кроме того, управлять учетными записями можно и из подпункта "Подразделения и пользователи", в котором учетные записи сгруппированы в соответствии с принадлежностью пользователей к конкретным подразделениям (организационной структурой). Подробно управление учетными записями описывается в разделе 5.1. Управление учетными записями.

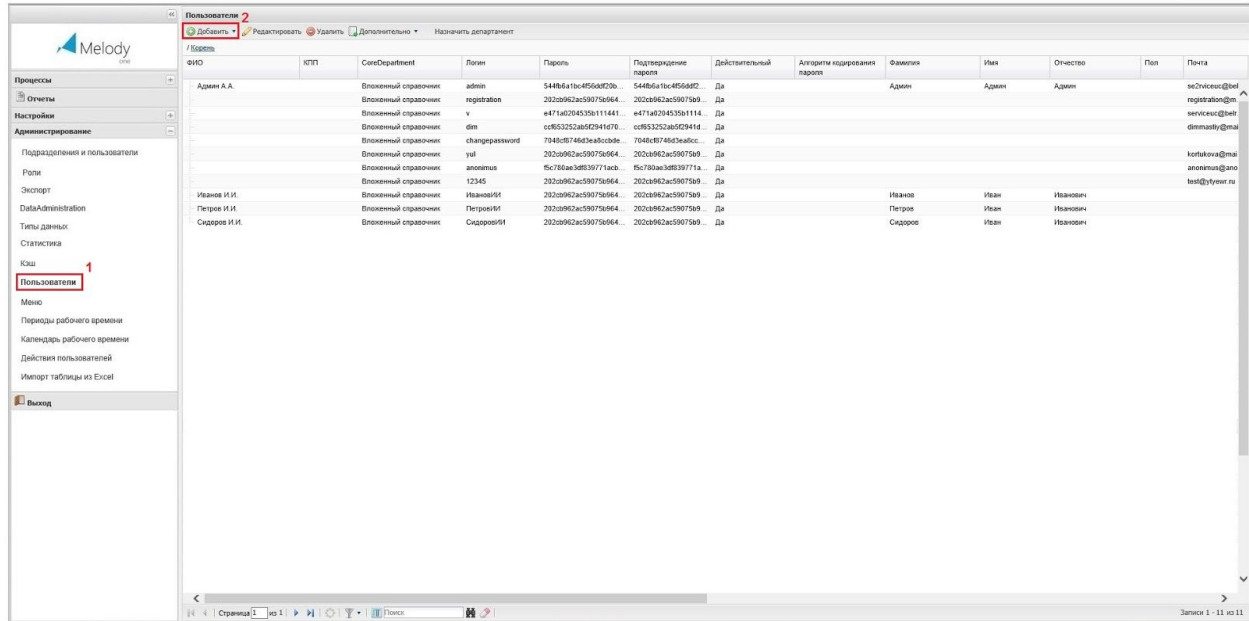


Рис. 3.3.2.7.1. Подпункт "Пользователи"

3.3.2.8. Подпункт "Меню"

Подпункт "Меню", как представлено на рисунке 3.3.2.8.1. - (1), позволяет получить доступ к полному списку подпунктов меню в левом окне, здесь же администратор системы имеет возможность добавить новые пункты меню (2), настроить права доступа к пунктам меню для всех ролей. Подробно добавление пунктов в меню рассмотрено в разделе 5.5. Настройка пунктов меню.

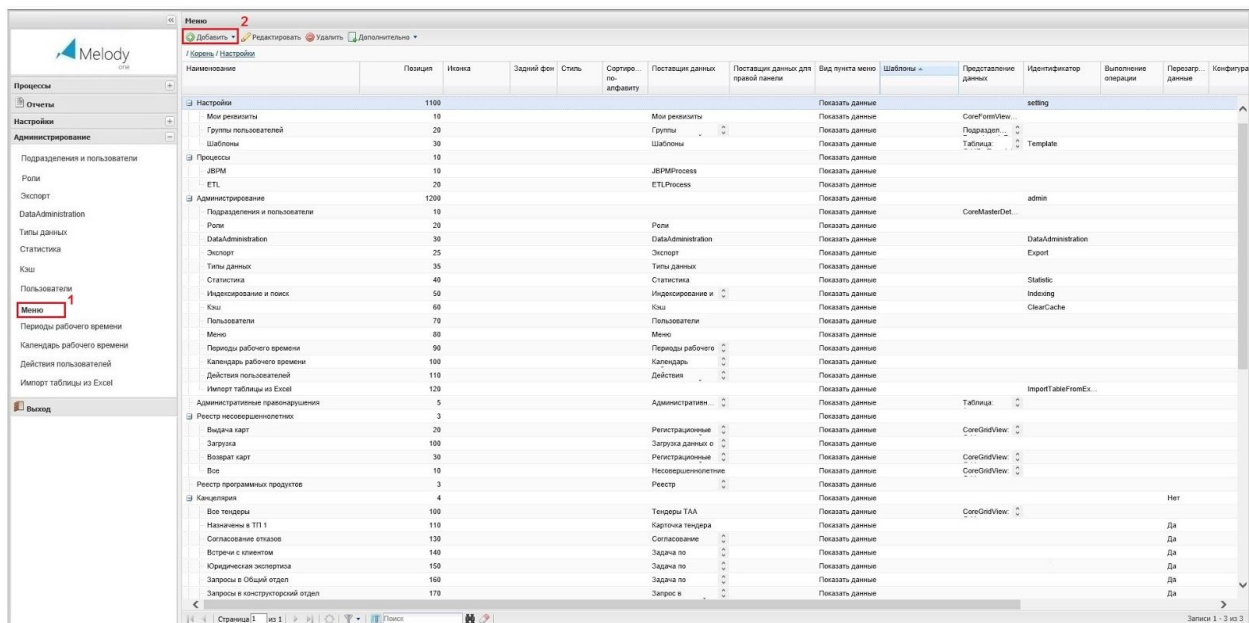


Рис. 3.3.2.8.1. Подпункт "Меню"

3.3.2.9. Подпункт "Периоды рабочего времени"

Подпункт "Периоды рабочего времени", как показано на рисунке 3.3.2.9.1 - (1), позволяет задать такие значения как: рабочая, нерабочая и календарная минута, рабочий, нерабочий и календарный час, рабочий, нерабочий и календарный день, календарная неделя. В этом подпункте можно задать и новый период (2). Эти данные могут потребоваться для расчетов времени в календарях (см. 3.3.2.10. Подпункт "Календарь рабочего времени").

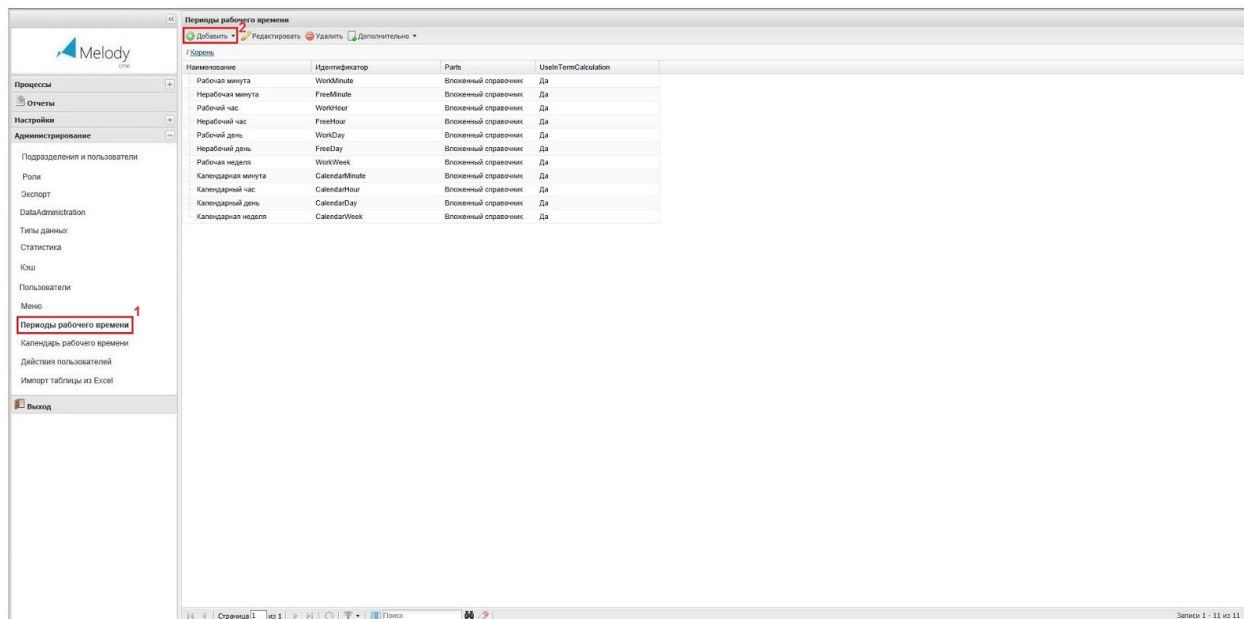


Рис. 3.3.2.9.1. Подпункт "Периоды рабочего времени"

3.3.2.10. Подпункт "Календарь рабочего времени"

Подпункт "Календарь рабочего времени", как отображено на рисунке 3.3.2.10.1. - (1), используется для настройки календарей рабочего времени (производственных календарей).

В системе, настроенной для определенного предприятия, возможно задание своего календаря. Причем, календарей может быть несколько, но в конкретном поле может использоваться только один календарь.

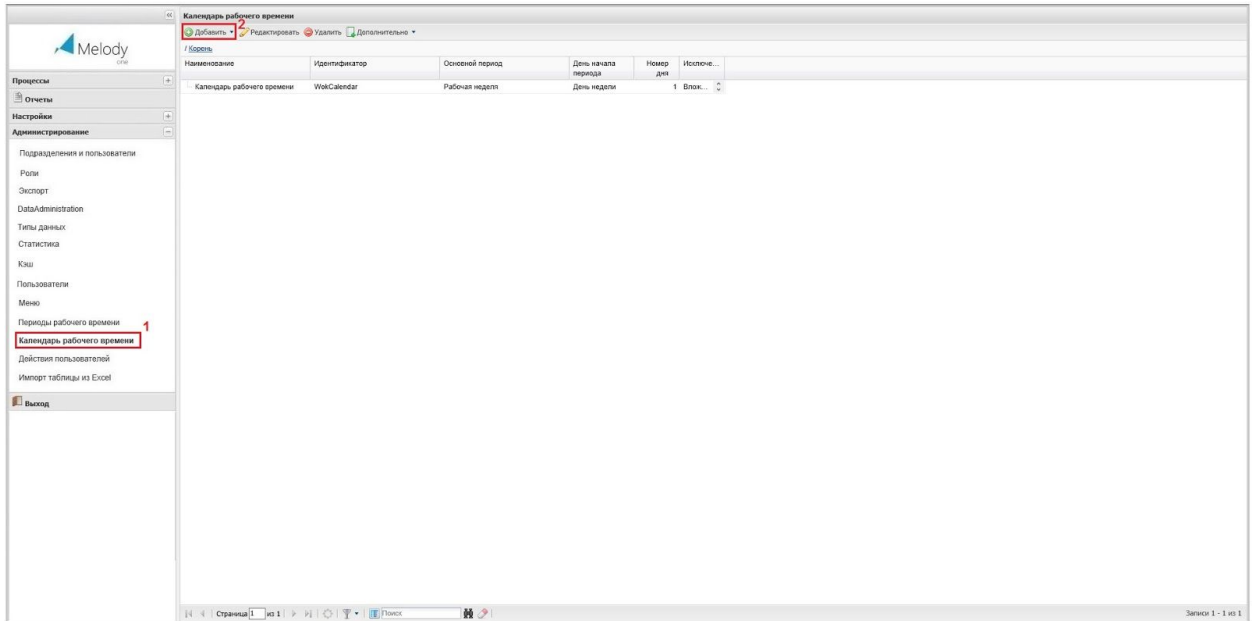


Рис. 3.3.2.10.1. Подпункт "Календарь рабочего времени"

В этом же пункте можно добавить новый календарь - (2).

При добавлении нового календаря откроется форма, показанная на рисунке 3.3.2.10.1. В этой форме необходимо задать наименование создаваемого календаря - (1), указать идентификатор - (2), выбрать период из списка - (3), назначить день начала периода - (4) и установить номер дня - (5). После выполнения этих действий. необходимо определить перечень исключений, которыми являются праздничные и выходные дни в форме - (6).

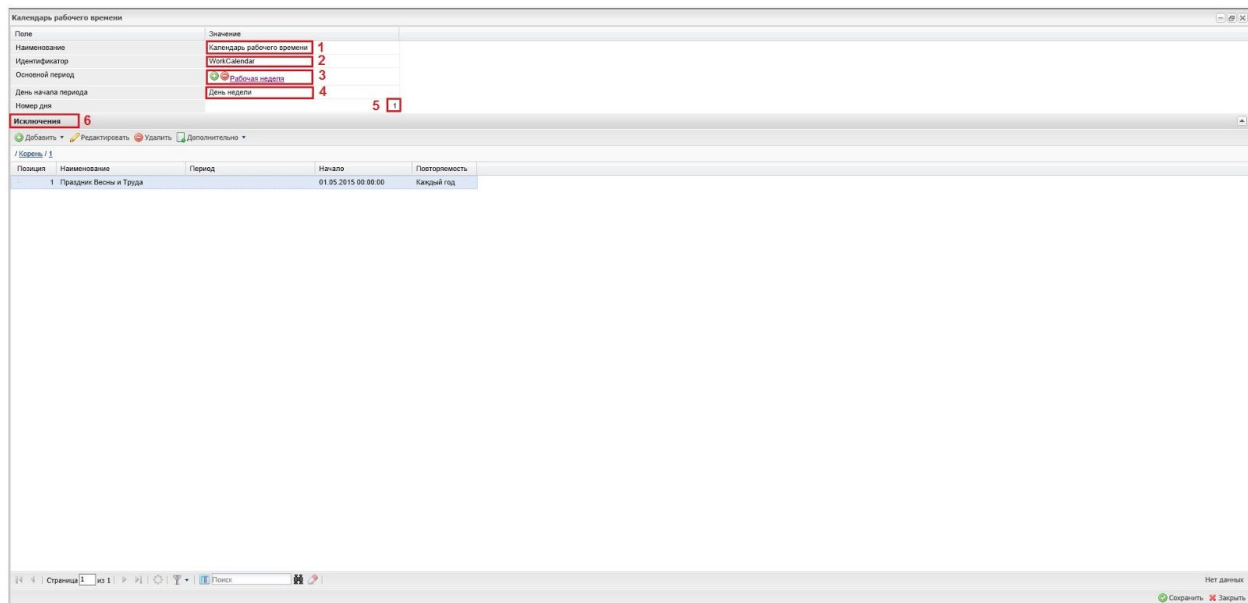


Рис. 3.3.2.10.2. Добавление календаря

После выполнения всех действий (создание или редактирование) с календарем необходимо сохранить сделанные изменения.

3.3.2.11. Подпункт "Действия пользователей"

Подпункт "Действия пользователей", как представлено на рисунке 3.3.2.11.1. - (1), предназначен для фиксирования (логирования) общих действий пользователей (открытия, запуск, нажатие кнопок). Эта информация отображается в журнале действий в табличном виде. Детальные действия над объектом хранятся в системной таблице CoreLog и привязаны к типу данных.

Время действия	Логин	Событие	Фамилия	Имя	Отчество	IP	ID session	Заголовок запроса
25.12.2014 16:17:51	adm	Вход в систему				37.208.65.153	C0565C2D3AE8A81B131AD05968...	host = partnermeb...
25.12.2014 16:17:47	adm	Авторизован	Админ	Админ	Админ	37.208.65.153	C0565C2D3AE8A81B131AD05968...	host = partnermeb...
17.12.2014 17:53:50	adm	Выход из системы	Админ	Админ	Админ	37.208.65.153	E7595FF69196C391838F88293719...	host = partnermeb...
17.12.2014 16:48:59	adm	Вход в систему	Админ	Админ	Админ	37.208.65.153	E7595FF69196C391838F88293719...	host = partnermeb...
16.12.2014 22:30:00	adm	Выход из системы	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 18:25:00	adm	Выход из системы	Админ	Админ	Админ	37.208.65.153	E7595FF69196C391838F88293719...	host = partnermeb...
16.12.2014 17:38:34	adm	Открытие записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:38:13	adm	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:30:44	adm	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:34:11	adm	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:33:27	adm	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:32:59	adm	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:32:48	adm	Открытие записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:32:38	adm	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:31:56	adm	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:31:22	adm	Изменение записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:30:52	adm	Открытие записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:30:09	adm	Открытие записи	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:27:45	adm	Вход в систему	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
16.12.2014 17:27:41	adm	Авторизован	Админ	Админ	Админ	109.62.163.74	A02381B55883D0746C2AD8BC58...	host = partnermeb...
15.12.2014 16:24:18	adm	Открытие записи	Админ	Админ	Админ	37.208.65.153	E7595FF69196C391838F88293719...	host = partnermeb...
15.12.2014 16:24:07	adm	Открытие записи	Админ	Админ	Админ	37.208.65.153	E7595FF69196C391838F88293719...	host = partnermeb...
15.12.2014 16:22:50	adm	Вход в систему	Админ	Админ	Админ	37.208.65.153	E7595FF69196C391838F88293719...	host = partnermeb...
15.12.2014 16:22:46	adm	Авторизован	Админ	Админ	Админ	37.208.65.153	E7595FF69196C391838F88293719...	host = partnermeb...
10.12.2014 12:46:60	adm	Выход из системы	Админ	Админ	Админ	37.208.65.153	B15980B651AA0E9A58D047A1507D...	host = partnermeb...
09.12.2014 13:12:00	adm	Выход из системы	Админ	Админ	Админ	37.208.65.153	79A7B889E3957558F9F187AE7FF...	host = partnermeb...
08.12.2014 11:09:57	adm	Вход в систему	Админ	Админ	Админ	37.208.65.153	79A7B889E3957558F9F187AE7FF...	host = partnermeb...
08.12.2014 10:22:10	adm	Выход из системы	Админ	Админ	Админ	37.208.65.153	79A7B889E3957558F9F187AE7FF...	host = partnermeb...
08.12.2014 18:23:33	adm	Открытие записи	Админ	Админ	Админ	37.208.65.153	B15980B651AA0E9A58D047A1507D...	host = partnermeb...
08.12.2014 18:22:40	adm	Изменение записи	Админ	Админ	Админ	37.208.65.153	B15980B651AA0E9A58D047A1507D...	host = partnermeb...
08.12.2014 18:22:19	adm	Изменение записи	Админ	Админ	Админ	37.208.65.153	B15980B651AA0E9A58D047A1507D...	host = partnermeb...
08.12.2014 18:22:40	adm	Изменение записи	Админ	Админ	Админ	37.208.65.153	B15980B651AA0E9A58D047A1507D...	host = partnermeb...
08.12.2014 18:22:19	adm	Изменение записи	Админ	Админ	Админ	37.208.65.153	B15980B651AA0E9A58D047A1507D...	host = partnermeb...
08.12.2014 18:21:24	adm	Открытие записи	Админ	Админ	Админ	37.208.65.153	B15980B651AA0E9A58D047A1507D...	host = partnermeb...
08.12.2014 18:21:21	adm	Открытие записи	Админ	Админ	Админ	37.208.65.153	B15980B651AA0E9A58D047A1507D...	host = partnermeb...
08.12.2014 18:20:55	adm	Открытие записи	Админ	Админ	Админ	37.208.65.153	B15980B651AA0E9A58D047A1507D...	host = partnermeb...
08.12.2014 18:22:04	adm	Открытие записи	Админ	Админ	Админ	37.208.65.153	B15980B651AA0E9A58D047A1507D...	host = partnermeb...

Рис. 3.3.2.11.1. Подпункт "Действия пользователей"

3.3.2.12. Подпункт "Импорт таблицы из Excel"

Подпункт "Импорт таблицы из Excel", как показано на рисунке 3.3.2.12.1. - (1), позволяет создавать типы данных на основе электронных excel - таблиц, выбор таблицы производится посредством окна (2) "загрузка файлов".

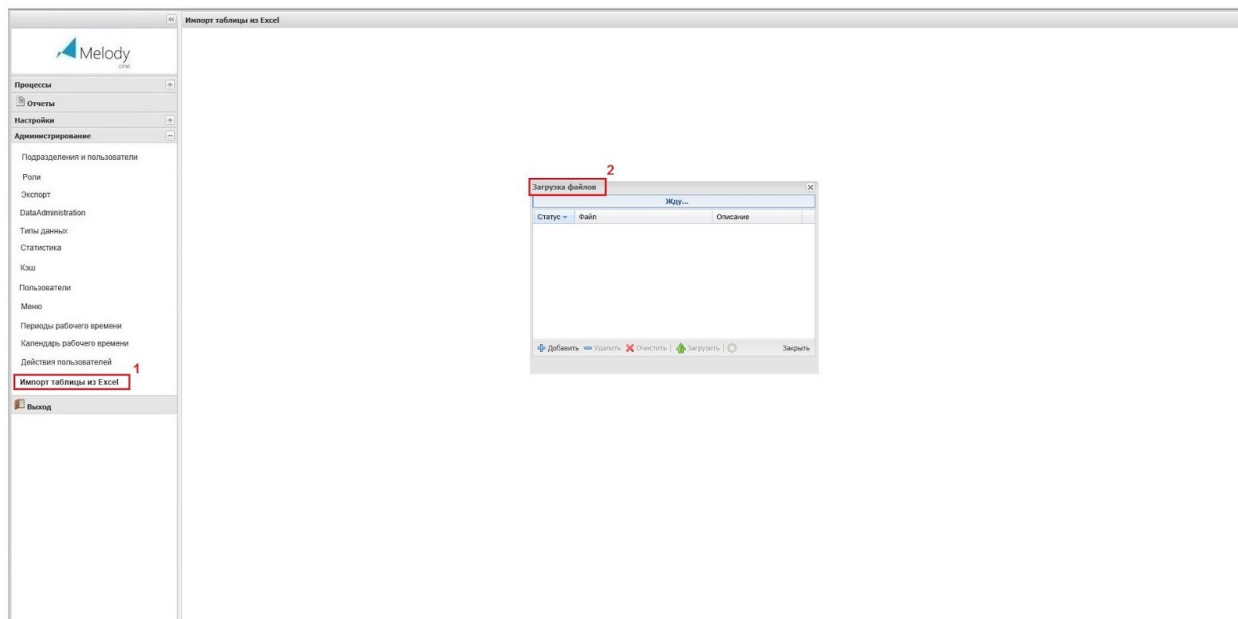


Рис. 3.3.2.12.1. Подпункт "Импорт таблицы из Excel"

После загрузки таблицы откроется форма, показанная на рис. 3.3.2.12.2., структура которой соответствует структуре загруженной таблицы, причем все формулы (если таковые были) в ячейках таблицы - excel сохраняются. Открывшуюся форму можно отредактировать добавив поля - (1) или строки - (2), изменить расположение полей (горизонтально на вертикально, или наоборот) при помощи кнопки "Транспонировать" - (3).

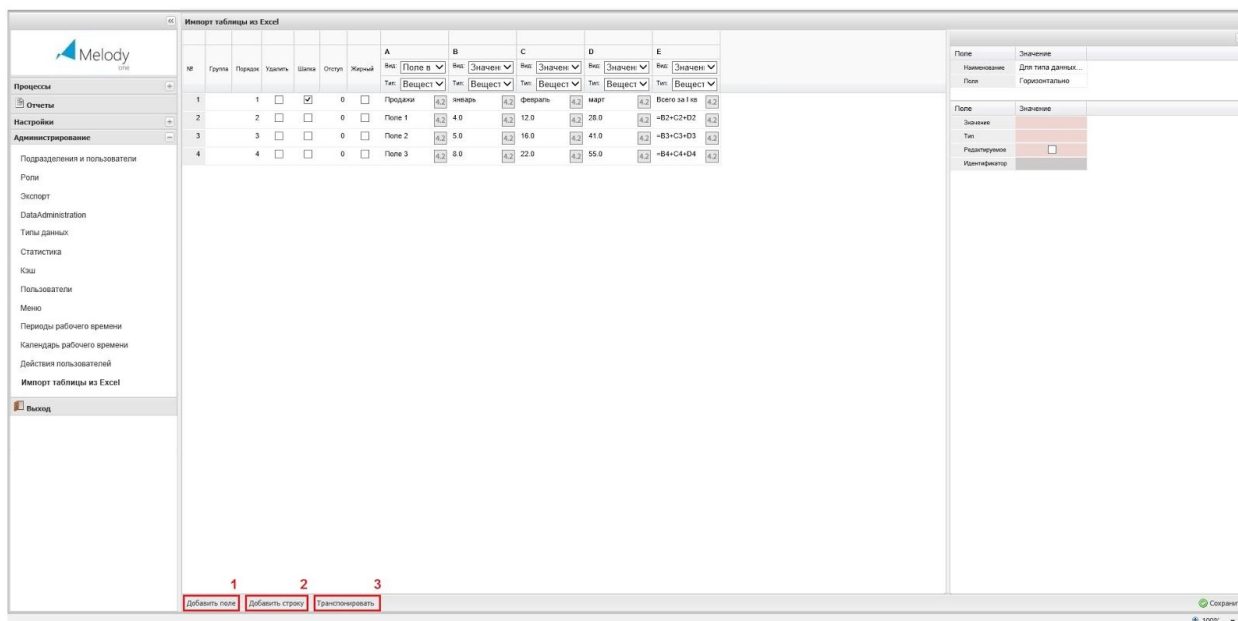


Рис. 3.3.2.12.2. Создание типа данных из импортированной таблицы Excel

По завершению редактирования, создаваемого на основе таблицы Excel, типа данных необходимо сохранить, сделанные изменения. Этот тип данных будет создан в каталоге “Импортируемые из excel типы данных”.

3.3.2.13. Подпункт “Шаблоны”

Подпункт “Шаблоны”, как показано на рисунке 3.3.2.13.1.- (1), предоставляет возможность создавать и сохранять шаблоны пользователя.

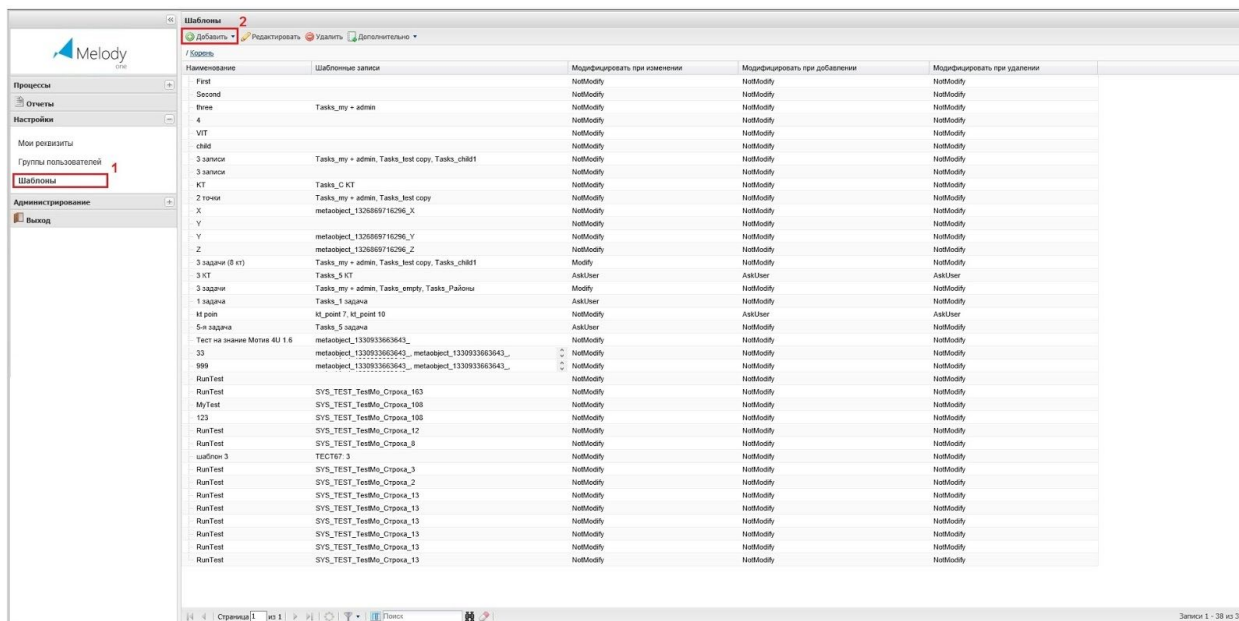


Рис.3.3.2.14.1. Подпункт “Шаблоны”

Доступ к форме, позволяющей добавлять пользовательские шаблоны, производится посредством нажатия на кнопку “добавить” рис. 3.3.2.13.1. - (2), после этого действия открывается форма, которая содержит поля: “Наименование” для внесения названия шаблона пользователя; “Шаблонные записи”, где можно выбрать соответствующий шаблон или создать новый, как и задать способ модификации, а именно: “Модифицировать при изменении”, “Модифицировать при добавлении”, “Модифицировать при удалении”. Каждое из полей, задающих способ модификации, может принимать различные значения, заданные в выпадающем списке (см. рис. 3.3.2.13.2.). В полях типа “Модифицировать при...” определяется действие, которое будет применяться к записям, созданным по шаблону, при изменении/удалении/добавлении записей в соответствующем шаблоне.

Шаблоны/Добавление	
Поле	Значение
Наименование	
Модифицировать при изменении	NotModify x v
Модифицировать при добавлении	Изменять
Модифицировать при удалении	Не изменять
Шаблонные записи	
✎ Редактировать ✖ Удалить	
/ Корень	

Рис.3.3.2.13.2. Добавление шаблона

3.3.2.14. Подпункт "Планировщик"

Данный подпункт меню предназначен для создания и изменения периодически выполняемых операций в системе. Это задачи, выполняемые по расписанию с определенной периодичностью или однократно в определенное время, заданное заранее. К таким операциям относятся: отправка электронной почты, формирование очереди сообщений, сброс счетчиков документов и др.

Для добавления нового планировщика необходимо, находясь в пункте меню, нажать кнопку «Добавить» на верхней панели. При этом откроется форма, состоящая из двух вкладок – «Общие» и «Настройка». Вкладка «Общие», как показано на рисунке 3.3.2.14.1., предназначена для добавления общей информации о планировщике.

Планировщик задач/Добавление	
Общие Настройка	
Наименование	Значение
Наименование 1	
Описание 2	
Процесс 3	
Активная 4	<input type="checkbox"/>
Запускать только один экземпляр 5	<input type="checkbox"/>
Приложение 6	

Выбор записей произвольного типа

- AbstractProcess
- ETLProcess
- CounterProcess
- JBPMProcess
- ReportProcess
- CustomProcess
- ActivitiProcess

Рис. 3.3.2.14.1. Вкладка "Общее" (Планировщик добавление)

В поле «Наименование» - (1) вводится имя планировщика, под которым он будет храниться в списке и доступен для выбора. Поле «Описание» - (2) служит для пояснения назначения планировщика. Поле «Процесс» - (3) позволяет выбрать выполняемый процесс, который будет выполняться при запуске операции, а именно:

- ETLProcess – (от англ. Extract, Transform, Load) – группировка охватывает процессы извлечения, трансформации (преобразования) и загрузки данных;
- CounterProcess – группировка содержит список процессов обработки счетчиков. Указывается идентификатор счетчика, который будет обрабатываться процессом. Идентификатор, как правило, заполняется автоматически при создании нового значения экземпляра CoreCounter;
- JBPMProcess – группировка включает в себе процессы реализации потоков (workflow);
- RepotrProcess – группировка включает в себя список процессов формирования отчетов по расписанию;
- CustomProcess – группировка содержит обычные (простые) процессы. В качестве параметров запуска указывается скрипт;
- ActivitiProcess – группировка включает в себе процессы активности.

После выбора на вкладке «Общие» Планировщика требуемого процесса в строке «Активность» - (4) допустимо установить признак активности данного Планировщика. В поле “Запускать только один экземпляр” - (5) можно установить флаг, который позволит применить Планировщик только к одному экземпляру при условии, что экземпляров несколько.

Вкладка «Настройка», как представлено на рисунке 3.3.2.14.2. предназначена для настройки времени выполнения процесса (настройка таймера). В поле «Шаблон» - (1) есть возможность указать один из сохраненных шаблонов выполнения. Поле “Периодичность” - (2) предназначено для задания периодичности (расписание запуска) выполнения операции. Это можно сделать двумя способами или посредством выбора одного значения из выпадающего списка:

- Однократно;
- Ежедневно;
- Еженедельно;
- Ежемесячно.

или выбрав требуемый шаблон в формате “cron”, как показано на рисунке 3.3.2.14.3., здесь необходимо выбрать требуемый шаблон кликнув по нему мышью - (1) и нажать на кнопку “Выбрать” - (2).

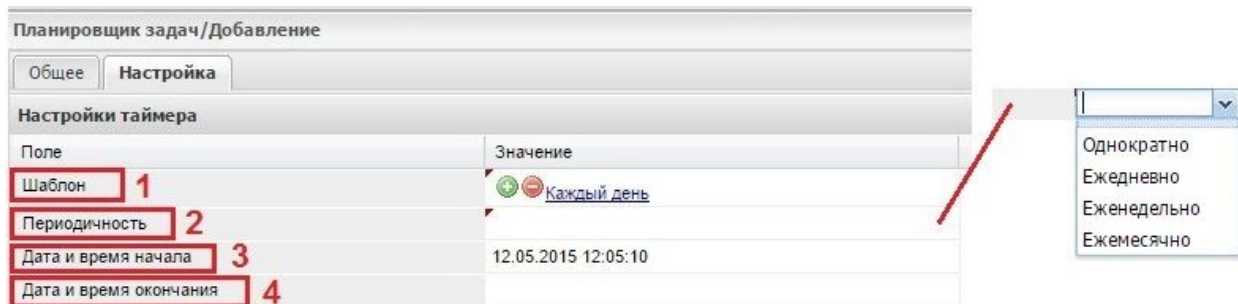


Рис. 3.3.2.14.2. Вкладка “Настройка” (Планировщик добавление)

Поля, как отображено на рисунке 3.3.2.14.2., “Дата и время начала” - (3) и “Дата и время окончания” - (4) определяют промежуток времени выполнения задания Планировщика. Причем если поле “Дата и время окончания пустое” действие будет выполняться бесконечно.

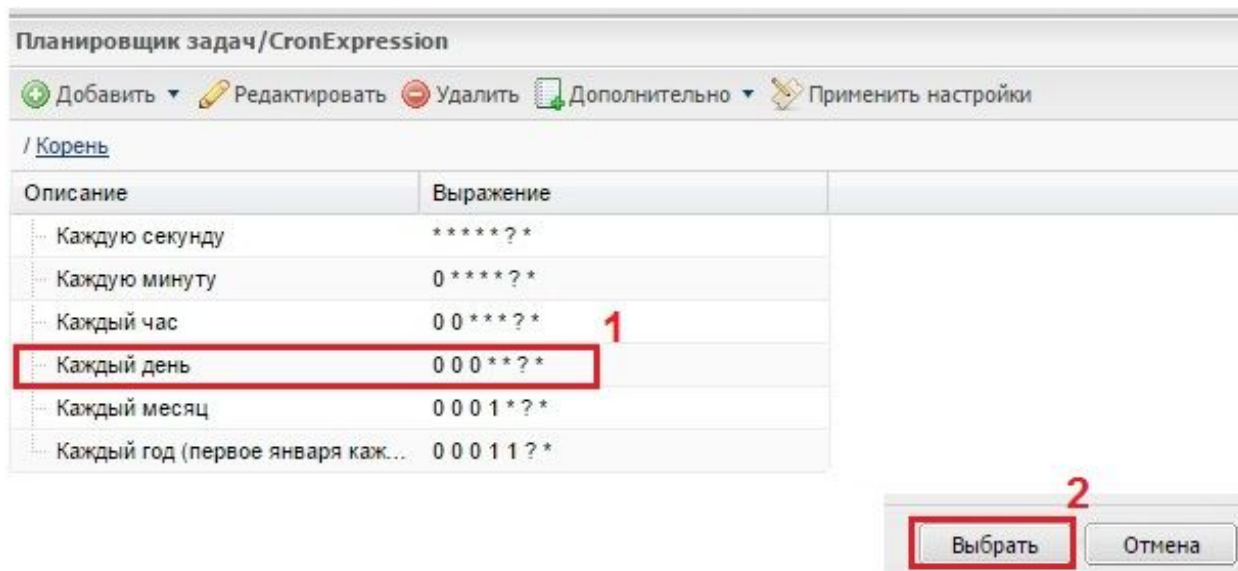


Рис. 3.3.2.14.3. Выбор шаблона в формате “cron”.

При выборе значения “Еженедельно” открывается вкладка с установкой конкретного дня выполнения процесса, как показано на рисунке 3.3.2.14.2. В полях которой можно установить конкретный день недели (понедельник, вторник и т.д.) - (1), так и задать день в соответствии с определенными правилами: каждый 1-й день, каждый 2-й день и т.д. - (9).

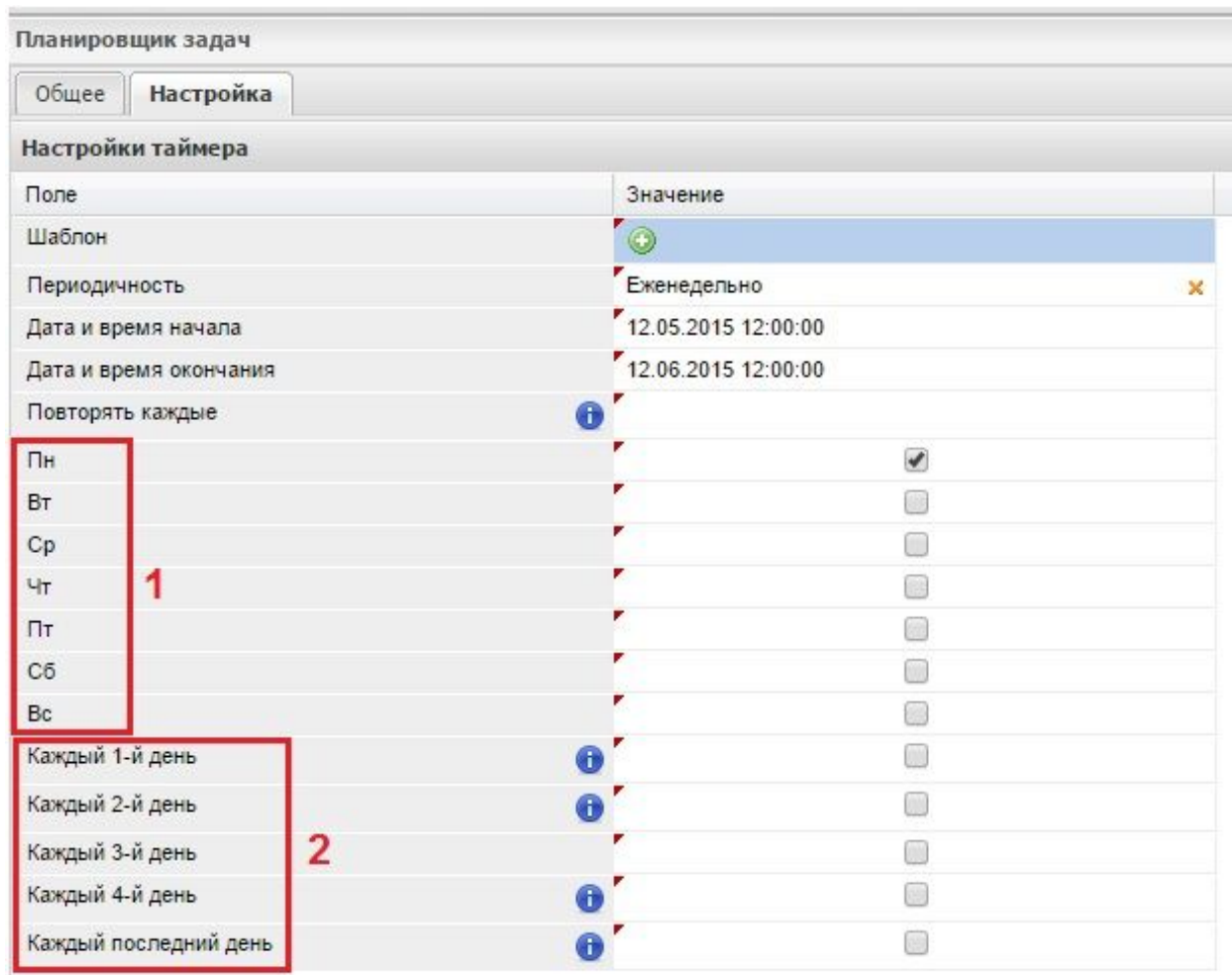


Рис.3.3.2.14.4. Вкладка для настройки еженедельной периодичности

3.3.3. “Пункт “Подразделения, должности и пользователи”

В новых инсталляциях систем, выполненных на платформе Motiware Melody One присутствует пункт меню "Подразделения, должности и пользователи" (вместо подпункта "Подразделения и пользователи"), как показано на рисунке 3.3.3.1. Этот пункт содержит два вертикальных окна “Подразделения” - (1) и “Должности” - (2). В каждом из этих окон можно создать, отредактировать или удалить определенное подразделение или должность.

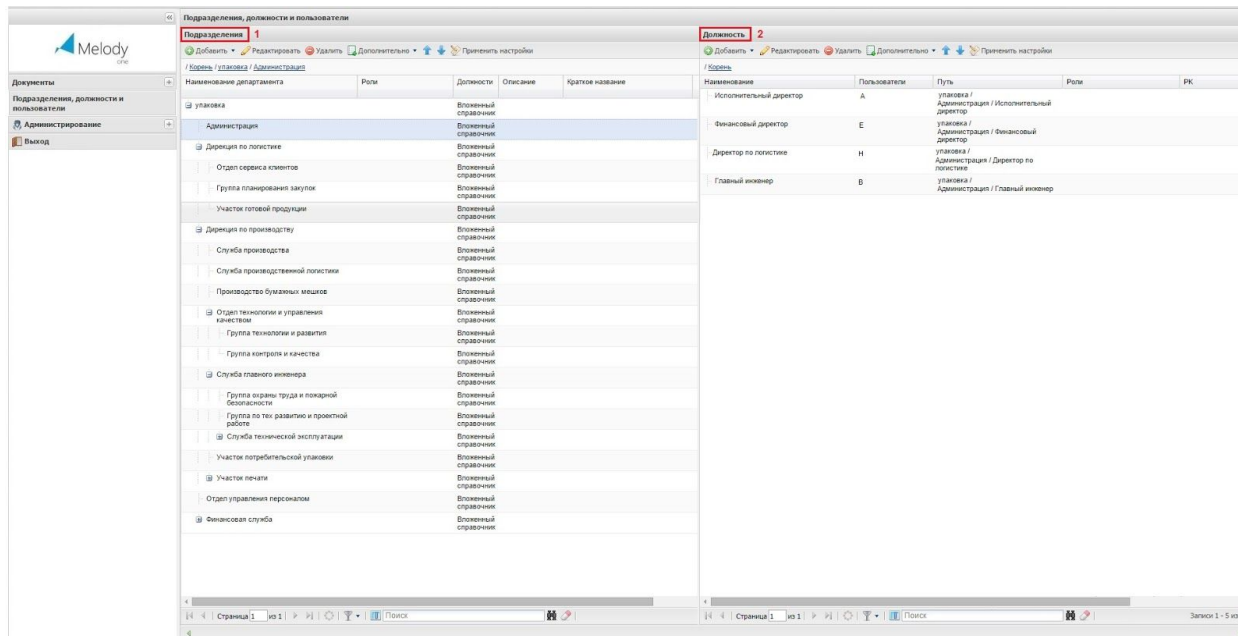


Рис. 3.3.3.1. Пункт “Подразделения, должности и пользователи”

Вкладка “Подразделение/Добавление” (см. рис. 3.3.3.2.) содержит в себе две вкладки “Общее” и “Должности”. На вкладке “Общее” обязательным полем является поле “Наименование департамента” - (1), в котором задается название добавляемого подразделения. В поле “Роли” - (2) можно установить список ролей в которые будут включаться пользователи назначенные в данное подразделение. Поле “Описание” - (3) предназначено для справочной информации. А в поле “Краткое название” - (4), указывается сокращенное наименование добавляемого подразделения. Кроме того, в зависимости от пожеланий заказчика, вкладка “Общее” может содержать ряд других необходимых конкретному заказчику полей - (5).





Подразделение / Добавление	
Общее Должности	
Наименование	Значение
* Наименование департамента	1
Роли	2 
Описание	3 
Краткое название	4
ИНН	5
ОГРН	
Телефон	
Факс	
Email	
Код подразделения	
Адрес юридический	
Адрес почтовый	

Рис. 3.3.3.2. Вкладка “Общее”

Вкладка “Должности” предназначена для добавления новой должности или редактирования (удаления) информации в рамках должностей уже существующих. При нажатии на кнопку “Должности” открывается таблица, для добавления должности аналогичная таблице в окне - (2), см. рис. 3.3.3.1. Здесь присутствуют две вкладки “Общее” (рис. 3.3.3.3.) и “Пользователь”, которая в свою очередь содержит таблицу с кнопками для добавления пользователей, подробно создание и редактирование учетной записи пользователя описано в п. 5.1.2 и в п.5.1.3.


Должность / Добавление	
Общее Пользователь	
Наименование	Значение
Наименование	
Пользователи	
Путь	
Роли	
РК	
Упрощенное (для стат.) название	

Рис. 3.3.3.3. Вкладка “Общее” (Должность/Добавление)

3.3.4. Пункт "Процессы"

Пункт меню "Процессы", изображенный на рисунке 3.3.4.1. - (1), предназначен для выбора и настройки JBPM - (2) и ETL - (3) процессов. Например, процессы могут позволять считывать данные из каких-либо файлов или баз данных, загружать файлы по FTP. Более подробно настройка процессов описана в п. 5.7.

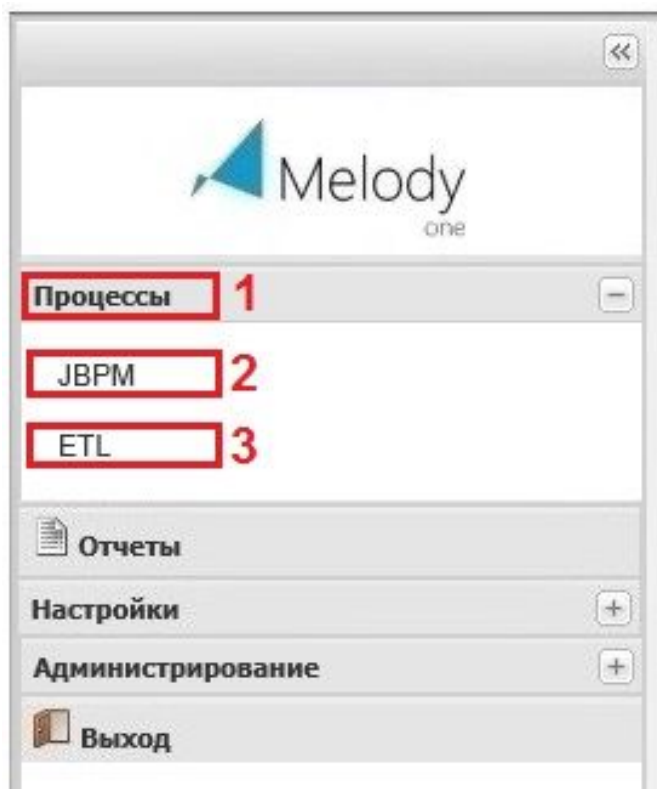


Рис. 3.3.4.1. Пункт "Процессы"

3.3.4.1. Подпункт "JBPM"

В подпункте "JBPM", как показано на рисунке 3.3.4.1.1. - (1), производится настройка процесса (подробнее см. раздел 6.7.), здесь же можно добавить новый процесс - (2).

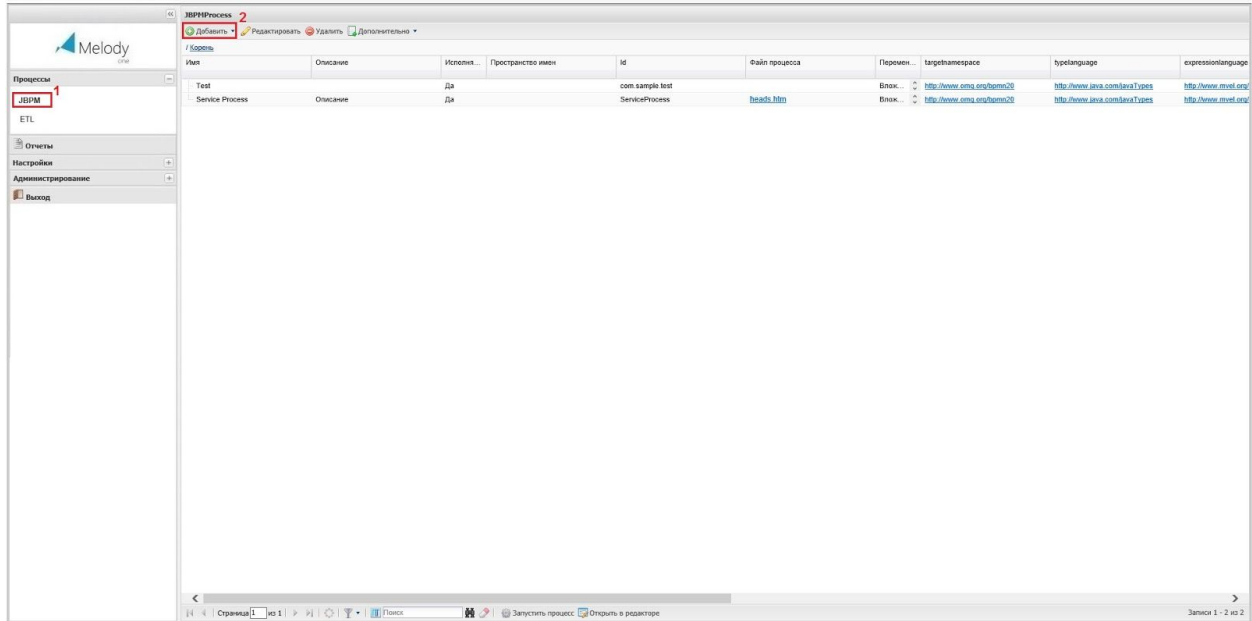


Рис. 3.3.4.1.1. Подпункт "JBPM"

3.3.4.2. Подпункт "ETL"

В подпункте "ETL", как изображено на рисунке 3.3.4.2.1., осуществляется настройка процесса (подробнее см. раздел 5.7.), здесь же можно добавить новый процесс - (2).

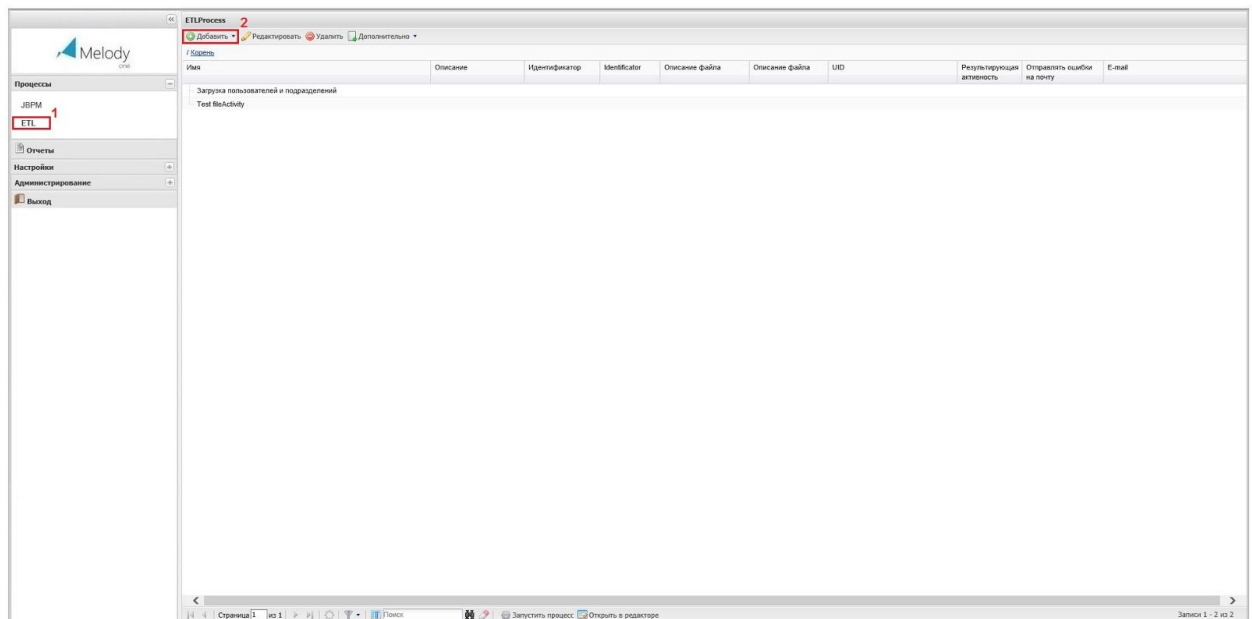


Рис. 3.3.4.2.1. Подпункт "ETL"

3.3.5. Пункт "Отчеты"

Пункт меню "Отчеты", как представлено на рис. 3.3.5.1. - (1), позволяет создать конкретному пользователю новый отчет, отредактировать существующий или запустить процесс выполнения отчета - (2).

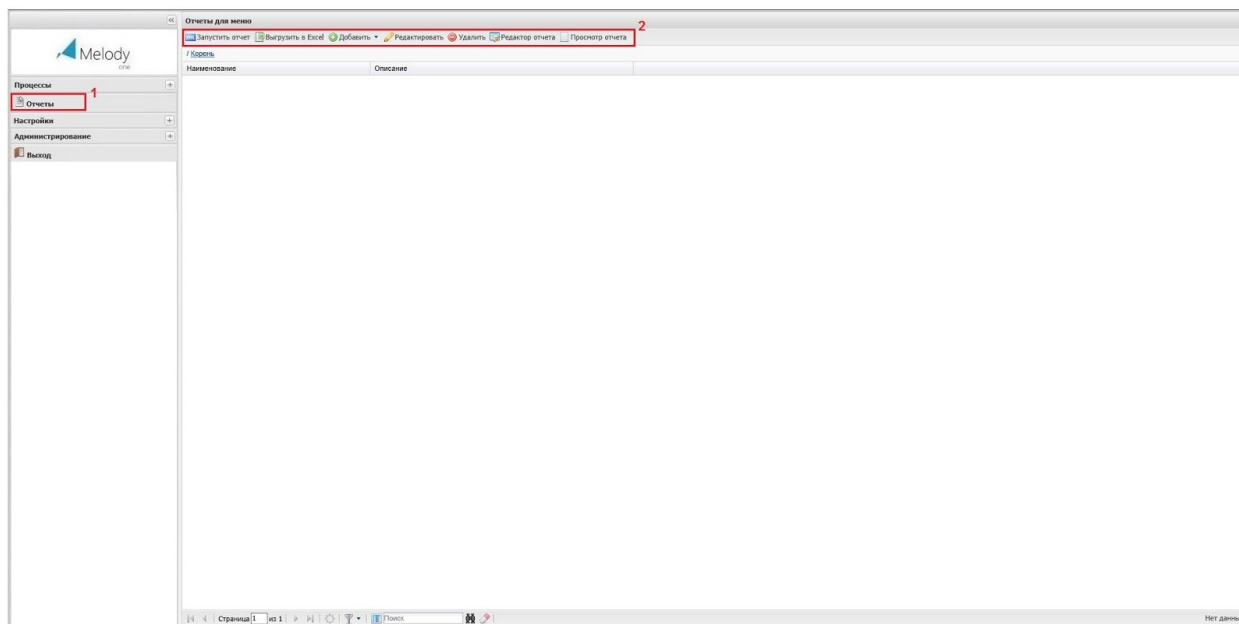


Рис. 3.3.5.1. Пункт "Отчеты"

При добавлении нового отчета (нажата кнопка "добавить"), открывается окно с вкладками:

- Общее;
- Права доступа;
- Получение данных;
- Отчет.

Вкладка "Общее" предназначена для заполнения общей информации о создаваемом отчёте. Вкладка "Права доступа" служит для назначения пользователям прав на создание, редактирование и удаление отчёта. Во вкладке "Получение данных" определяется источник данных для отчёта. Вкладка "Отчет" предназначена для определения выходных параметров отчета. Подробно работа с отчетами описана в п. 5.6.

3.3.6. Кнопка "Выход"

В самом низу главного меню расположена кнопка "Выход" для выхода из системы (см. рис. 3.3.6.1.).

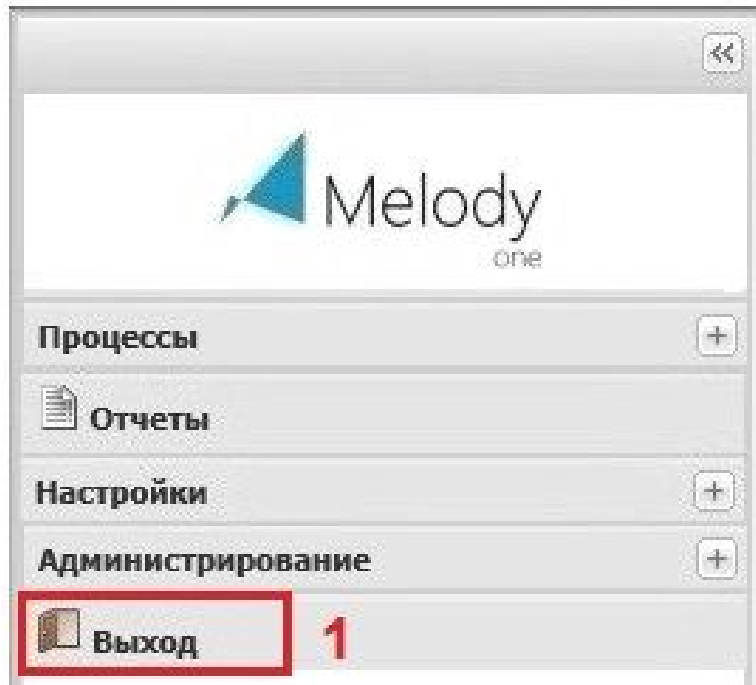


Рис. 3.3.6.1. Кнопка выход

3.4. Пункты пользователей

Платформа Motiware Melody One располагает широкими возможностями для создания любых пунктов меню (см. рис. 3.4.1.), необходимых пользователю (1). При этом все пункты и подпункты меню, включая системные, можно гибко настраивать, формируя наиболее подходящий набор для каждого конкретного случая. Более подробно настройка меню рассматривается в п. 6.5. Настройка пунктов меню.

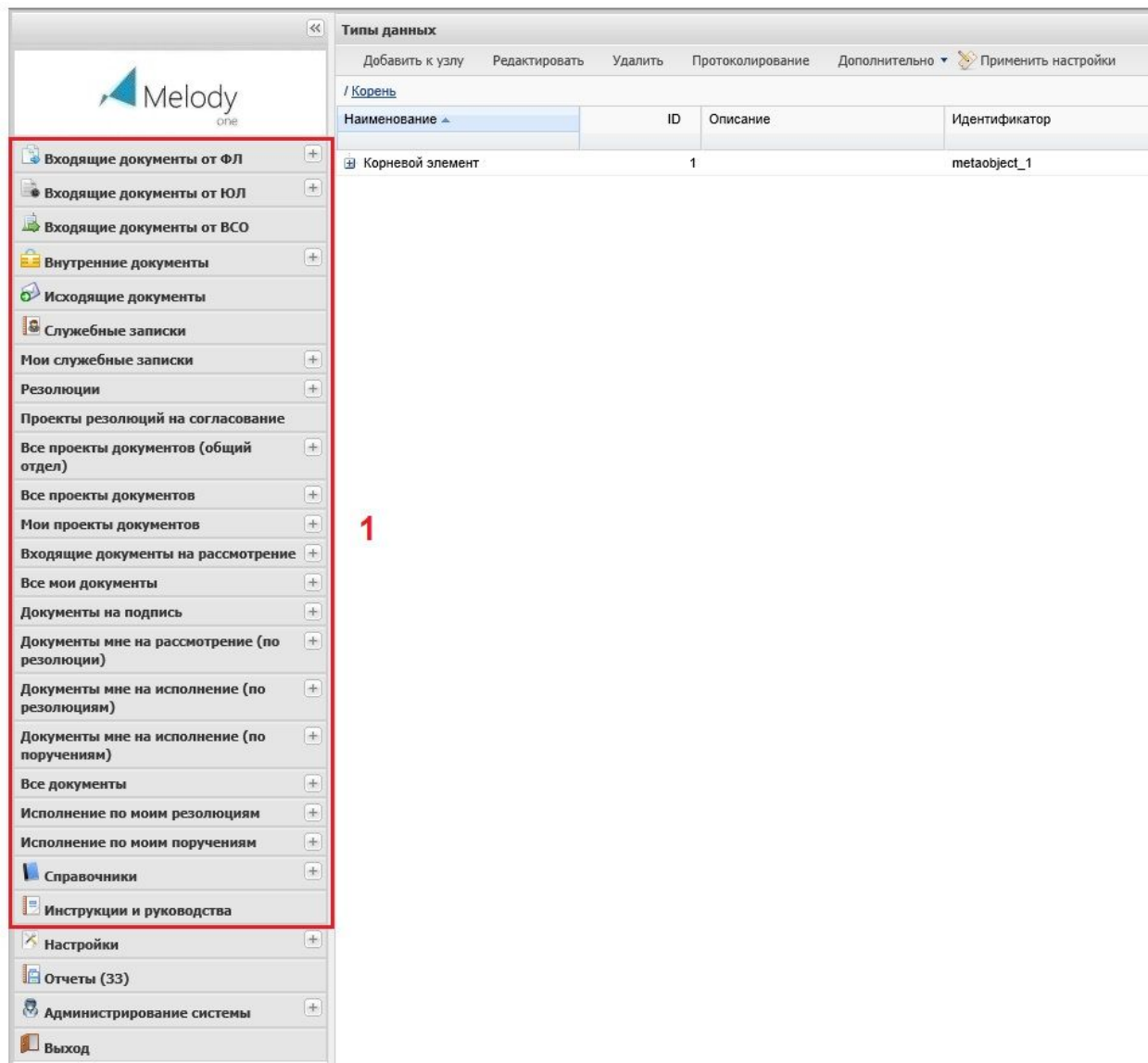


Рис. 3.4.1. Пример пользовательского меню

3.5. Рабочая область

Главным элементом пользовательского интерфейса платформы Motiware Melody One является рабочая область, как показано на рисунке 3.5.1. - (1), которая занимает основную часть экрана и представляет собой область, ограниченную прямоугольником, располагающуюся справа от главного меню - (2). Вид рабочей области определяется выбором пунктов или подпунктов главного меню.

Как правило, в верхней части рабочей области размещаются кнопки - (3) для вызова системных операций.

Системные операции - это операции, которые настраиваются и выполняют действия заданные явно, причем поменять логику для них нельзя. К ним относятся операции:

- Добавить к узлу
- Создать по шаблону
- Добавить (объект)
- Создать по шаблону к узлу
- Редактировать (форму, таблицу, запись)
- Удалить (таблицу, файл)

И дополнительно:

- Копировать (объект)
- Вставить (объект)
- Импортировать (объект)
- Экспортировать (объект)
- Добавить шаблон
- Переместить вверх
- Переместить вниз
- Протоколирование
- Применить настройки по-умолчанию
- On-line просмотр
- On-line просмотр родителя

Системные операции располагаются: Корень / Корневой элемент / Системные типы / Forms / Системные операции.

В некоторых случаях к системным кнопкам добавляются кнопки пользовательские.

В нижней части располагаются: счетчик страниц, панель навигации, кнопки обновления, фильтрации - (4), сохранения и закрытия окна.

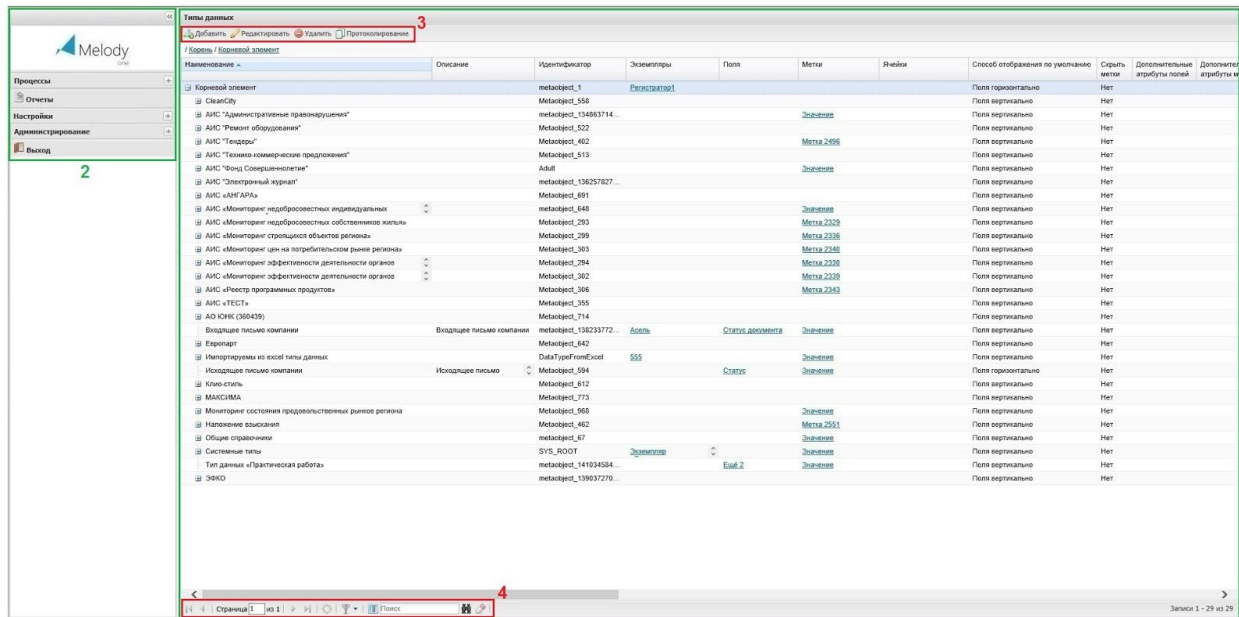


Рис. 3.5.1. Рабочая область платформы Motiware Melody One

В центре рабочей области отображается пользовательская информация. Если в пункте меню задано табличное представление, то в центре рабочей области выводится таблица с записями, вверху которой находятся заголовки столбцов (полей), а в столбцах выводятся значения пользовательских данных.

В рабочей области пользовательская информация может отображаться в соответствии с настройкой представления данных для таблицы, например в виде дерева каталога (с корневым элементом), которое содержит в себе иерархию каталогов, вложенных в корневой каталог (иерархию системы).

В левом нижнем углу рабочей области располагаются кнопки для перехода по страницам вперед и назад, к первой и к последней странице, счетчик страниц с окном выбора конкретной страницы, кнопка обновления, кнопка вызова фильтра, кнопка расширенного поиска, панель навигации, кнопка запуска поиска, кнопка «очистить».

Если значения в рабочей области отображаются в виде таблицы, этим отображением можно управлять, а именно, настраивать отображение так, как будет удобно конкретному пользователю. По всем полям возможна сортировка (см. рис. 3.5.2.) как по возрастанию значений, так и по убыванию - (1). Выполнить сортировку можно выбрав тип сортировки в выпадающем списке, кликнув мышкой справа от наименования конкретного столбца или щелкнув мышью по названию столбца. Один щелчок - сортировка в прямом направлении, повторный щелчок - сортировка в другом обратном направлении. Кроме того, в выпадающем списке предусмотрена возможность выделить

для отображения в таблице те или иные столбцы. Для этого необходимо выбрать поле "столбцы" и в выпавшем списке столбцов поставить или убрать отметку (в виде "галки") у соответствующего столбца - (2). Кроме того, порядок полей в таблице можно менять захватив и перетаскивая соответствующий столбец (поле) при помощи мыши.

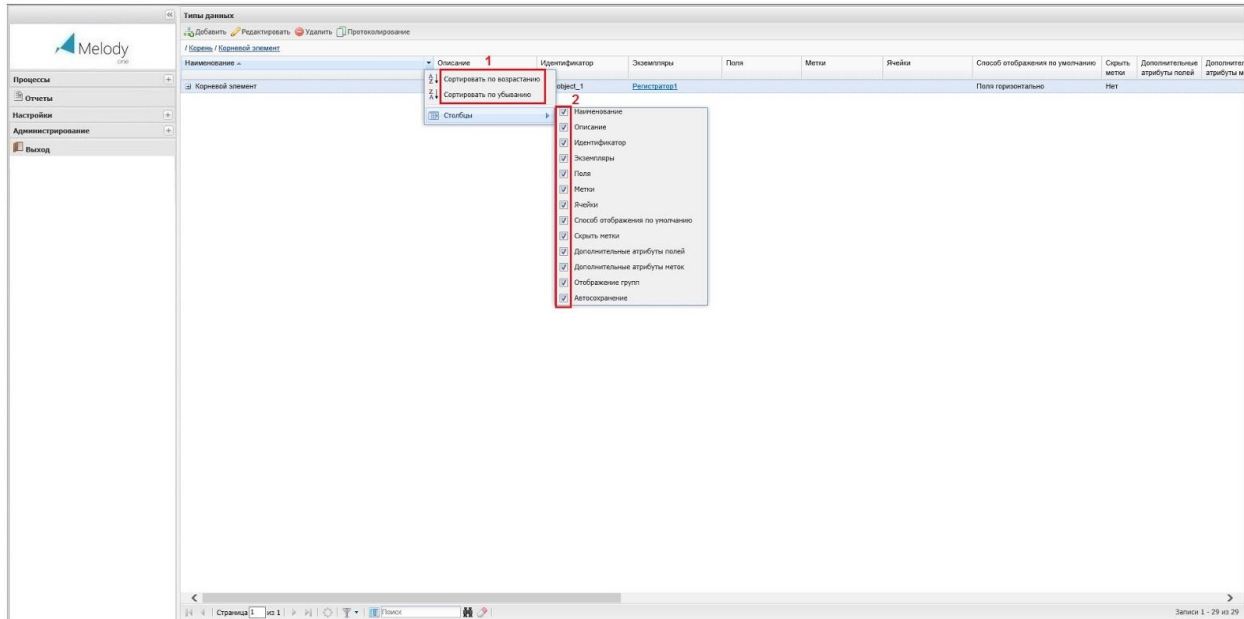


Рис. 3.5.2. Отображение таблиц в рабочей области

3.5.1. Кнопки для вызова системных операций

Для осуществления системных операций используются кнопки (см. рис. 3.5.1.1.), которые могут располагаться отдельно - (1), а могут размещаться в окне выпадающего списка - (2) или объединяться в группы по несколько кнопок горизонтально на панели - (3).

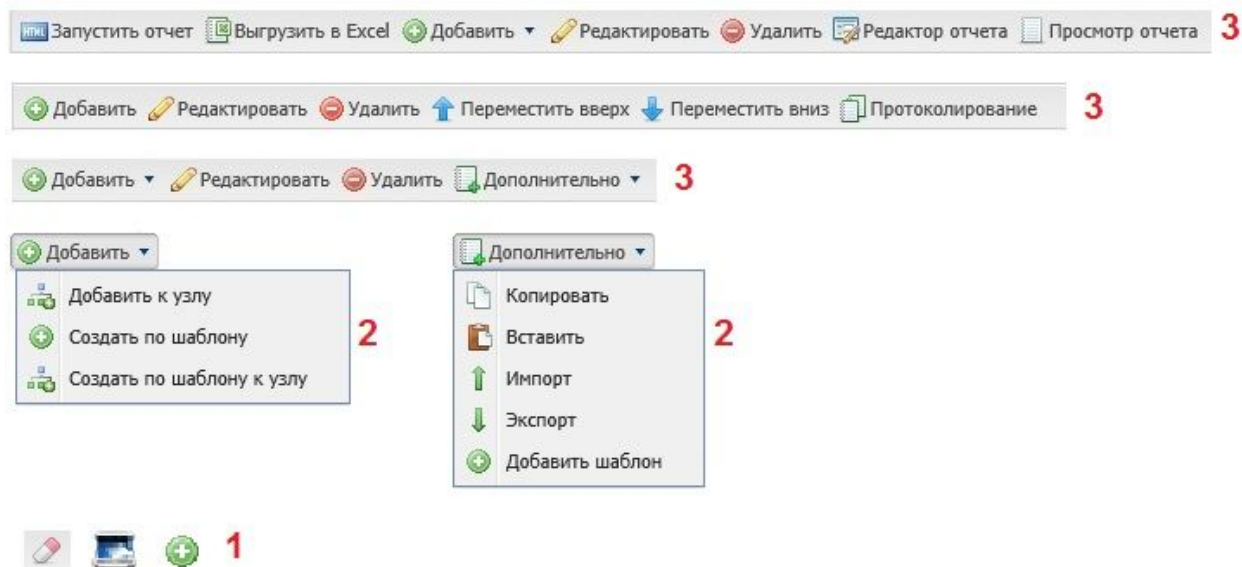


Рис. 3.5.1.1. Примеры группировки кнопок для вызова системных операций

К кнопкам для вызова системных операций. т.е. таких операций, которые настраиваются, но выполняют действия, которые заданы явно и поменять логику для них нельзя относятся:

- Добавить к узлу
- Создать по шаблону
- Добавить (объект)
- Создать по шаблону к узлу
- Редактировать (форму, таблицу, запись)
- Удалить (таблицу, файл)
- Дополнительно:
- Копировать (объект)
- Вставить (объект)
- Импортировать (объект)
- Экспортировать (объект)
- Добавить шаблон
- Переместить вверх (перемещение выделенной строки на одну позицию вверх)
- Переместить вниз (перемещение выделенной строки на одну позицию вниз)
- Протоколирование (операция открытия журнала действий)
- Применить настройки по умолчанию
- On-line просмотр
- On-line просмотр родителя
- Выход (выход из системы)

Системы разработанные на базе платформы Motiware Melody One могут содержать и кнопки для вывоза других операции, такие как:

- Сохранить (информацию в окне рабочей области)
- Закрывать объект в окна рабочей области (форму, таблицу)
- Выгрузить в Excel (таблицу)
- Обновить (окно рабочей области)
- Очистить (окно рабочей области, кнопка с иконкой обозначающей ластик)
- Фильтрация (настройка фильтра)
- Поиск
- Расширенный поиск
- Навигация
- Сканирование документа
- Данные (загрузка данных в таблицу)
- Расширенный редактор (открытие поля текстового редактора)
- Подсказка
- Добавление (файла, таблицы)
- Удаление (файла, таблицы)
- Просмотр (открытие файлов для просмотра)
- Автозаполнение (выполнение автозаполнения)
- Запустить процесс (запуск ETL процесса)
- Отчеты (переход ко вкладке с настройками отчетов)
- Запустить отчет (запуск формирования отчета)
- Редактор отчета (вызов редактора отчетов)
- Просмотр отчета (открытие окна для просмотра отчета)

В правом нижнем углу рабочей области располагаются кнопки (см. рис. 3.5.1.2.) : “Сохранить” - (1), для сохранения информации в окне рабочей области, а “Закрывать” для закрытия объекта (таблицы, формы) в окне рабочей области - (2).

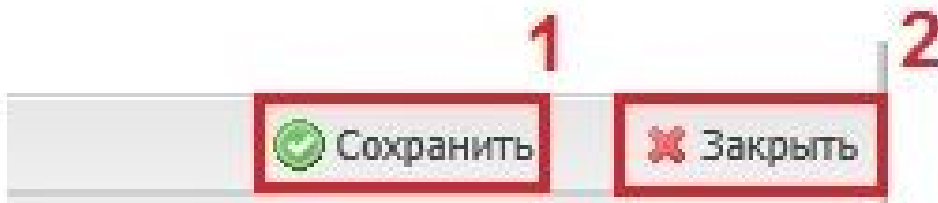


Рис. 3.5.1.2. Кнопки “Сохранить” и “Закрывать”

Кроме системных кнопок могут создаваться и пользовательские кнопки, настроенные на выполнение конкретных операций пользователя. Внешний вид системных и некоторых пользовательских кнопок приведен в Приложении 1.

3.5.2. Панель навигации

В левом нижнем углу рабочей области (см. рис.3.5.2.1.) располагается панель навигации, которая позволяет переходить по страницам. Если записей много, они разбиваются на страницы. Причем количество записей, отображающихся каждой странице настраивается, для каждого представления данных, индивидуально. Эта настройка производится в поле "Записей на странице" CoreFormView. Кроме того, можно осуществить настройку таким образом, чтобы в пунктах (подпунктах) меню содержалась информация об общем количестве записей в таблице, которая связана с данным пунктом.

Панель навигации - (1) позволяет переходить постранично вперед и назад, на первую и последнюю страницы, как и осуществлять переход на страницу с конкретным номером, выбрав номер страницы и нажав на клавишу "Enter". Рядом с панелью навигации размещена кнопка "Обновление" - (2), нажав на которую можно обновить (актуализировать) записи в таблице. В добавок рядом располагается кнопка "Фильтр" - (3), панель поиска - (4) и кнопка "Очистить" - (5).



Рис.3.5.2.1. Системные кнопки, расположенные в левом нижнем углу рабочей области

Для быстрого поиска необходимо ввести искомое слово (фразу) в окно поля ввода и нажать на иконку с биноклем или на клавишу "Enter". В режиме быстрого поиска можно подключить "живой" поиск, т.е. по мере ввода будет выполняться поиск записей. При этом поиск осуществляется по всем полям записей и по окончании поиска в правом нижнем углу будет выведена информация о количестве строк, в которых присутствует искомое слово. Для вывода из поиска и возврата в текущий (список, выводившийся до осуществления поиска) присутствует кнопка "Очистить", выполненная в виде ластика.

Помимо быстрого поиска существует поиск расширенный, кнопка запуска которого располагается слева от строки ввода поиска быстрого. При нажатии на эту кнопку выводится форма (см. рис.3.5.2.2.) , в которой отражаются записи всех полей и в которой можно осуществлять детальный поиск по значениям одного или нескольких полей (по их

сочетаниям). В рамках данного поиска для возврата к текущему списку можно воспользоваться кнопкой "Очистить".

Поле	Значение 1	Значение 2
Время действия		
Логин		
Событие		
Фамилия		
Имя		
Отчество		
IP		
ID session		
Заголовки запроса		
Тип данных		
Таблица		
Номер записи		
Имя записи		

Рис.3.5.2.2. Форма для расширенного поиска

В правом нижнем углу размещается информация об общем количестве записей и о том количестве записей, которые отображаются на данной странице.

3.5.3. Кнопки для настройки фильтров

Платформа содержит встроенный механизм фильтрации. Кнопка фильтрации содержит выпадающий список пред-настроенных фильтров, которые уже были настроены в системе и сохранены. Некоторые из этих фильтров являются общедоступными и их можно выбрать из списка.

Пользователь может настроить и свой собственный фильтр, для этого необходимо нажать на саму иконку воронки, и откроется другая форма, в которой в свою очередь присутствует список пред-настроенных фильтров (в левой части окна).

Настройка фильтра производится в специальном окне (см. рис. 3.5.3.1), где выбирается поле из типа данных, по которому будет осуществляться фильтрация.

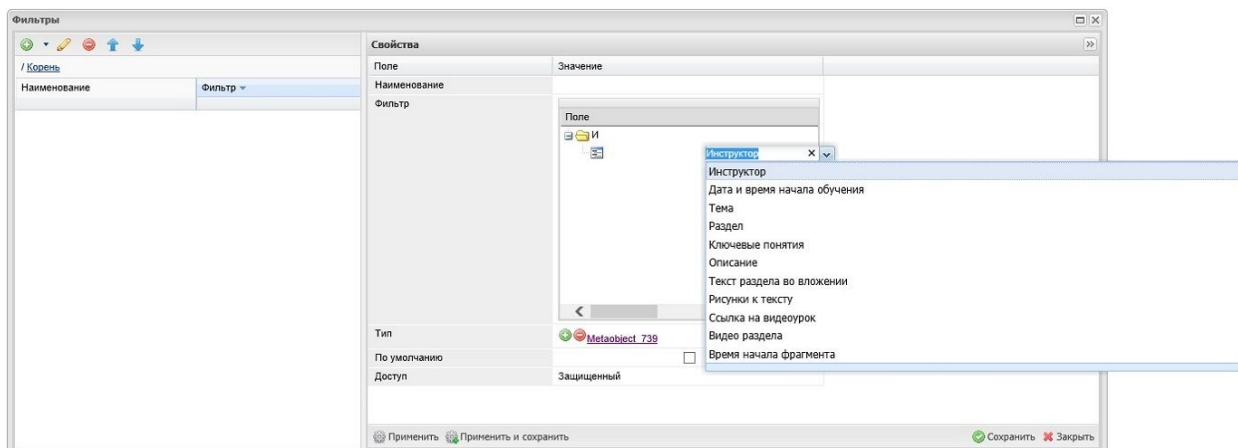


Рис. 3.5.3.1. Окно настройки фильтра

Далее выбирается операция, т.е. некий логический оператор, который будет служить условием (равно, не равно, меньше чем, больше чем, между, содержит, не содержит, начинается, не начинается и т.п.). Затем вводится какое-либо значение, при этом можно добавлять дополнительные условия, добавлять ветки условий, указывать логические операторы ("И", "ИЛИ", "НЕ") для присоединения условий, чтобы настроить необходимые и при этом достаточно сложные условия фильтра.

Кроме того фильтр может содержать дополнительное поля для ввода значения ("Значение 2"), которое используется для выполнения операции "между". Фильтр можно настроить по умолчанию с помощью установки флага в соответствующем поле.

Настроенную операцию фильтрации можно применить без сохранения, а можно применить и сохранить все параметры фильтрации. Соответствующие кнопки располагаются внизу окна "Фильтры".

ГЛАВА 4. Информационные объекты платформы Motiware Melody One

Статическая модель платформы Motiware Melody One определяет отображение сущностей (реальных объектов) из прикладной предметной области в системе, структуру этих сущностей и их взаимные связи. Все прикладные сущности представляются в системе в виде типов данных. Свойства сущностей определяются полями типов данных, которые, в свою очередь, обладают типизацией. Типы данных в системе образуют иерархию типов.

4.1. Типы данных

Тип данных представляет собой информационный объект, необходимый для классификации и описания реальных пользовательских сущностей. Тип данных содержит в себе набор полей, образующих информационный каркас, позволяющий:

- задать множество значений;
- установить набор операций (вычисления) со значениями;
- определить способ внешнего представления значений.

Все решения, разработанные на платформе Motiware Melody One, строятся на основе типов данных, их взаимодействии, их свойствах и их поведении .

Типы данных подразделяются на системные и пользовательские. Системные типы данных присутствуют в каждой системе (решении) в обязательном порядке, а пользовательские создаются, как правило, в соответствии с требованиями к конкретному решению, хотя и могут в определенной мере совпадать в инсталляциях, созданных для решения однотипных задач.

В любой системе, на любой площадке, разработанной на платформе Motiware Melody One, всегда присутствует корневой элемент, который является некоторой первоначальной записью, в которую вложены все остальные элементы, дочерние записи, типы данных, и которые относятся как к системным типам данных, так и к типам данных пользователей, привязанных конкретному решению.

4.1.1. Системные типы данных

Системные типы данных формируют основу системы и служат для обеспечения базового функционала. На рисунке 4.1.1.1. показано размещение системных типов данных в подпункте “Типы данных” - (1). Системные типы данных располагаются в ветке - Core (ядро системы) - (2), в которой в свою очередь содержится еще ряд вложенных веток (перечень системных типов данных приведен в приложении 2).

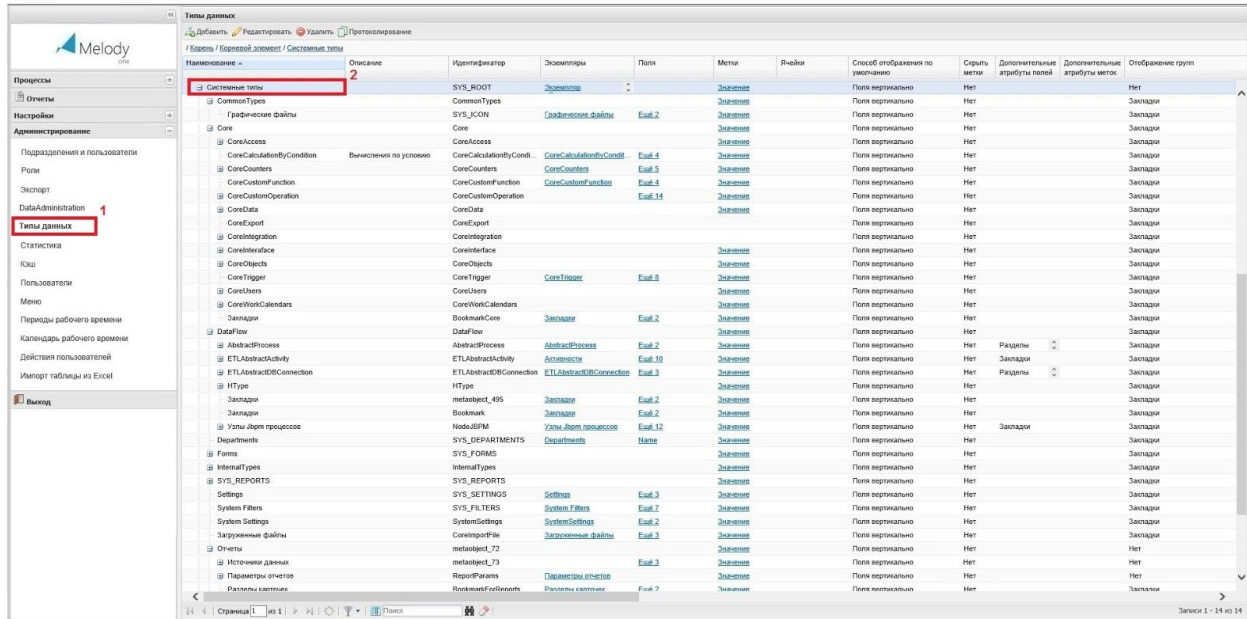


Рис.4.1.1.1. Ветка системных типов данных (Core)

В данной ветке содержатся основные системные настройки - настройки отображения, настройки работы фильтров, отчетов, планировщиков, подразделений, пользователей и многое другое.

При этом иерархия (подчиненность и вложенность) типов данных внутри ветки Core может изменяться администратором или пользователем в случае необходимости.

Основными типами данных платформы Motiware Melody One являются типы данных с идентификаторами:

- Metaobject - позволяет создавать различные пользовательские типы данных;
- Metafield - задает поля в типах данных.

Кроме того, используются "Метки" - это значения для полей, по умолчанию всегда будет создаваться одна метка и администратор (пользователь) может создавать еще необходимые метки (набор меток). В свою очередь здесь присутствует такой тип данных как "Ячейка". Каждая ячейка соответствует пересечению поля и метки. Кроме того, существуют другие служебные типы данных, такие как: "Экземпляры" и д.р. В пользовательском профиле системные типы данных должны быть скрыты.

4.1.2. Пользовательские типы данных

Пользовательские же типы данных создаются и настраиваются администратором системы и соответствуют конкретному решению, дополняя таким образом базовый функционал.

4.1.3. Свойства типов данных

Каждому типу данных присущи определенные свойства (поля). Общими полями для всех типов данных будут являться следующие:

- Наименование (название типа данных);
- Описание (краткая текстовая информация об этом типе, описание);
- Идентификатор (служебная информация);
- Способ отображения по умолчанию (поля вертикально или горизонтально);
- Отображение меток (отображать или скрыть метки);
- Дополнительные атрибуты полей ;
- Дополнительные атрибуты меток;
- Отображение групп (способ);
- Автосохранение (включено/выключено);
- Автовычисление (включено/выключено);
- Доступ (общедоступный/защищенный);
- Поле наименования;
- Уровень протоколирования (определяет в каком случае осуществляется протоколирование: всегда, только при добавлении, только при обновлении, только при удалении или нет);
- Таблица для хранения протокола;
- Конфигурация (ссылка на конфигурацию в соответствующей таблице);
- Настройка подписи (выбор характеристик цифровой подписи).

4.2. Типы полей

Тип данных можно определить на основании набора входящих в него характеристик. Эти характеристики представлены в платформе Motiware Melody One в форме различного вида полей, куда пользователь может внести информацию (значения), выбрать один из предлагаемых вариантов, указать ссылку, поставить флаг и т.д. Причем заданное множество полей со своими настройками - одна из определяющих характеристик типа данных. Таким образом, для каждого типа данных будет создаваться свой набор полей.

Список типов полей в платформе Motiware Melody One ограничен и задан заранее. Имеющегося списка типов полей (Core/CoreObjects/CoreEnumeration), см. рис. 4.2.1., вполне достаточно для разработки различных решений.

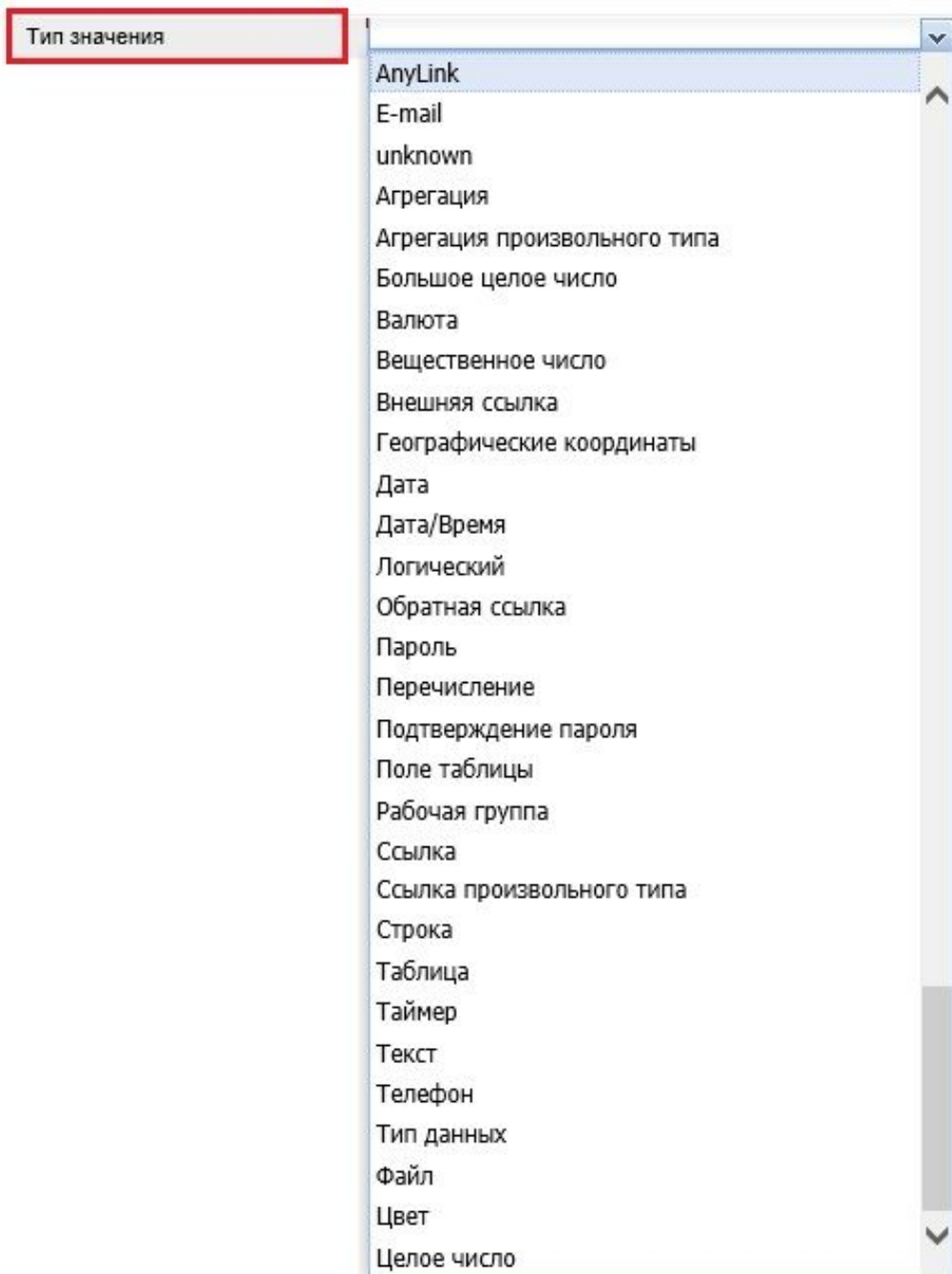


Рис. 4.2.1. Типы полей (типы значений)

Поля могут быть обязательными для заполнения и необязательными. Причем, обязательность заполнения поля может быть задана в поле "Обязательное", которое отображается в форме конкретного поля. При этом, если обязательность установлена (поле отмечено звездочкой красного цвета), то пока данные в это поле не будут внесены, сохранить изменения и перейти к следующей записи будет невозможно.

Кроме того, поля подразделяются на простые и сложные.

4.2.1. Простые типы полей

Простые типы полей - это такие типы, которые не включают в себя другие объекты (типы) и являются основой для построения других объектов (типов)

Тип поля	Описание типа поля
E-mail	Данное поле позволяет вводить значения с учетом ограничений, налагаемых на написание адресов электронной почты, т.е. используемый шаблон не позволит пользователю ввести простой набор символов (обязательное наличие "@", ".", отсутствие символов кириллицы);
Большое целое число	Знаковое (целое) 64-битное число в диапазоне: -2×10^9 ... $+2 \times 10^9$;
Валюта	Тип для отображения валют. В системных типах можно настроить вид отображения (рубли, доллары и т.п.) Фактически это вещественное число с разделителями между разрядами, в конце которого будет указатель на ту или иную валюту;
Вещественное число	Вещественное число двойной точности в диапазоне $1,7 \times 10^{-308}$... $1,7 \times 10^{308}$
Географические координаты	Поле для записи данных в формате географических координат по долготе и широте.
Дата	Дата в формате: DD.MM.YYYY, возможен как ввод в ручную, так и выбор из встроенного календаря
Дата/Время	Дата с временем в одном поле в формате: DD.MM.YYYY HH:MM:SS, возможен как ввод даты в ручную, так и выбор из встроенного календаря, время можно ввести ручную или выбрать из выпадающего списка значений с интервалом в 15 минут;

Логический	Принимает значения да или нет, не выбрано. Отображается в виде флажка, причем флажок может принимать три значения (установлено, запрет, пусто). Если на вкладке ограничения будет выбрано "ноль или одно значение", логических значений будет два - установлено или снято. Если "ноль или больше значений", как и "одно или больше значений", логическое поле может принимать три значения. По умолчанию (если поле на вкладке ограничения не задано) поле пустое. Этот факт важно иметь ввиду при построении ряда логических операций и при настройке фильтров;
Пароль (подтверждение пароля)	Текстовое поле, в котором при вводе символы заменяются звездочками.
Строка	Поле ввода, тип по умолчанию. Существует ограничение на длину поля ввода (по умолчанию 400 символов, максимально 3000 символов), при этом длина поля для ввода это длина поля в базе данных, т.е. число символов, которое может быть записано в поле базы данных, если при вводе число символов в строке превысит установленный размер, возможны ошибки при записи в поле базы данных;
Текст	Текстовые данные, состоящие из произвольных символов произвольной длины. Это поле отличается от поля строки тем, что можно вводить много-строчный текст, кроме того, можно воспользоваться расширенным редактором.
Телефон	Текстовое поле, имеющее формат телефонного номера
Целое число	Целое число в диапазоне $-2^{31} \dots +2^{31}$
Таймер	Позволяет установить время срабатывания таймера. Время задается в формате HH:MM:SS изменяя цифры гибко или с шагом в одну минуту. нажимая на стрелочки вверх или вниз. По умолчанию задается текущее время;

Цвет	Здесь можно установить выделение поля цветом при помощи цветовой палитры или вручную задавая код цветового оттенка;
-------------	---

4.2.2. Сложные (составные) типы полей

К сложным (составным) типам полей относятся типы (объекты), которые имеют внутреннюю структуру и содержат в себе простые объекты (типы). Сложные типы полей представлены в таблице:

Тип поля	Описание типа поля
Агрегация (вложенный справочник)	Задаёт отношение «главный-подчинённый» с объектами этого же или другого типа. Ссылка на экземпляр другого типа данных. Это поле с типом данных, встроенным в редактируемый тип данных, которое будет иметь вид таблицы (формы), с записями другого типа данных, вложенного в данный тип данных. Для агрегации всегда выбирается связанный объект (таблица) из перечня доступных;
Агрегация произвольного типа	Агрегация произвольного типа отличается от агрегации тем, что записи могут добавляться не из одного типа данных, а из нескольких (дочерних или вложенных);
Перечисление	Позволяет выбрать объект из списка (тоже самое, что и ссылка с настройкой “только из списка”);
Поле таблицы	Ссылка на поле этой же таблицы или же ссылка к полям произвольной таблицы (сама таблица задается в полях типа “таблица” или “ссылка на тип”);
Рабочая группа	Позволяет выбрать и включить в список пользователей, выбрав их из таблицы пользователей
Ссылка	Позволяет связать данное поле с записью в конкретной таблице, при выборе этого типа открывается дополнительное поле “поставщик данных” с кнопкой выбора элемента, с помощью которой необходимо выбрать связанный с этим

	<p>полем объект (тот тип данных, на который этот объект будет ссылаться). При этом для ссылки указывается поставщик данных (полный перечень поставщиков данных находится в ветке Core/CoreData/). Сами поставщики данных могут называться по разному. Особенностью типа данных типа "ссылка" является то, что это поле всегда связано с записями одного определенного типа данных. Отличается от Агрегации необходимостью настройки отношений на вкладке "Ограничение". Пример – поле "Омывается морями" объекта типа "Страна" ссылается на один или несколько объектов типа "Море";</p>
<p>Ссылка произвольного типа</p>	<p>Это более сложный тип ссылки. Ссылка произвольного типа позволяет указать в качестве поставщика данных некий тип данных, у которого имеется вложенный дочерний элемент. Например, есть поле ссылка на зарегистрированные документы, причем документы могут быть нескольких типов (входящие, исходящие и т.п.). В этом случае ссылка произвольного типа предоставляет возможность пользователю выбирать ссылку на любой из этих видов документов (если бы использовалась простая ссылка, то можно было бы выбрать только один из видов документов: или входящие, или исходящие, или какие-либо другие);</p>
<p>Обратная ссылка</p>	<p>Если имеются два типа данных, связанных ссылкой, то обратное отношение будет являться обратной ссылкой. Можно в том типе данных, на который ссылаются, отобразить конкретные записи, которые ссылаются на эту запись;</p>
<p>Внешняя ссылка</p>	<p>Ссылка на определенный ETL - процесс;</p>
<p>Таблица</p>	<p>Тип поля, в котором в качестве значения можно выбрать таблицу;</p>
<p>Файл</p>	<p>Данное поле позволяет прикреплять файлы, либо находящиеся на компьютере пользователя при помощи</p>

	стандартного диалога (так, например, работает “перетаскивание” из проводника в браузер), либо отправлять документы непосредственно со сканера, используя механизм встроенного сканирования с последующим прикреплением файла в поле ввода без сохранения на компьютере.
--	---

4.2.3. Атрибуты полей

Каждому полю могут быть установлены определенные ограничения (условия), которые в дальнейшем будут определять свойства данного поля. Настройка осуществляется при помощи следующих атрибутов:

Наименование атрибута	Свойство поля
1. Обязательное	Если выбрано "Да", то сохранение объекта в системе с пустым значением в данном поле запрещено.
2. Значение по умолчанию	Выражение, которое применяется для вычисления значения данного поля при создании нового объекта.
3. Вид значения	Если в списке выбирается “скаляр” - значение не привязано к пользователю, если “вектор” - значение привязано к пользователю.
4. Уникальное	При выборе этого атрибута делает поле становится как обязательным, так и проверяемым.
5. Комплексное поле	Данный атрибут используется в случае, если необходимо отобразить некоторые поля из агрегации. Указывается идентификатор поля агрегации и задаются функции для фильтрации.
6. Поле наименования	Применяется в том случае, если тип данных используется в качестве ссылки в каком-то другом типе, то определяется из каких полей будет формироваться наименование записи.

7.Тип вычисляемого поля	Определяет тип вычисления: вычисляемое виртуальное, вычисляемое выражение, вычисляемое комплексное, копия родительского поля, не задано.
8.Режим вычисления	Задаёт режим вычисления, т.е. в когда будет выполняться вычисление до или после: добавления, обновления или сохранения.
9. Редактируемое	Если выбрано "Нет",то ввод информации в данное поле будет запрещён.
10.Список для выбора значений	В случае если такой список задан, помимо ввода произвольной информации в данное поле появляется возможность выбора значений из этого списка.
11.Только из списка	Если выбрано "Да", то ввод произвольной информации в данное поле запрещён, и можно выбрать одно из значений из заданного списка в атрибуте Список для выбора значений.
12.Скрыть в форме	Если установлено "Да", то данное поле не отображается в форме создания/редактирования объекта.
13.Скрыть при создании	Если установлено "Да", то данное поле не отображается в списке полей, доступных для редактирования.
14.Скрыть при поиске	Если установлено "Да", то данное поле не отображается при выводе результатов поиска.
15.Скрыть в таблице	Если установлено "Да", то данное поле не отображается в табличных представлениях списков объектов.
16.Условие видимости	Атрибут позволяет задать правила видимости данного поля в зависимости от значений других полей объекта.
17.Условие редактируемости	Атрибут позволяет задать правила редактируемости данного поля в зависимости от значений других полей объекта.

18.Условие обязательности	Атрибут позволяет задать обязательность данного поля в зависимости от значений других полей объекта.
19. Связанный объект.	Атрибут предназначен для создания отношения с другим или этим же типом данных для полей типа Ссылка, Агрегация, Агрегация иерархии. Указывается конкретная таблица. Связанный объект это ссылка на конкретный экземпляр, конкретную таблицу, а поставщик это не только таблица.
20. Отношение	Атрибут предназначен для определения характеристик отношения для полей типов Ссылка, Агрегация, Агрегация иерархии и может принимать значения «0..1», «0..*», «1», «1..*».
21. Дополнительные атрибуты полей	Данный атрибут задаёт расположение поля на одной из закладок формы создания/редактирования и используется для настройки полей, а также в случае, когда в качестве поля используется не одно значение, а несколько значений, например номер и наименование поля. В таких случаях используется одно поле, отображаемое двумя колонками.
22.Дополнительные атрибуты меток	Аналогично дополнительным атрибутам полей.
23.Шаблон для валидации	Этот атрибут позволяет установить шаблонную запись с которой в последствии будут сравниваться значения в ячейке.
24. Автосохранение	Запускает пересохранение записи при каждом изменении поля.
25. Автовычисление	Задаёт пересчет значений во всех полях с формулами при каждом изменении поля.

26.Режимы вычисления	При помощи атрибутов “Вычислять при открытии” и “Вычислять после ввода” назначается определенный режим вычисления значений.
27.Визуальные настройки	Определяют свойства отображения полей. “Уровень” - задает смещение относительно левого края. “Штифт жирный” - позволяет выбрать данный тип шрифта. “Количество строк” - устанавливает число строк для отображения информации.
28. Использовать в ЭЦП	Определяет правило использования ЭЦП в отношении создаваемого (редактирования) типа данных, а поле “Настройка подписи”, позволяет выбрать определенный вид ЭЦП.
29. Представление данных	Данный атрибут задает определенные визуальные характеристики отображения данных, которые устанавливаются на основе системного типа расположенного ветке Core/ CoreInterface /Представление.
30.Функция фильтрации вложенных данных	Этот атрибут определяет функцию при помощи которой будет осуществляться фильтрация вложенных данных.
31.Функция получения отображаемого текста	Атрибут устанавливает функцию для получения отображаемого текста.
32.Функция получения ссылок на объект	Данный атрибут задает функцию получения ссылок на объект.

4.3. Метки и Ячейки

Метка - это значение поля. По умолчанию для каждого поля всегда создается одна метка (значение). В большинстве случаев для поля достаточно одной метки. Однако

существуют решения в которых требуется создание нескольких меток для отдельных полей.

Ячейка - это область, находящаяся на пересечении конкретного поля и определенной метки, в которой можно дополнительно задавать: тип значения и значение по умолчанию, устанавливать шаблон для валидации, определять поле как редактируемое или нет, вычисляемое или нет.

4.4. Таблицы (экземпляры типов данных)

Понятия экземпляр и таблица в в платформе Motiware Melody One равнозначны. Отдельный экземпляр или таблица служит контейнером для записи данного типа данных. Для того чтобы использовать для записи созданный тип данных, необходимо создать хотя бы один экземпляр. При этом экземпляров у одного типа данных может быть несколько. Причем у одного типа данных может быть несколько (один или больше) экземпляров, но экземпляр всегда привязан к одному типу данных. Созданные в системе типы данных служат своего рода каркасом для формирования экземпляров (таблиц).

4.5. Поставщики данных

Если экземпляр (Таблица) представляет собой некоторую группу записей, созданных по типу данных, Поставщик данных может выбирать из таблицы все данные, а может осуществлять выборку по определенным условиям. Поставщик данных используется в качестве источника данных для пунктов меню, полей типа "ссылка". Это позволяет ограничивать используемый перечень записей из таблицы. При создании экземпляра поставщик данных создается автоматически, причем создается поставщик, который выбирает все записи из таблицы.162

Поставщики данных предназначены для получения данных вне зависимости от способа их хранения.

Поставщик данных - это системный тип данных CoreDataProvider. Все созданные поставщики данных хранятся в системной ветке данных Core/CoreData/Поставщик данных. После создания таблицы (экземпляра типа данных) поставщик данных для неё создаётся автоматически.

При этом имеется два типа поставщика – таблица и процесс. Если в соответствующем поле типа данных установлено, что поставщик данных имеет тип «Таблица», то данные будут выгружаться из выбранной таблицы, если «Процесс» – будет запускаться выполнение выбранного процесса.

Форма для добавления поставщика данных приведена на рисунке 4.5.1. Эта форма содержит три вкладки “Общее” - а), “Поля вложенного справочника” - б), “Дополнительные поля для сортировки” - в.)

а)

б)



в)

Рис.4.5.1. Добавление поставщика данных

Вкладка “Общее” (рис.4.5.1. а) содержит следующие поля (свойства) для задания поставщика данных поставщика, а именно:

1. Наименование - произвольное наименование поставщика данных;
2. Идентификатор - уникальный идентификатор;
3. Описание - текстовое поле для внесения информации о этом поставщике данных;
4. Тип провайдера данных - выбор типа провайдера данных осуществляется на основе системного типа данных CoreDataProvider;
5. Таблица - позволяет выбрать источник данных;
6. Тип данных - вычисляемое поле в котором отображается ссылка на тип данных, в котором создается поставщик;
7. Использовать фильтр - определяет использование заданного фильтра;
8. Использовать операцию - определяет использование операции, которая задается в поле “Операция” (13);
9. Включен в рабочую группу - если установлен флаг, будет использоваться включение в рабочую группу. Пользователю доступны только те объекты, в рабочую группу которых он входит;
10. Включен в роль, если в этом поле установлен флаг, будет использоваться включение в роль. В данном случае присутствует следующая логика если в объекте есть ссылка на CoreRole с идентификатором "LinkRole" - записи становятся доступны, если роли пользователя совпадают с ролью записи этого поля;

11. Показывать шаблонные записи, если установлен флаг, будут отображаться шаблонные шаблонные записи;
12. Фильтр, Если установлен флаг в строке «Фильтр» поле “Использовать фильтр”, будет использоваться указанный здесь фильтр. Для его добавления необходимо

нажать на кнопку  . Откроется дополнительная форма. После этого требуется нажать на кнопку  . Будет предоставлена форма для назначения операций фильтрации;

13. Операция - Пункт предназначен для добавления операции, написанной на языке Python. запуск операции из CoreCustomOperation, в которой может быть указана произвольная функция из CoreCustomFunction (сейчас есть поддержка Python и Groovy). В настоящее время Система поддерживает создание функций на языках Python и Groovy;
14. Сортировка по умолчанию - указываются поля, по которым будет происходить сортировка по умолчанию;
15. Обратное направление - сортировка будет выполняться в обратном порядке;
16. Показывать через поисковую систему - установка флага приводит к отображению при поиске;
17. Процесс, если в поле Тип поставщика данных выбран «процесс», выбирается используемый процесс.

Кроме того, существуют две вкладки “Поля вложенного справочника” для показа полей вложенных данных в основной таблице (рис.4.5.1.- б) и “Дополнительные поля для сортировки” - для задания правил сортировки по нескольким полям (рис.4.5.1.- в).

4.6. Операции над типами данных

Динамические свойства системы определяют правила изменения данных в системе и могут быть разделены на две части:

Часть первая. Возможные операции над объектами системы. Для каждого типа данных в системе необходимо определить перечень операций, которые могут быть к нему применены. Операции задаются наименованием, входными параметрами и алгоритмом выполнения.

Пример: Операция «Расчёт графика платежей» в типе данных «Договор аренды», параметром для которого может служить, например, период времени, для которого этот график рассчитывается.

Часть вторая. Правила, по которым изменение одних сущностей ведёт к изменению других.

Пример: Поле Площадь в аренде объекта недвижимости содержит сумму арендованных площадей по нескольким договорам аренды по данному объекту недвижимости и должна модифицироваться при создании новых договоров или прекращении действия истекших.

Все операции объединяются в два типа: системные и пользовательские.

4.6.1. Системные операции

К системным операциям относятся такие как: добавить, редактировать, удалить, копировать, вставить, экспорт, импорт, перемещение строк вверх или вниз (порядок) в таблице и т.п. Как правило, системные операции отображаются в виде кнопок на панели управления (подробнее настройка операций описана в пункте 4.8.6).

4.6.2. Пользовательские операции

Кроме системных операций могут встречаться и пользовательские операции, которые представляют собой определенный код, выполняемый при нажатии на соответствующую кнопку. К примеру, такой пользовательской операцией в системе электронного документооборота может быть отправка документа на согласование или визирование. Называться такая операция может как "Отправить на согласование", "Отправить на подпись" и т.п., т.е. некоторые специфические действия, которые необходимы в рамках данного документооборота (подробнее настройка операций описана в пункте 4.8.6).

4.6.3 Триггеры

Триггер - это хранимая в платформе Motiware Melody One процедура особого типа (код, последовательность действий, операция), запуск которой осуществляется при наступлении определенных событий (действий), связанных с модификацией данных: добавлением, удалением строки в заданной таблице, или изменением данных в определенном столбце заданной таблицы, и которую пользователь не вызывает непосредственно. Список событий, при наступлении которых срабатывает триггер, задан в системе и приведен в приложении 3. Для каждого типа данных создается свой набор триггеров. При этом определяется очередность их срабатывания (подробнее настройка триггеров описана в пункте 4.8.6).

4.7. Создание и настройка типов данных

4.7.1. Создание типа данных.

Создание нового типа данных, как показано на рис.4.7.1.1. осуществляется в главном окне пункта меню "Типы данных" - (1). Для этого необходимо в иерархии типов данных выделить (в списке) родительский тип - (2), а затем нажать кнопку "Добавить" - (3).

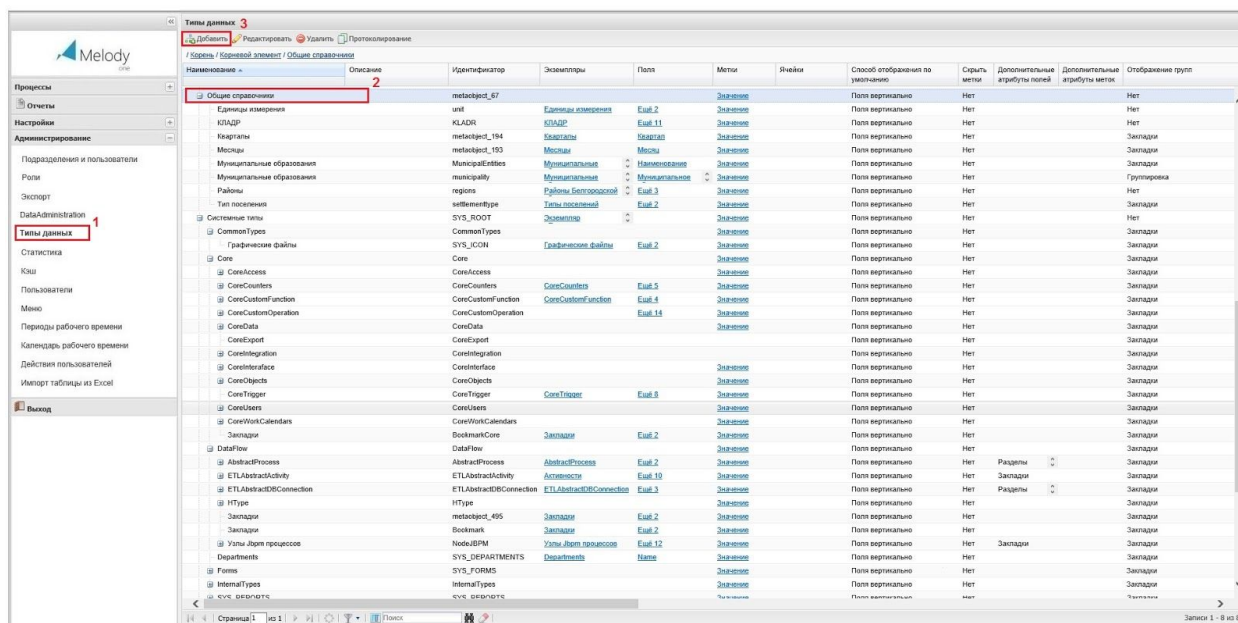


Рис.4.7.1.1. Создание нового типа данных

После нажатия на кнопку "Добавить" откроется окно создания типа данных (см. рис.4.7.1.2.), которое включает в себя несколько вкладок - (1): свойства, поля, метки, ячейки, экземпляры, дополнительно, ссылки, в которых в свою очередь присутствует набор полей (атрибутов). По умолчанию в окне создания типа данных отобразится вкладка, которая была открытой последней, перед этим действием.

Переход по вкладкам осуществляется щелчком по названию соответствующей вкладки. Внизу окна создания типа данных располагается панель управления, содержащая кнопки "Сохранить" и "Заккрыть" - (2).

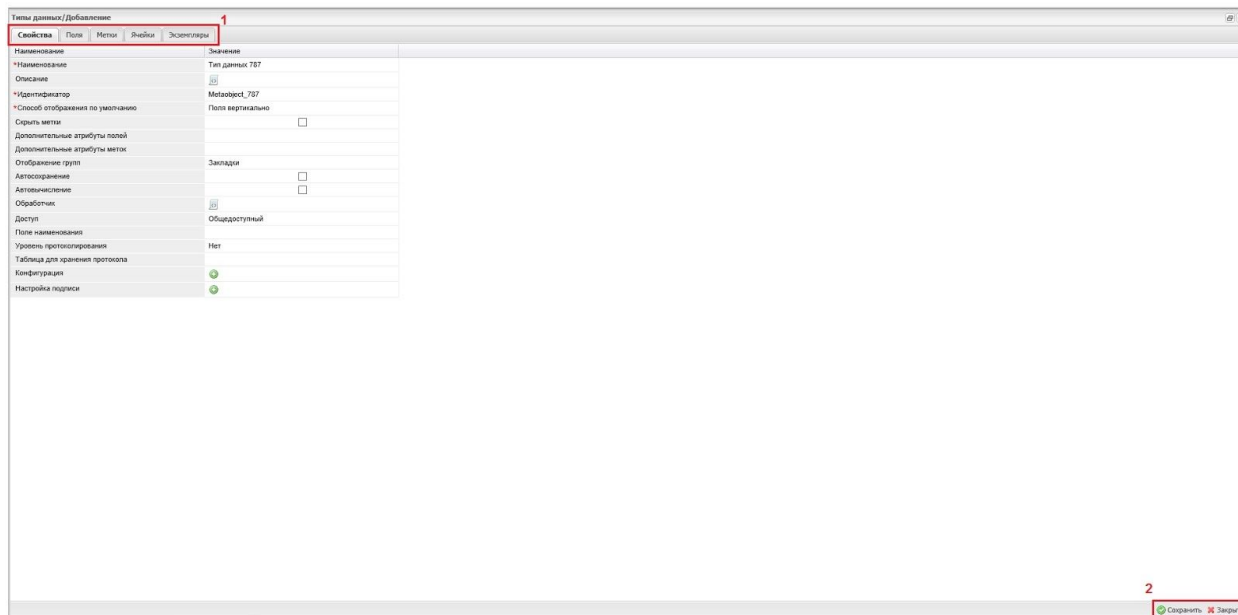


Рис.4.7.1.2. Создание нового типа данных

Данная панель отображается всегда, независимо от выбранной вкладки. Сохранение настроек типа данных осуществляется нажатием кнопки "Сохранить". Для перехода в главное окно рабочей области с деревом типов данных следует нажать кнопку "Закрыть".

Для создания нового типа данных требуется заполнить несколько обязательных полей на вкладке "Свойства" (см. рис. 4.7.1.3.). Поля, требующие обязательного заполнения, отмечены знаком "*" (звездочка) красного цвета (при отсутствии записей в данных полях форма не будет сохранена).

Прежде всего, указывается название типа данных в поле "Наименование" - (1), к названию можно добавить описание (поле описание не является обязательным для заполнения) в виде текста непосредственно в строке "Описание". Если же описание содержит много текста, можно, кликнув на иконку, расположенную в начале строки ввода, открыть окно расширенного редактора, в котором и осуществить ввод требуемого текста.

Типы данных/Добавление	
Свойства	
Наименование 1	Значение
*Наименование	Тип данных 787
Описание 2	
*Идентификатор 3	Metaobject_787
*Способ отображения по умолчанию	Поля вертикально
Скрыть метки	<input type="checkbox"/>
Дополнительные атрибуты полей	
Дополнительные атрибуты меток	
Отображение групп	Закладки
Автосохранение	<input type="checkbox"/>
Автовывчисление	<input type="checkbox"/>
Обработчик	
Доступ	Общедоступный
Поле наименования	
Уровень протоколирования	Нет
Таблица для хранения протокола	
Конфигурация	
Настройка подписи	

Рис. 4.7.1.3. Обязательные поля на вкладке свойства

Следующее обязательное для заполнения поле - "Идентификатор", запись в котором должна быть уникальной в рамках конкретной инсталляции (площадки). По умолчанию присутствует запись в виде "Metaobject_728", которую пользователь может изменить, задав собственный идентификатор - (2). Еще одно обязательное для заполнения поле - "Способ отображения по умолчанию" - содержит два значения по выбору "поля вертикально" и "поля горизонтально" - (3).

Заполнив форму во вкладке "Свойства", необходимо сохранить сделанные изменения кликнув мышью на кнопку "Сохранить", которая находится в правом нижнем углу вкладки (рис. 4.7.1.2. - 2). Выйти из этой формы можно воспользовавшись кнопкой "Закреть", расположенной в правом нижнем углу рядом с кнопкой "Сохранить" или кликнув на стандартный крестик в правом верхнем углу формы. Теперь в списке (таблице) типов данных можно наблюдать вновь созданный тип данных.

Полями "наименование", "идентификатор", "способ отображения" по умолчанию исчерпывается перечень обязательных для заполнения полей в форме "Свойства". В

зависимости от инсталляции (по требованиям заказчика), признаки обязательности могут быть настроены и для других полей.

Если необходимо создать несколько типов данных, то удобно объединить их в ветки, например, выделив новый тип данных и нажав на кнопку "Добавить", расположенную в верхнем левом углу рабочей области, внести записи в обязательные поля и сохранить создаваемый (после первого) тип данных. При этом во вкладке "Типы данных" у созданного первоначально типа данных (родительского) появится знак плюса, кликнув на который можно открыть строку соответствующую вложенному в него дочернему типу данных.

Таким образом можно формировать ветки и иерархии. По такому принципу формируются ветки информационных систем (ИС).

В дальнейшем можно создавать новые поля данных добавляя к этому набору или редактировать поля, созданные ранее. По окончании редактирования полей необходимо сохранить сделанные изменения в типе данных. При этом добавление поля и редактирование осуществляется поэтапно с сохранением данных после каждого добавления или редактирования. Кроме того, другой пользователь системы, обладая соответствующими правами, может внести изменения в существующие поля или добавить новые. Таким образом, возможен режим многопользовательской работы.

4.7.2. Настройка типа данных

После того, как конкретный тип данных создан, в любое время можно вернуться к его настройке или редактированию. Для этого необходимо выделить тип данных в списке и либо нажать кнопку "Редактировать", расположенную в левом верхнем углу рабочего поля, либо дважды щелкнуть по этому типу мышью. По умолчанию отобразится вкладка, которая была открытой последней, перед этим действием.

4.7.2.1. Вкладка "Свойства"

На вкладке "Свойства" (рис.4.7.2.1.1.) задается перечень свойств типа данных - (1), а именно: наименование, идентификатор, описание (в описании может быть введен произвольный текст, который может выступать в качестве некоторого комментария или подсказки и отображаться только для администратора системы, пользователям описание не доступно).

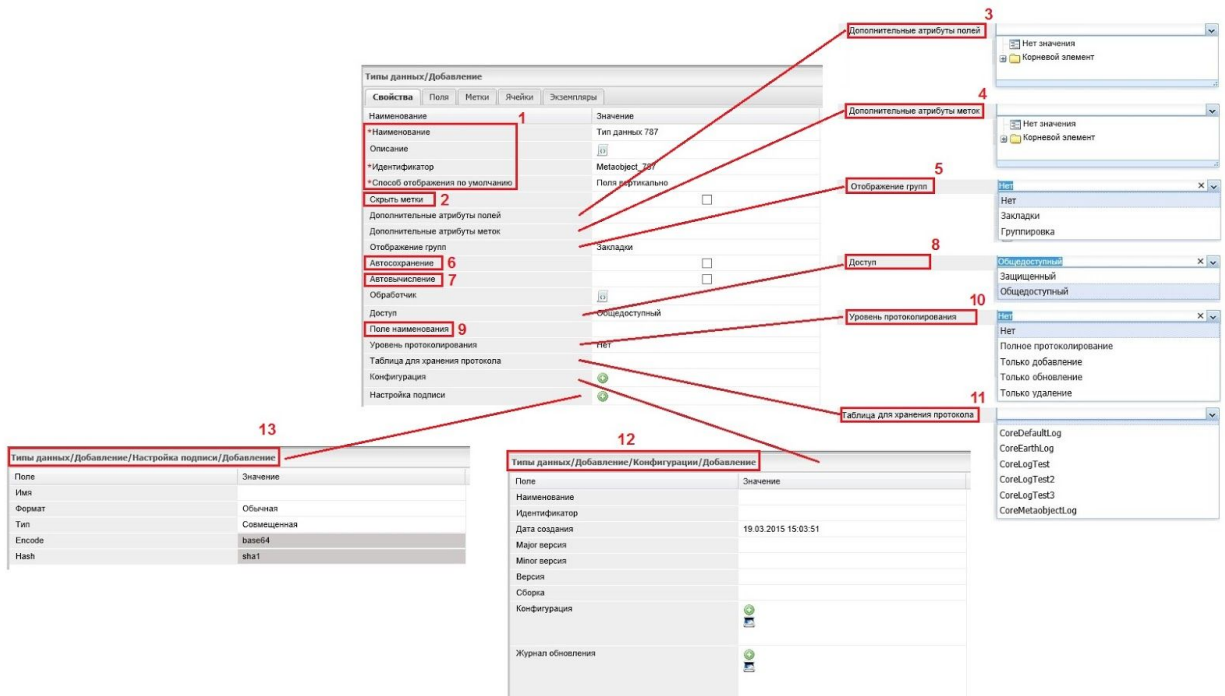


Рис. 4.7.2.1.1. Вкладка “Свойства”

Кроме того, вкладка "Свойства" содержит поле "Скрыть метки" - (2), позволяющее не отображать метки в пользовательском профиле.

Поле “Дополнительные атрибуты полей” - (3) используется в тех случаях, когда поле определяется несколькими значениями - (1), как это изображено на рисунке 4.7.2.1.2.

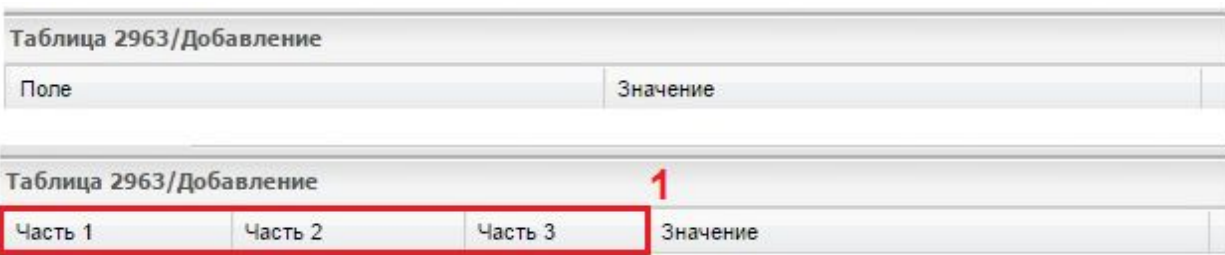


Рис. 4.7.2.1.2. Вид таблицы с полями, включающими в себя дополнительные атрибуты

При отображении “поля вертикально” дополнительные атрибуты (часть 1, часть 2, часть 3) образуют колонки в области поля.

Для этого вначале создается тип данных, как показано на рисунке 4.7.2.1.3. задающий дополнительные атрибуты (колонки), в котором число полей должно соответствовать количеству необходимых атрибутов.

Типы данных			
Добавить к узлу Редактировать Удалить Протоколирование Дополнительно ▾ Применить настройки			
/ Корень / Корневой элемент / Тест / Тип данных ВПВ / Дополнительные атрибуты полей			
Наименование ▲	ID	Описание	Идентификатор
Корневой элемент	1		metaobject_1
Демо	2190		Metaobject_842
Импортируемы из excel типы данных	2494		DataTypeFromExcel
Общие справочники	67		metaobject_67
Системные типы	69		SYS_ROOT
Тип данных ВПВ	2538	Тип данных для формирования руководства ВПВ	Metaobject_1097
Дополнительные атрибуты меток	2541	Дополнительные атрибуты меток для типа данных ВПВ	Metaobject_1101
Дополнительные атрибуты полей	2539	Дополнительные атрибуты полей для типа данных ВПВ	Metaobject_1099

Рис. 4.7.2.1.3. Тип данных для задания дополнительных атрибутов полей

У этого типа данных формируется один экземпляр (таблица), который содержит требуемое количество дополнительных атрибутов, как отображено на рисунке 4.7.2.1.4.

Таблица 2963/Добавление			
Часть 1	Часть 2	Часть 3	Значение

Рис. 4.7.2.1.4. Экземпляр для задания дополнительных атрибутов

Затем из списка в поле “Дополнительные атрибуты полей” - (1) основного типа данных (см. рис. 4.7.2.1.5.) выбирается необходимый экземпляр - (2). Если необходимо добавить еще один атрибут, он добавляется в типе данных с дополнительными атрибутами в виде поля.

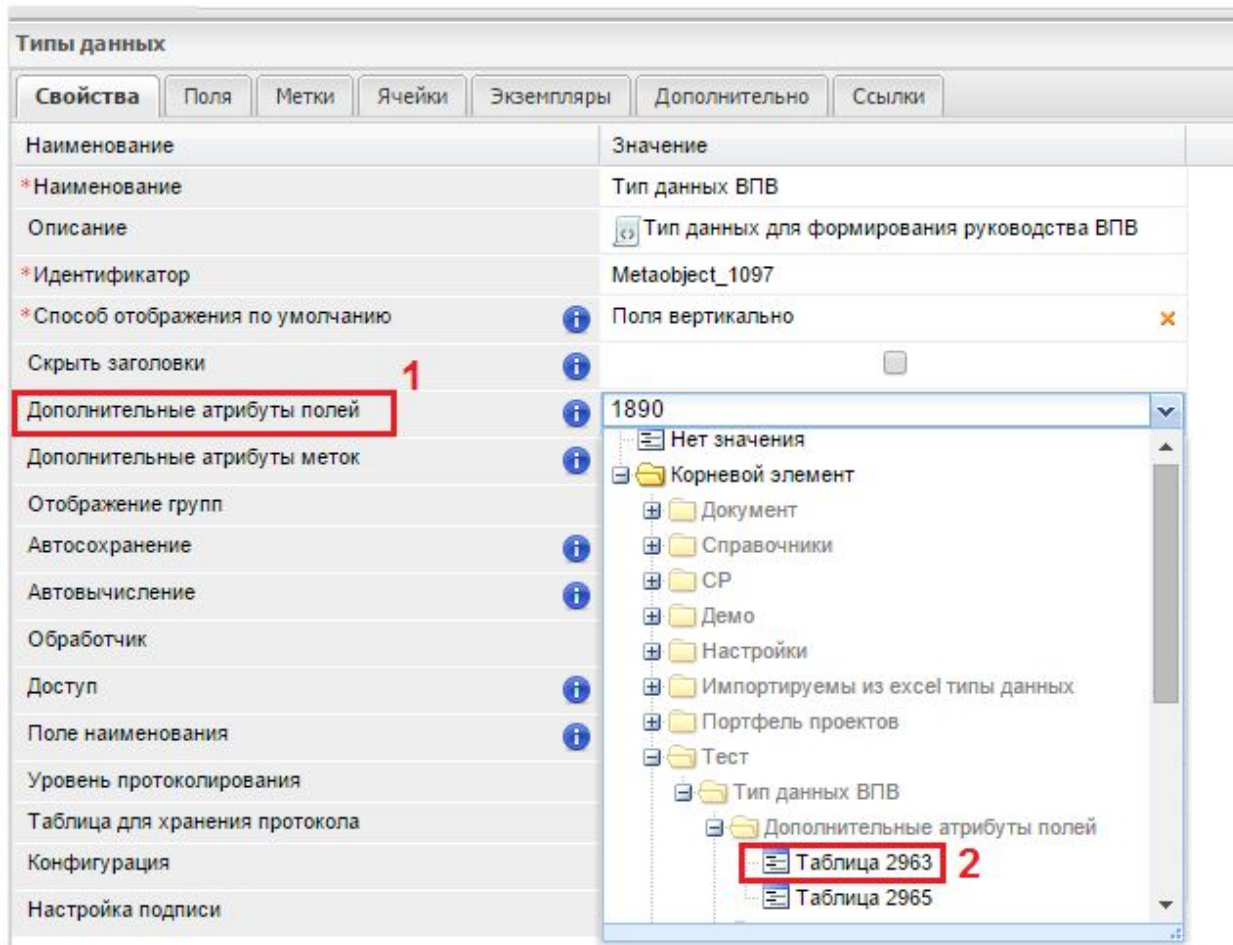


Рис. 4.7.2.1.5. Дополнительные атрибуты полей (задание)

Добавление дополнительных атрибутов меток в поле "Дополнительные атрибуты меток" - (4), см. рисунок 4.7.2.1.1., производится аналогичным образом.

Поле "Отображение групп" - (5) позволяет задать отображение поля на закладках или без закладок. Поле "Автосохранение" - (6) задает автоматическое пересохранение записей при каждом изменении какого-либо поля. А поле "Автовычисление" - (7), запускает пересчет значений всех полей, в которых заданы формулы при редактировании любого поля.

В поле "Доступ" - (8) можно настроить и доступ к типам данных, который может быть общедоступным или защищенным. Поле "Поле наименования" - (9) применяется если тип данных используется в другом типе данных в качестве ссылки и определяет какое из полей будет наименованием заголовка (по умолчанию берется первое поле). Поле "Уровень протоколирования" - (10) позволяет администратору системы настроить протоколирование с каждым типом данных в системе: будет ли вообще вестись журнал (выбрать "-Нет"), если да, то на каком уровне, или это будет только протоколирование

добавление действий в системе ("Только добавление"), или только по изменению ("Только обновление"), или только при удалении ("Только удаление"), или же будет вестись полное протоколирование ("Полное протоколирование"). т.е. все действия пользователя будут фиксироваться в журнале. Если выбрано любое действие, кроме "Нет", в поле "Таблица для хранения протокола" будет доступна одна из заранее созданных таблиц, в которые журнал действий необходимо сохранять. По умолчанию это таблица CoreDefaultLog, а при необходимости можно создать отдельную таблицу, в качестве таблицы назначений для хранения протокола действий и разместить ее в этом справочнике - (11).

Поле "Конфигурация" (12) предназначено для использования механизма конфигурации, который позволяет переносить настройки отдельных частей или блоков системы из одной инсталляции в другую.

В этой же вкладке можно настроить и электронную цифровую подпись (ЭЦП) поле - "Настройка подписи" - (13).

Заполнив форму во вкладке "Свойства", необходимо сохранить сделанные изменения кликнув мышью на кнопку "Сохранить", которая находится в правом нижнем углу вкладки.

4.7.2.2. Вкладка "Поля"

Основной каркас типа данных составляет набор полей, который редактируется на вкладке "Поля", как представлено на рисунке 4.7.2.2.1. Здесь администратором системы добавляются основные атрибуты типа данных, которые будут доступны или не доступны пользователям в формах ввода значений. Для того чтобы добавить поле, используется кнопка "Добавить" (уже созданное поле можно отредактировать при помощи кнопки "Редактировать"), расположенная в левом верхнем углу рабочего поля - (1).

На этой же вкладке присутствуют кнопки для перемещения выбранного поля по списку вверх или вниз - (2), кроме того, поле можно переместить и захватив его мышкой

(drag-and-drop).

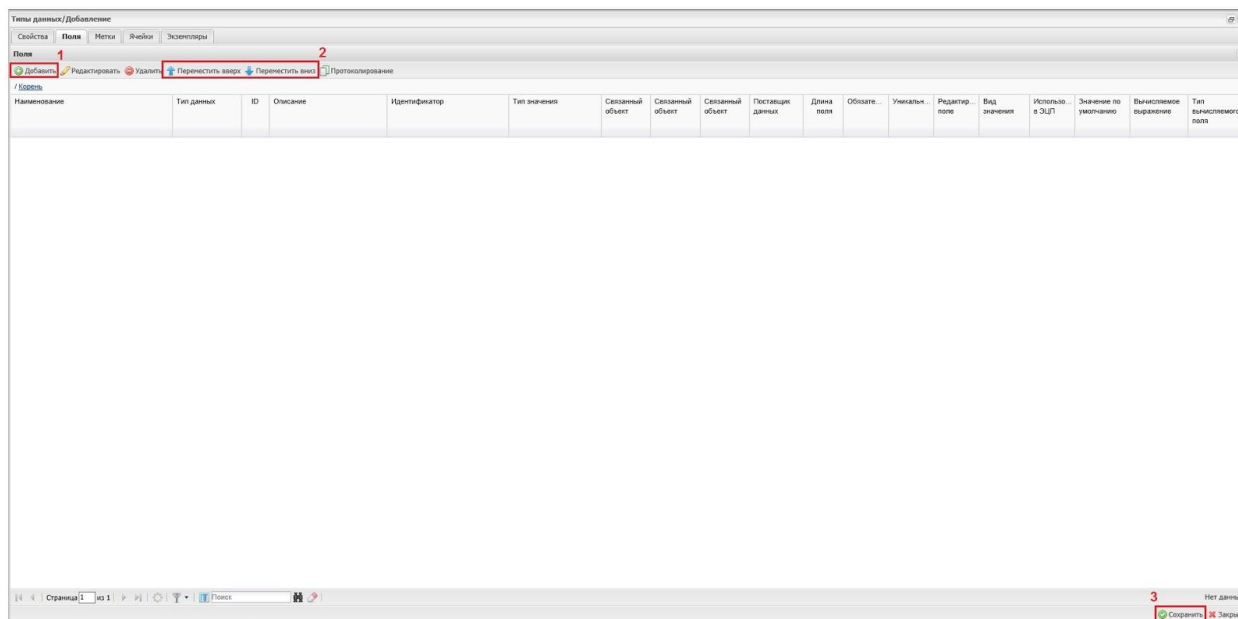


Рис. 4.7.2.2.1. Добавление поля

По завершению добавления (редактирования) списка полей необходимо сохранить сделанные изменения при помощи кнопки “Сохранить” - (3).

При добавлении нового поля открывается форма новой записи (Типы данных/Поле/Добавление), которая определяется системным типом данных в ветке системных типов Core/CoreData/Metafields и содержит в себе несколько вкладок со сгруппированными на них полями (рис. 4.7.2.2.2.):

- Свойства;
- Вычисление;
- Отображение;
- Ограничение;
- Вычисление по типам.

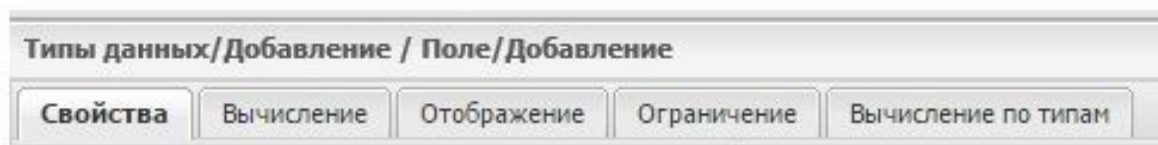


Рис. 4.7.2.2.2. Вкладки при добавления поля

Причем, отобразится та вкладка, которая была открытой последней, перед этим действием. Признак, запоминать вкладки или нет, настраивается в представлении данных. По умолчанию вкладки запоминаются.

Для редактирования полей на вкладках "Свойства", "Вычисление", "Отображение", "Ограничение" необходимо открыть соответствующую вкладку.

Вкладка "Свойства"

Для того чтобы добавить поле, необходимо заполнить обязательные поля (реквизиты) на вкладке "Свойства" (см. рис. 4.7.2.2.3.) Здесь обязательными реквизитами являются:

- Наименование (1);
- Идентификатор (3);
- Тип данных (тип значения) - (4).

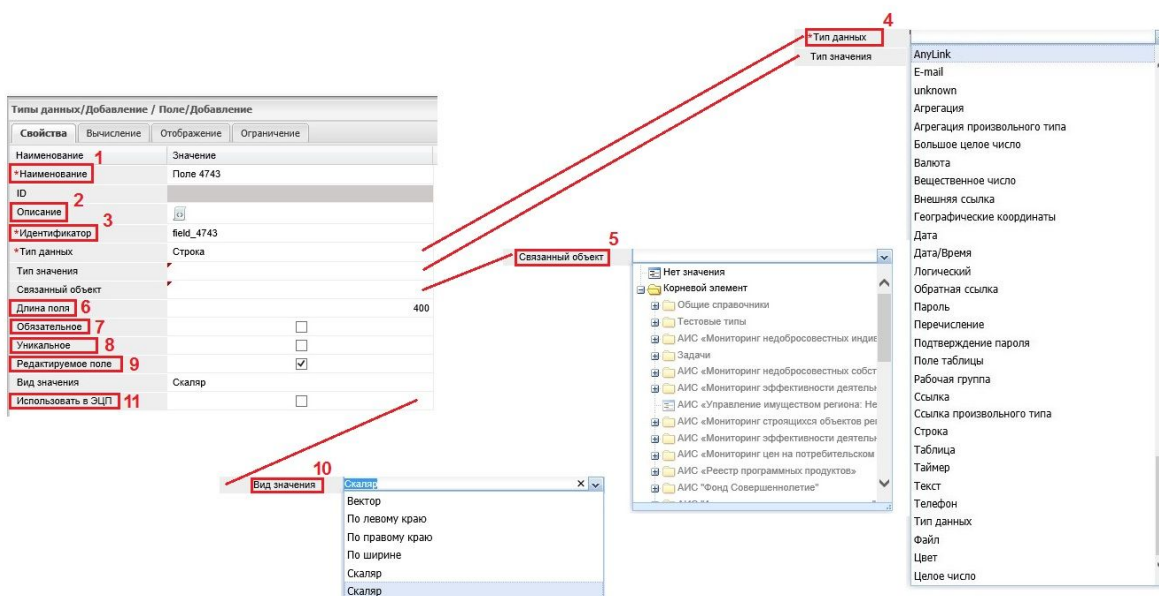


Рис. 4.7.2.2.3. Вкладка "Свойства"

В наименовании может использоваться кириллица, числа, подчеркивание и любые другие символы. Длина строки "Наименование" ограничивается 1000 символами. При необходимости данное ограничение можно изменить в ту, или в другую сторону.

В поле "Описание" можно добавить текст, который будет доступен пользователями и отобразится в виде подсказки в тот момент, когда пользователь приступит к внесению записей в данное поле.

Для удобства внесения текста присутствует кнопка расширенного редактора в поле "Описание" - (2), который позволяет оформлять текст используя различное форматирование (выделять текст жирным шрифтом, использовать курсив и подчеркивание, как и выравнивать текст разными способами), делая его более

наглядным и красочным. Кроме того в расширенном редакторе присутствуют кнопки переключения различных режимов, необходимые, например, для редактирования текста кода. Переключившись в режим редактора кода, можно отобразить смещения строк, служебные символы, подсвечивать строки и т.п. Здесь же присутствует кнопка (в виде воронки) вызова редактора фильтра.

Идентификатор - (3) аналогичен тому, который задается для типов данных, причем идентификатор должен быть уникальным в рамках типа данных, т.е. в нескольких типах данных могут быть поля с одинаковым идентификатором. В рамках одного типа данных два (или более) поля с одинаковым идентификатором быть не должно. Несмотря на то, что по умолчанию система проставляет идентификатор по шаблону, идентификатор рекомендуется изменять. Существует традиция именовать идентификатор в соответствии с наименованием поля (только латиницей полно или сокращенно), что в дальнейшем позволит проще идентифицировать данное поле, как и отличить это поле от других полей.

Поле "Тип данных" - (4) определяет тип поля. Типы полей в системе ограничены и заданы в виде выпадающего списка (описание типов полей приведено в п.4.2. Типы полей). Поля разделяются на простые и сложные, простые - это вещественное число, целое число, валюта, строка, текст и т.п. Сложные - это: агрегация, ссылка, ссылка произвольного типа и т.п. Как правило, заданного списка типов достаточно для разработки различных решений.

Поле "Связанный объект" задает ссылку на таблицу из справочника (5).

Поле "Длина поля" - (6) определяет длину строки, причем максимальная длина строки явно не ограничена (максимальное ограничение устанавливается используемой СУБД). Поле "Обязательное" - (7) устанавливает обязательность заполнения конкретного поля. Уникальность поля обозначается в поле "Уникальное" (8). Редактируемость поля (запрет для внесения изменений пользователем) устанавливается при помощи флага в поле "Редактируемое поле" - (9).

Помимо указанных выше полей на вкладке "Свойства" присутствует поле "Вид значения" - (10). При том, что большинство полей - скалярные, в некоторых случаях может потребоваться векторное значение поля, которое можно установить в данном поле. Здесь под "векторным значением" понимается тот факт, что векторное значение рассчитывается для каждого пользователя индивидуально (пример с индикацией обращений в техподдержку, у разных пользователей количество непрочитанных сообщений будет различным).

Кроме того для поля можно настраивать использование электронной цифровой подписи (ЭЦП), и если данное поле необходимо подписывать (что может быть актуально для поля типа "файл"), ставится галочка в поле "Использовать в ЭЦП" - (11).

Заполнив форму во вкладке “Свойства”, необходимо сохранить сделанные изменения кликнув мышью на кнопку "Сохранить", которая находится в правом нижнем углу вкладки.

Вкладка "Вычисление"

На вкладке "Вычисление" можно заранее задать определенные правила для полей (см. рис. 4.7.2.2.4.), так можно задать значение по умолчанию в поле “Значение по умолчанию” - (1), которое будет вычисляться один раз при создании записи, т.е. это значение будет отображаться в поле изначально. В свою очередь можно сделать это значение вычисляемым в поле “Тип вычисляемого поля” - (2), для этого необходимо выбрать тип вычисляемого поля, кликнув мышью на строчке в выпадающем списке. Тип вычисляемого поля может принимать следующие значения:

- Вычисляемое виртуальное;
- Вычисляемое выражение;
- Вычисляемое комплексное;
- Копия родительского поля;
- Нет.

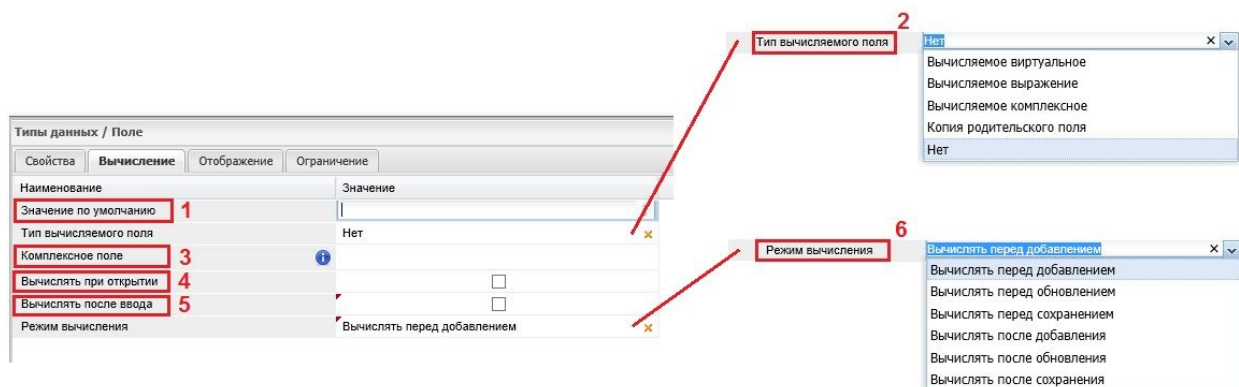


Рис. 4.7.2.2.4. Вкладка “Вычисление”

Как правило, используется или "Вычисляемое выражение", которое является по сути обычным вычислением в соответствии с некоторым алгоритмом или по формуле, или "Вычисляемое виртуальное", когда вычисления производятся что называется "на лету". При выборе "Вычисляемое выражение" открывается строка "Вычисляемое выражение", в которой может быть прописана функция в ручном режиме. Описания используемых функций приведены в приложении 4 и доступны на странице "МотиВики" по ссылке: [Описание функций](#).

Поле “Комплексное поле” - (3) применяется в том случае если необходимо отобразить некоторые поля из агрегации. Указывается идентификатор поля агрегации и

задаются функции для фильтрации. Поле "Вычислять при открытии" - (4) задает вычисление при открытии формы (таблицы).

В свою очередь, во вкладке "Вычисления" можно установить признак "Вычислять после ввода" - (3), который позволяет пересчитывать значение после ввода пользователем), равно как установить соответствующий режим вычисления в поле "Режим вычисления" - (4), т.е. когда будет производиться вычисление: вычислять перед (или после) добавлением, вычислять перед (или после) обновлением, вычислять перед (или после) сохранением. Выбор режима вычисления определяется конкретными требованиями в конкретной ситуации. После того как условия вычисления заданы, необходимо сохранить сделанные изменения.

Вместе с этим, в некоторых случаях, для вычисляемых полей может потребоваться изменить признак "Редактируемое поле" (см. рис. 4.7.2.2.5.) на вкладке "Свойства" - (1), сняв разрешение на редактирование, т.к. по умолчанию все поля редактируемые. Для пользователей нередатируемые поля будут отображаться заливкой поля серым цветом. Помимо этого, во вкладке "Вычисления" можно задавать и некоторые константы по умолчанию. Значения для строки задаются в кавычках, при этом перед строкой с кириллицей ставится символ "u".

1

Типы данных/Добавление / Поле/Добавление

Свойства		Вычисление	Отображение	Ограничение
Наименование	Значение			
*Наименование	Поле 4744			
ID				
Описание	<input type="text"/>			
*Идентификатор	field_4744			
*Тип данных	Строка			
Тип значения				
Связанный объект				
Длина поля				400
Обязательное				<input type="checkbox"/>
Уникальное				<input type="checkbox"/>
Редажируемое поле				<input type="checkbox"/>
Вид значения	Скаляр			
Использовать в ЭЦП				<input type="checkbox"/>

2

Рис. 4.7.2.2.5. Снятие признака редактируемости поля

Возможно и задание особой функции CoreCalculationByCondition, которая позволяет производить вычисления при задании различных условий, например, для одной роли пользователя вычисления производятся одним образом, а для другой роли другим.

Тип вычисляемого поля "Вычисляемое виртуальное" задает вычисление полей на лету без открытия и сохранения полей (по типу пересчета on-line).

Тип вычисляемого поля "Вычисляемое комплексное" может использоваться тогда, когда необходимо отобразить записи в виде ссылок, например - фамилии пользователей в таблице отображаются со ссылками на карточки пользователей, по которым осуществляется переход к полям, находящимся непосредственно в карточках.

В некоторых случаях бывает необходимо переносить создаваемые поля вместе со значениями из родительского поля в некую агрегацию, например, подчиненную, для этого используется механизм копирования полей при помощи операции "Копия родительского поля", вызываемой в поле "Тип вычисляемого поля", осуществляя таким образом некоторую трансляцию по иерархиям. Копирование полей осуществляется из вкладки

"Дополнительно" по операции "Копирование полей", задавая при этом поля откуда и куда производить копирование. Выбрав условие "Копировать данные", вместе с полями можно скопировать и данные.

Вкладка "Отображение"

Вкладка "Отображение" (см. рис. 4.7.2.2.6.). задает визуальные настройки определенного поля. Первое поле "Только из списка" - (1) задает список выбираемых значений посредством дополнительного поля "Список выбора значений", в котором значения непосредственно и задаются - (2). Значения вводятся в строке или в теле расширенного редактора через клавишу "Enter", и у пользователя не возникнет необходимости заносить эти значения вручную, он сможет выбрать их из выпадающего списка.

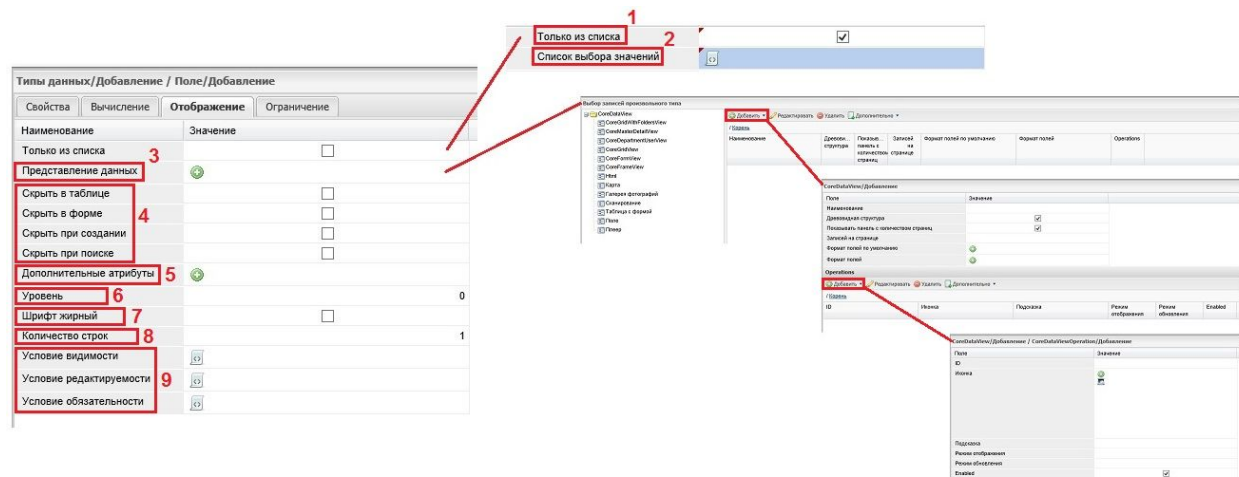


Рис. 4.7.2.2.6. Вкладка "Отображение"

Поле "Представление данных" - предоставляет возможность установить формат представления данных, используя при этом системный тип данных CoreDataView - (1). Здесь можно задать отображение древовидной структуры, определить количество записей на странице, выбрать формат полей по умолчанию, иконку, подсказку, режим отображения (меню, кнопка), режим обновления.

Поля с выбором операции (4):

- Скрыть в таблице;
- Скрыть в форме;
- Скрыть при создании;
- Скрыть при поиске.

Скрываются поля в каждом конкретном случае отображения таблицы у пользователя. Если выбран флаг "Скрыть в таблице", не будет доступен столбец (при

отображении поля вертикально). Если выбран флаг "Скрыть в форме", поле не будет отображаться в форме у пользователя.

Поле "Дополнительные атрибуты" - (5) служит для указания вкладки, на которой будет расположено данное поле.

Поле "Уровень" - (6) задает отображение поля в форме, оно позволяет визуально выделить запись, сдвигая ее относительно левого края на определенное число символов (аналогично использованию табуляции). Смещение относительно левого края задается цифрой в диапазоне от 1 до 20.

Поле "Жирный шрифт" - (7) позволяет выделить запись жирным шрифтом.

Поле "Количество строк" - (8) обычно влияет на текстовые поля и регулирует высоту поля, в котором будет отображаться информация. По умолчанию для текста отображается три строки. В системном типе данных CoreGridView в поле "Максимальная высота строки" можно установить и другое значение.

Дополнительные условия устанавливаются в полях - (9):

- Условие видимости;
- Условие редактируемости;
- Условие обязательности.

При этом поле "Условие видимости" задает условия, при которых поле будет видимо в форме. В отличие от операций "Скрыть в таблице" и "Скрыть в форме", поле "Условие видимости" позволяет задавать эти условия более гибко, т.е. в каких случаях поле будет появляться или исчезать. Для редактирования поля необходимо открыть расширенный текстовый редактор и перейти в редактор фильтра, т.к. условия настраиваются посредством фильтра. Причем эти условия могут копироваться из настроек других фильтров или создаваться самостоятельно. Если осуществлять настройку фильтра, в окошке фильтра первоначально формируется одна ветка с логическим оператором "И" и одно условие, в котором открывается выпадающий список полей редактируемого типа данных. После выбора поля по его наименованию в таблице отражается его идентификатор и следом выбирается операция из выпадающего списка (список условий достаточно широк). Затем в поля "Значение" и "Значение 2" необходимо внести требуемые данные. Условия можно усложнять добавляя ветки логических операторов. Процедуры формирования условий редактируемости и обязательности (красная звездочка в поле "наименование") аналогичны.

Вкладка "Ограничение"

Вкладка "Ограничения" (см. рис. 4.7.2.2.7.) содержит поле "Шаблон для валидации" -(1), - строку, в которую можно ввести шаблон по определенному значению (пользовательские ограничения).

Поле "Доступ" - (2) задает условия доступа: общедоступный или защищенный. Поле "Вес поля" - (3) в настоящее время не используется. Поле "Отношение" - (4) важно для логического поля, для ссылки, т.е. ссылка предполагает выбор элемента из какой-либо связанной таблицы и можно выбрать один или несколько элементов.

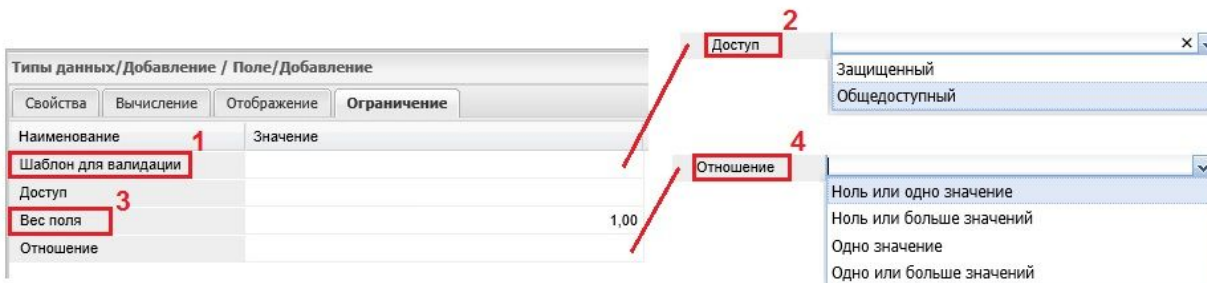


Рис. 4.7.2.2.7. Вкладка "Ограничение"

По завершению добавления (редактирования) полей необходимо сохранить сделанные изменения при помощи кнопки "Сохранить", которая расположена в правом нижнем углу рабочей области.

Вкладка "Вычисление по типам"

Вкладка "Вычисление по типам" в свою очередь содержит в себе три вкладки: "Поле", "Шаблоны", "Не шаблоны", как показано на рисунке 4.7.2.2.8

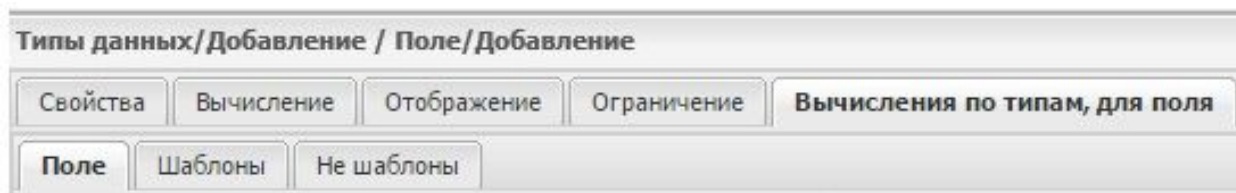


Рис. 4.7.2.2.8. Вкладка "Вычисление по типам"

На вкладке "Поле" можно выбрать грид для полей которого будет настраиваться вычисление (см. рис. 4.7.2.2.9.). Выбор осуществляется при помощи кнопки "Добавить" - (1), удаление с помощью кнопки "Удалить" - (2).

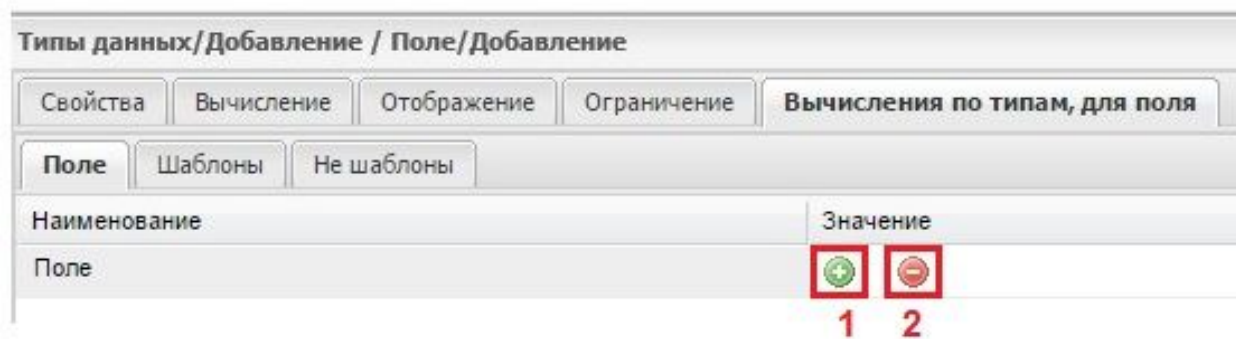


Рис. 4.7.2.2.9. Вкладка “Поле” (вкладки “Вычисление по типам”)

Вкладка “Шаблоны” предназначена для создания шаблонов вычислений и содержит поля для обработчика (программы), который будет выполняться при определенных условиях, сами условия сформулированы в поле “Наименование” (см. рис. 4.7.2.2.10.).

Типы данных/Добавление / Поле/Добавление	
Свойства Вычисление Отображение Ограничение Вычисления по типам, для поля	
Поле Шаблоны Не шаблоны	
Наименование	Значение
Значение по умолчанию при создании шаблона	<input type="text"/>
После создания	<input type="text"/>
Перед сохранением шаблона	<input type="text"/>
Перед вставкой шаблона	<input type="text"/>
После сохранения шаблона	<input type="text"/>
После вставки шаблона	<input type="text"/>
После изменения шаблона	<input type="text"/>
Перед открытием формы для шаблона	<input type="text"/>
Перед отображением в таблице для шаблона	<input type="text"/>
После загрузки шаблона	<input type="text"/>

Рис. 4.7.2.2.10. Вкладка “Шаблоны” (вкладки “Вычисление по типам”)

Вкладка “Не шаблоны” задает правила вычисления, вне шаблонов, и также содержит поля для обработчика (программы), который будет выполняться при определенных условиях, сами условия сформулированы в поле “Наименование” (см. рис. 4.7.2.2.11.).

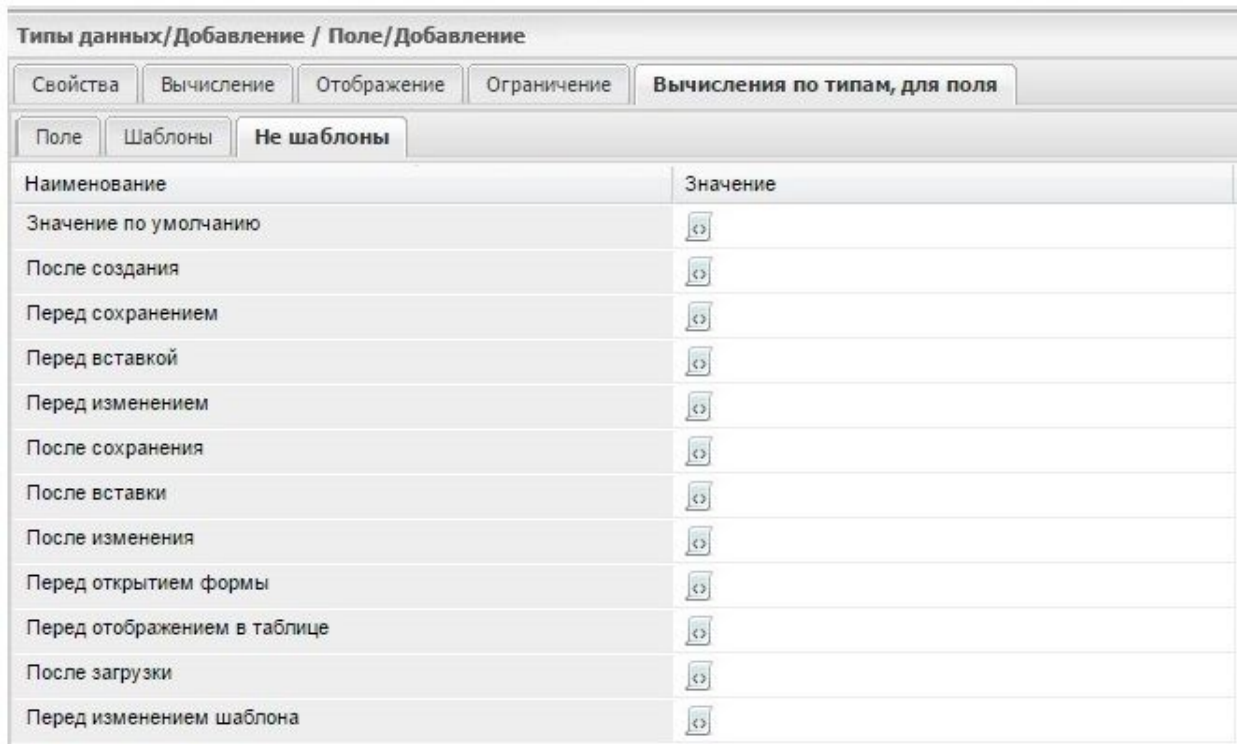


Рис. 4.7.2.2.11. Вкладка “Не шаблоны” (вкладки “Вычисление по типам”)

4.7.2.3. Вкладка “Метки”

Помимо полей у типа данных могут быть метки. Для их редактирования существует вкладка “Метки”. Всегда, по умолчанию, при создании типа данных создается одна метка, которая имеет наименование “Значение”, в большинстве случаев одной метки вполне достаточно. При этом, существуют особые случаи (особенно это касается систем по сбору отчетности), когда использование меток (больше одной) необходимо. Каждую метку можно настраивать индивидуально. Для меток, которые настроены вертикально (поля горизонтально), можно задать выбор дополнительных атрибутов (поле “Дополнительные атрибуты меток”) при помощи экземпляра (таблицы), как это описано для полей в п.4.7.2.1.

Обновление меток после настройки таблицы, выбранной в поле “Дополнительные атрибуты меток”, производится автоматически.

Для настройки меток необходимо на вкладке “Метки”, как показано на рисунке 4.7.2.3.1. - (1), нажать на кнопку “Добавить” - (2) при этом откроется форма “Добавление” (Типы данных/Метка/Добавление). По завершению добавления (редактирования) списка меток необходимо сохранить сделанные изменения при помощи кнопки “Сохранить” - (3).

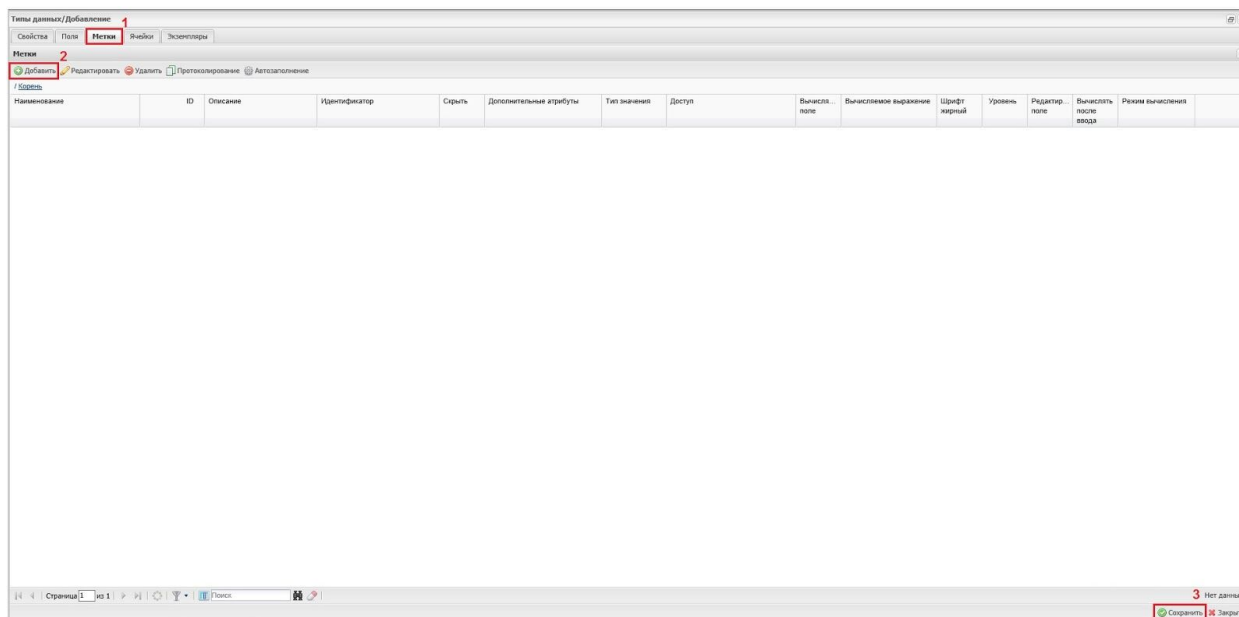


Рис. 4.7.2.3.1. Добавление метки

В форме "Добавление" (рис. 4.7.2.3.2.) обязательными для заполнения полями являются поля: "Наименование" - (1) и "Идентификатор" -(4). Значение в поле "ID" - (2) проставляется автоматически. В поле "Описание" - (3) можно привести описание данной метки, которое будет видно пользователю. Поле "Скрыть" - (5) позволяет установить флаг и в этом случае данная метка будет недоступна пользователю. В поле "Дополнительные атрибуты" задается ссылка на другие атрибуты меток - (6) В поле "Тип значения" - (7) задается тип поля (см. п.4.2. Типы полей).

Поле "Доступ" - (8) определяет будет ли метка общедоступной или с ограниченным доступом (защищенный). При помощи полей "Вычисляемое поле" - (9) и "Вычисляемое выражение" - (10) определяется будет ли поле вычисляемым и если да по какому выражению (формуле) будут вычисляться значения поля. Поле "Шрифт жирный" - (11) задает выделение шрифта. Поле "Уровень" - (12) определяет сдвиг наименования поля относительно левой границы. Возможность редактирования устанавливается отметкой (флагом) в поле "Редактируемое поле" - (13).

Поля "Вычислять после ввода" - (14) и "Режим вычисления" - (15) устанавливают правила вычисления значений поля.

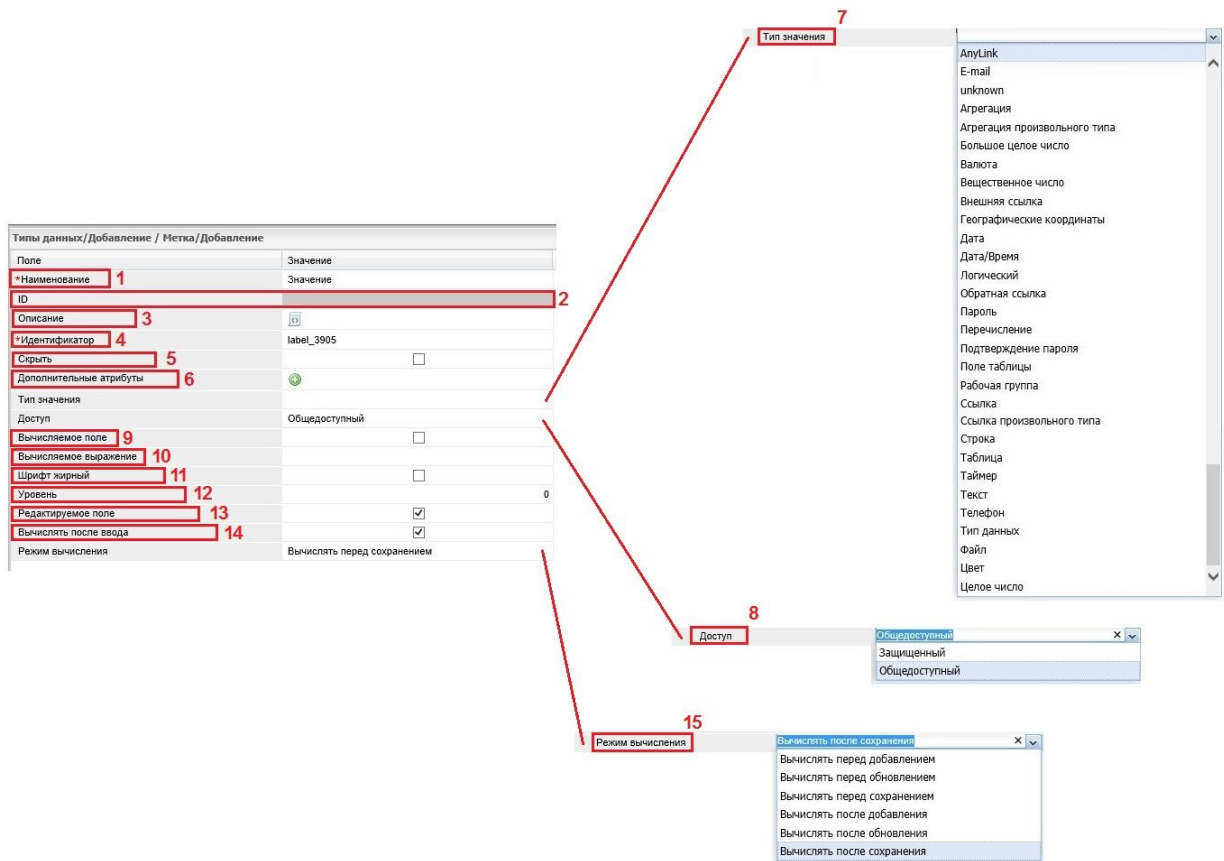


Рис. 4.7.2.3.2. Форма “Добавление”

По завершению добавления (редактирования) списка меток необходимо сохранить сделанные изменения при помощи кнопки “Сохранить”, которая расположена в правом нижнем углу.

4.7.2.4. Вкладка “Ячейки”

Ячейка является местом пересечения поля и метки. Как представлено на рис. 4.7.2.4.1., вкладка “Ячейки” - (1) позволяет отобразить все ячейки созданной таблицы, При этом по вертикали отображаются поля - (2), а по горизонтали метки - (3). Вкладка “Ячейки” используется для настройки свойств отдельных ячеек. При этом можно сделать так, чтобы отдельные ячейки были редактируемы или значения в отдельных ячейках вычислялись по определенному алгоритму. Те ячейки, для которых еще ничего не было задано или изменено, подсвечиваются розовым цветом - (4), при этом ячейка, в которой редактирование запрещено, подсвечивается серым цветом - (5).

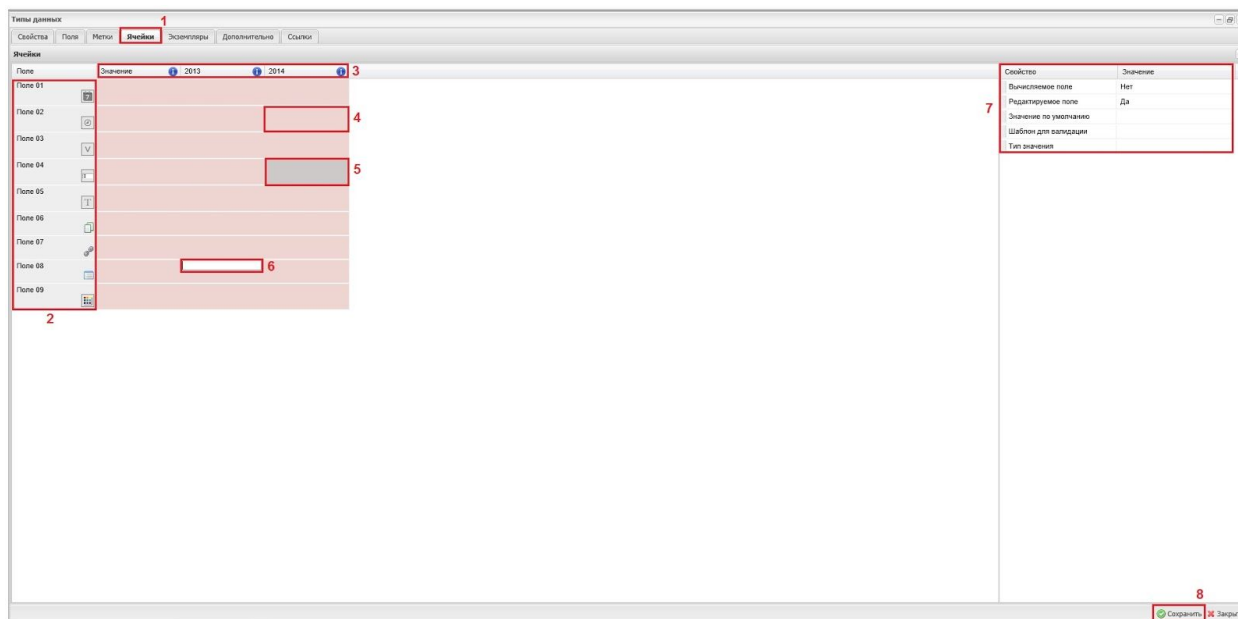


Рис. 4.7.2.4.1. Форма "Ячейки"

Для того чтобы изменить свойства конкретной ячейки, необходимо кликнуть по ячейке мышью, при этом откроется строка (6), в которую можно внести формулу для вычисления. К тому же, с помощью таблицы, расположенной в правом верхнем углу, можно отредактировать некоторые свойства данной ячейки - (7). А именно, установить будет ли поле (ячейка) вычисляемым, редактируемым, задать значение для ячейки по умолчанию, установить шаблон для валидации или определить тип значения.

Таким образом, значение в данной ячейке можно вычислять на основе значений в других ячейках. Если не включено вычисление, то по умолчанию ячейка является редактируемой. Как только в ячейку вводится формула, она становится вычисляемой и не редактируемой автоматически. При этом редактируемость ячейки можно изменить в поле на вкладке "Ячейки", расположенном справа сверху.

После редактирования полей ячеек необходимо сохранить сделанные изменения при помощи кнопки "Сохранить" - (8), которая расположена в правом нижнем углу.

4.7.2.5. Вкладка "Экземпляры"

Как показано на рисунке 4.7.2.5.1., вкладка "Экземпляры" - (1) предоставляет возможность создавать и редактировать экземпляры (таблицы) данных. Причем экземпляров (таблиц) для выбранного типа данных может быть несколько. При этом помимо добавления новой таблицы посредством нажатия на кнопку "Добавить" (2), можно и отредактировать уже существующий экземпляр с помощью кнопки "Редактировать" - (3).

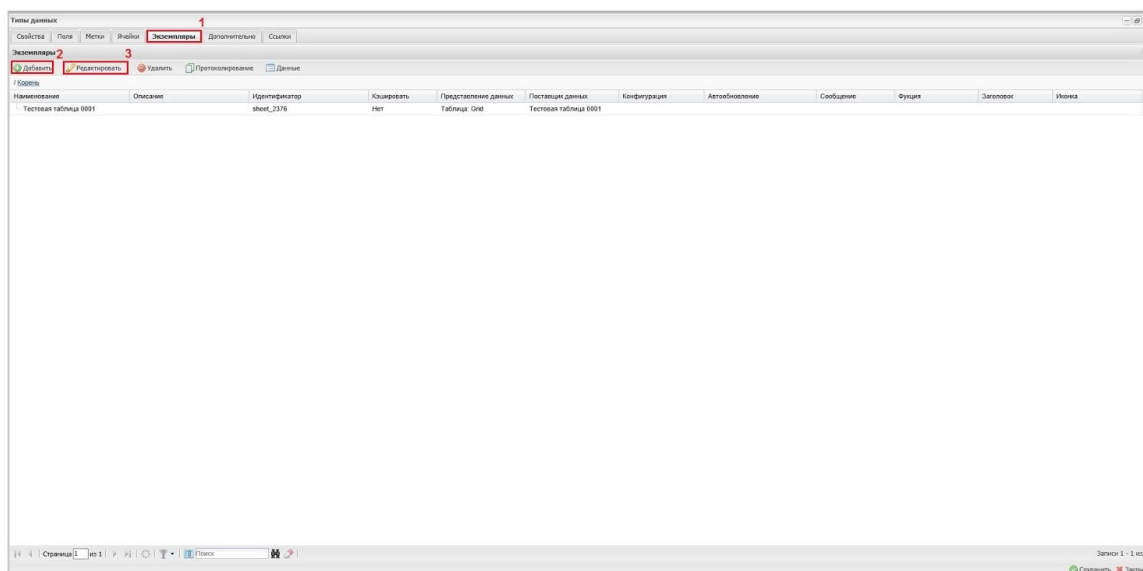


Рис. 4.7.2.5.1. Добавление экземпляра (таблицы)

При создании экземпляра (таблицы) в форме Типы данных/экземпляры/Добавление (см. рис. 4.7.2.5.2.) к полям "Наименование", "Описание", "Идентификатор" - (1) добавляется ряд полей. Это поля: "Кэшировать" - (2), "Представление данных" - (3) и "Поставщик данных" - (4), т.к. таблица представляет собой некоторую группу записей, созданных по типу данных, поставщик данных может выбирать из таблицы все данные, а может осуществлять выборку по определенным условиям.

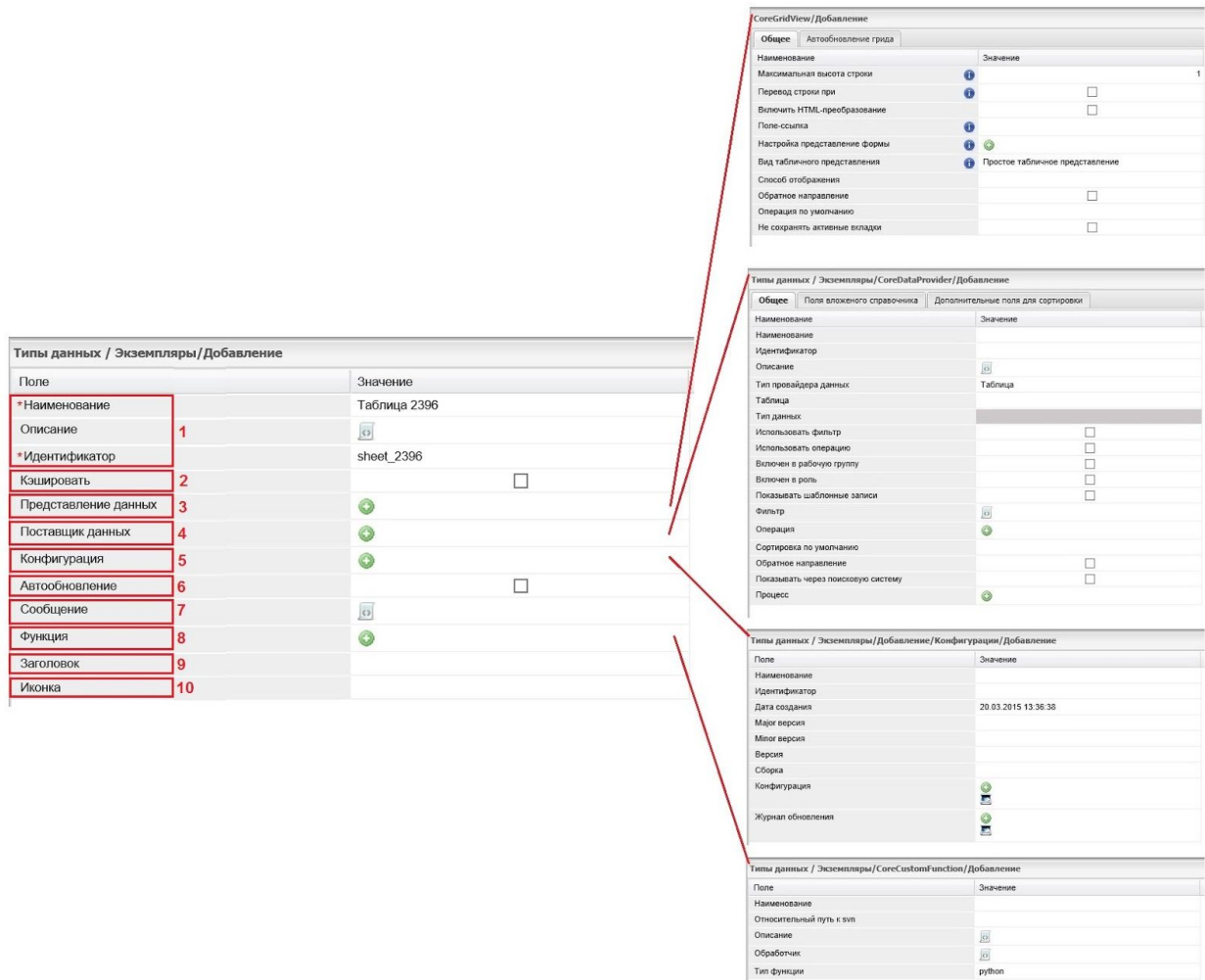


Рис. 4.7.2.5.2. Создание экземпляра

Поле "Представление данных" - это системный вид поля CoreGridView, который располагается в системной ветке Core/CoreInterface/, который определяет некоторые визуальные характеристики отображения этих данных. Существует два основных вида представлений - представление в виде таблицы и представление в виде формы. При нажатии на кнопку со знаком "+" в поле "Представление данных" осуществляется переход по ссылке в форму "Выбор записей произвольной формы", где открывается список, в котором можно выбрать представление в виде таблицы CoreGridView, или представление в виде формы CoreFormView. В представлении располагается набор записей, основная из которых и которая проставляется по умолчанию, это запись Grid. Можно создавать копию этой записи и менять настройки, к которым относятся: максимальная высота строки, перевод строки и д.р. В свою очередь в табличном представлении может присутствовать ссылка на представление в виде формы. Кроме того, представление в виде таблицы может иметь простое табличное представление и упорядоченное табличное

представление. Причем в простом табличном представлении изменить порядок строк невозможно. По умолчанию проставляется простое табличное представление. Для того чтобы иметь возможность менять местами порядок записей, необходимо выбрать упорядоченное табличное представление.

Поле "Поставщик данных" используется в качестве источника данных для пунктов меню, полей типа ссылка. Это позволяет ограничивать используемый перечень записей из таблицы. При создании экземпляра поставщик данных создается автоматически, причем создается поставщик, который выбирает все записи из таблицы. Поставщик данных - это системный тип данных CoreDataProvider. Все поставщики данных хранятся в системной ветке данных Core/CoreData/Поставщик данных. По умолчанию при создании экземпляра создается поставщик данных без всяких ограничений, без фильтров и т.п. в поле "Поставщик данных" указывается наименование, совпадающее с наименованием таблицы, с типом провайдера данных - таблица (кроме того возможен тип провайдера процесс). В строке "Таблица" указывается конкретно из какой таблицы будут использоваться данные и ссылка на тип данных в строке "Тип данных". Если необходимо задать какие-либо ограничения, например, отражать меню только у определенных пользователей, их можно выполнить во вкладке CoreDataProvider, например, установив галочку в поле "Настроить фильтр", при этом сами условия фильтрации задаются в поле "Фильтр".

В поле "Конфигурация" - (5) можно создать новую конфигурацию, для данного экземпляра, или выбрать и отредактировать конфигурацию из предлагаемой таблицы.

Настройка автообновления, при помощи поля "Автообновление" - (6), позволяет при наступлении определенных событий производить обновление записей в ячейках. Например, при регистрации документа обновлять данные о количестве документов в выбранных ячейках (входящие документы, документы поступившие от указанного адресата и т.п.) или использовать какой-либо способ условного форматирования для выделения ячеек (цвет шрифта или фона) в которых информация обновилась.

Поле "Сообщение" предназначено для внесения текстовой информации, а поле "Функция" - (8) для выбора функции, которая будет вызываться при наступлении определенного события.

Поле "Заголовок" - (9) содержит строку для внесения информации, которая будет отображаться в заголовке всплывающего сообщения, а поле "Иконка" - (10) область, в которой можно задать пиктограмму всплывающего сообщения.

По мимо настройки таблицы можно производить и настройку формы, выбрав системный тип данных CoreFormView. В форме можно настроить количество записей, отображаемых на каждой странице. Здесь же присутствует кнопка выбора древовидной структуры отображения, которая позволяет формировать и переподчинять иерархии путем перетаскивая элементы из одной родительской ветки в другую.

Вкладку “Экземпляры” можно использовать и для создания шаблонов (кроме описанного в пп. 3.3.2.13) непосредственно из грида записей конкретной таблицы. Для этого необходимо выделить строку записи определенной таблицы, отображаемой на вкладке “Экземпляры”, дважды щелкнуть по ней мышкой в открывшемся представлении данной таблицы, как показано на рисунке рис. 4.7.2.5.3. выбрать в выпадающем списке кнопки “Дополнительно” - (1), операцию “Добавить шаблон” - (2).

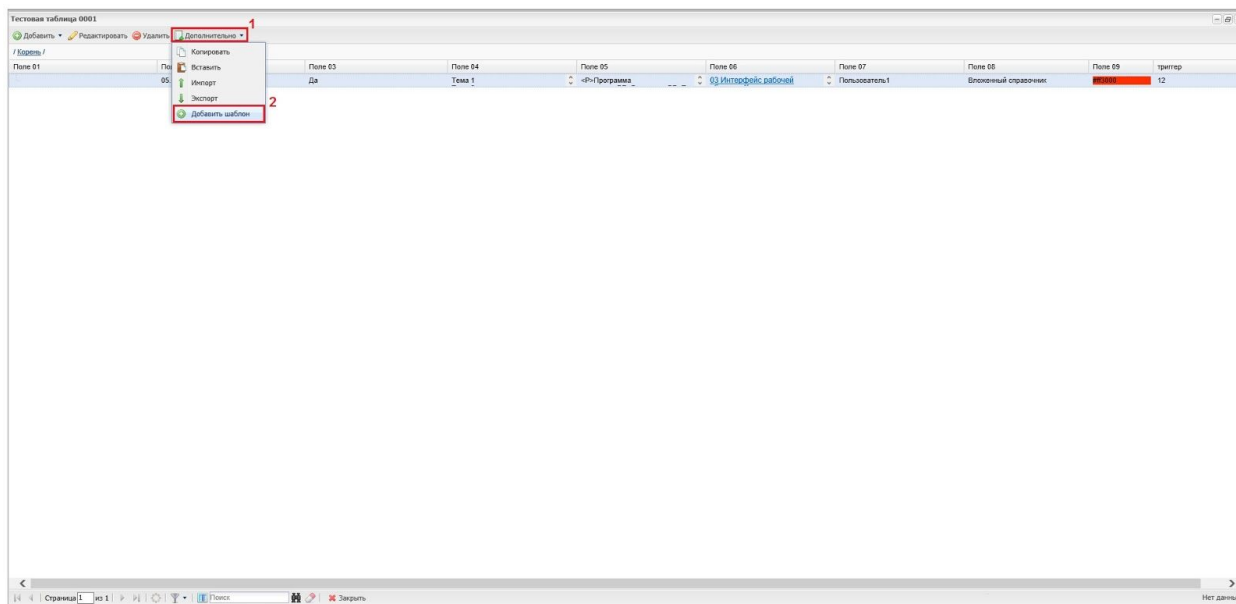


Рис. 4.7.2.5.3. Создание шаблона из грида записей конкретной таблицы.

После выполнения описанных выше действий, в форме добавления шаблонов (см. рис.3.3.2.13.2.) ввести наименование шаблона, установить правила модификации шаблона (при необходимости) и сохранить сделанные изменения.

4.7.2.6. Вкладка "Дополнительно"

На вкладке “Дополнительно” (см. рис. 4.7.2.6.1.) собраны ссылки на формы, которые могут быть полезными при настройке типов данных, а именно:

- Триггеры
- Операции
- Форматы
- Роли

- Изменение кодировки полей
- Копирование полей
- Индексирование и поиск
- Пересохранить данные

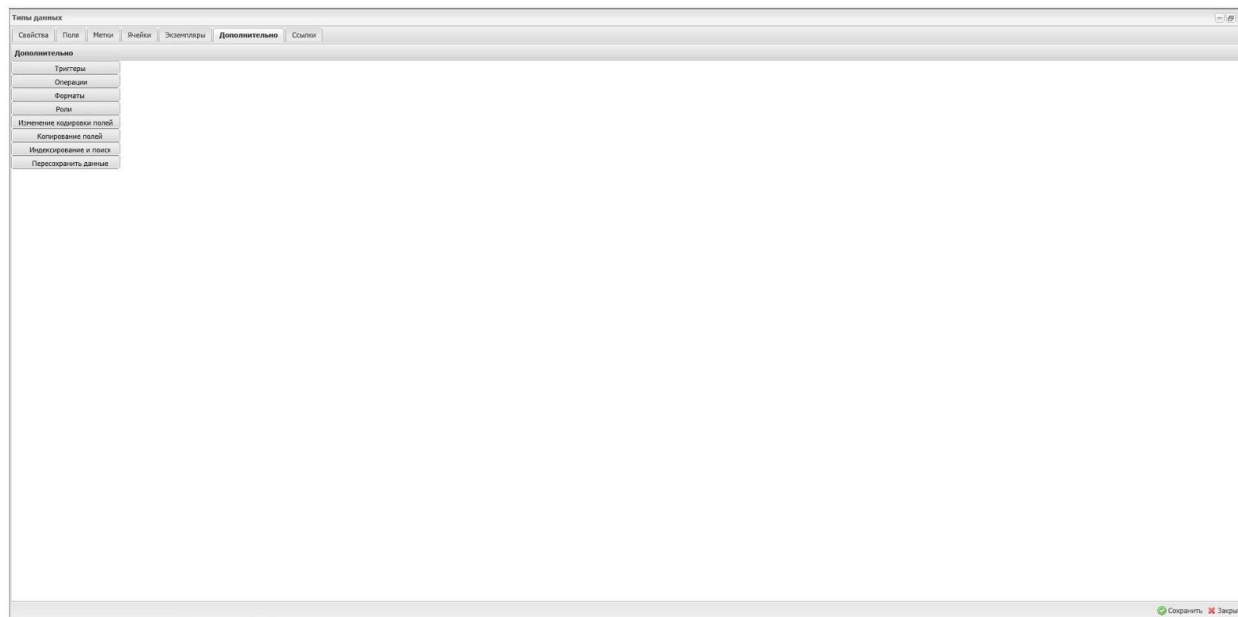


Рис. 4.7.2.6.1. Вкладка "Дополнительно"

Форма "Триггеры"

Для каждого типа данных можно создать свой набор триггеров. На вкладке Типы данных/Дополнительно располагается кнопка "Триггеры" при помощи которой можно осуществить настройку конкретного триггера. Для этого необходимо кликнуть по этой кнопке мышкой и в открывшейся таблице (рис. 4.7.2.6.2.) нажать на кнопку "Добавить" - (1), расположенную в левом верхнем углу, при этом откроется форма в которой можно осуществить конкретную настройку атрибутов триггера.

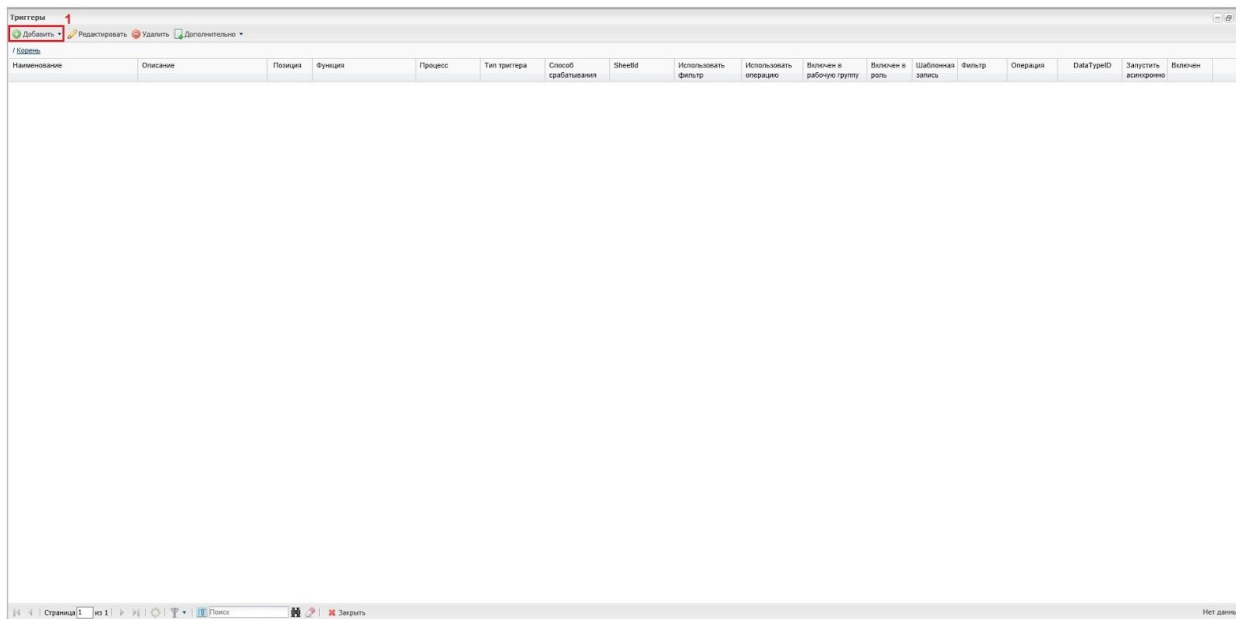


Рис. 4.7.2.6.2. Добавление триггера

Данная форма (рис. 4.7.2.6.3.) содержит обязательные поля: "Наименование", "Позиция", "Тип триггера", "Способ срабатывания". Поле "Наименование", в котором задаётся уникальное имя триггера, позволяющее однозначно его идентифицировать - (1). Поле "Описание" может содержать текстовое описание триггера выполненное с целью повышения удобства его дальнейшего использования - (2). Поле "Позиция" - (3) указывается очередность (порядок) срабатывания нескольких триггеров при наступлении определенного события (чем меньше позиция, тем раньше будет срабатывать триггер, чем больше тем позже), если событие вызывает более одного триггера и которое позволяет избежать коллизий.

Способ срабатывания - (7) триггера (позволяет указать, что будет представлять из себя триггер: функцию либо процесс). Функция задается в поле "Функция", данное поле позволяет перейти в меню выбора/создания/редактирования функции, которую выполнит триггер - (4), а процесс, задается в поле "Процесс", данное поле позволяет перейти в меню выбора/создания/редактирования процесса, который выполнит триггер - (5). Функция представляет собой некую программу (последовательность действий), которая запускается. При выборе способа срабатывания с помощью функции в поле "Функция" кликнув на кнопку добавить (обозначена знаком "+") открывается таблица типа данных "CoreCustomFunction" в которой присутствует наименование функции и код обработчика. Если рассмотреть одну из наиболее простых функций - функцию "print", то в поле "Тип функции" откроется выпадающий список с языком на котором написан обработчик, это

"python" или "groovy". Для более сложных функций код может быть многострочным. Его можно "прокрутить" в поле "Обработчик". Вдобавок код можно просмотреть и отредактировать (при необходимости) при помощи интегрированного расширенного редактора. В этом редакторе есть кнопка переключения режима, которая в свою очередь позволяет отображать код в структурированном и удобном для редактирования виде.

В поле "Тип триггера" - (6) задан выпадающий список позволяет выбрать один из доступных типов триггеров, которые в свою очередь можно разделить на две группы по времени выполнения: before («до») и after («после») вызвавшего их события (подробнее см. приложение 3). Если выбрать какой-либо тип события, например "beforeSave", то в тот момент, когда пользователь нажмет на кнопку "Сохранить" и до того, как данные сохранятся в базе данных, будет выполнена определенная операция.

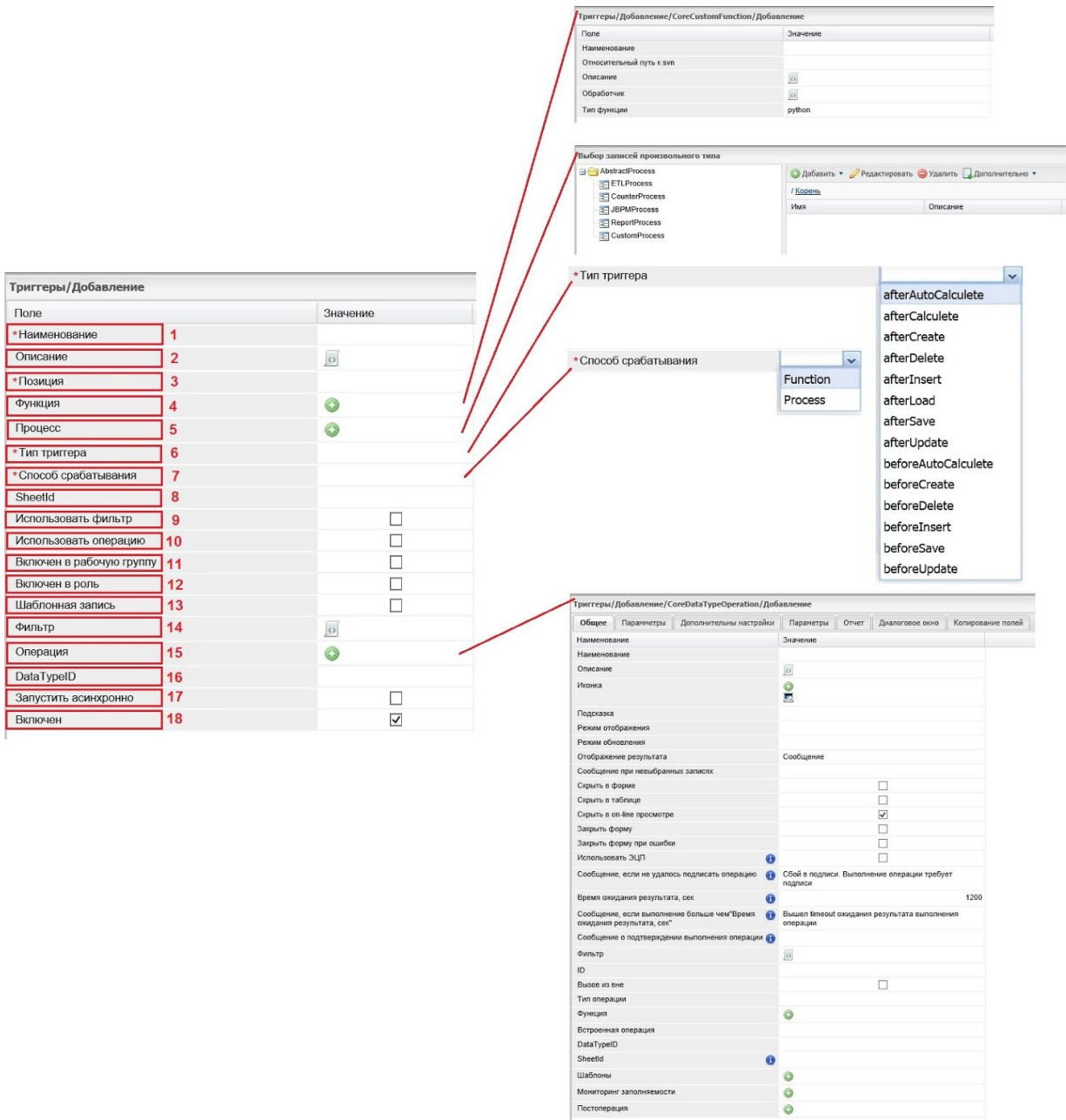


Рис. 4.7.2.6.3. Форма “Триггеры”

Поле “SheetId” позволяет указать таблицы (кроме редактируемого типа данных), для которых будет действовать данный триггер - (8). Поля “Использовать фильтр” -(9), “Использовать операцию” - (10), “Включен в рабочую группу” - (11), “Включен в роль” - (12), “Шаблонная запись” - (13) включают дополнительные настройки.

Поле “Фильтр” - (14) предоставляет возможность задать параметры фильтра. А поле “Операция” - (15) соответственно подключить операцию. Значение в поле “DataTypeId” - (16), устанавливается автоматически. Поле “Запустить асинхронно” - (17)

запускает триггер асинхронно. А поле “Включен” позволяет включить собранный триггер в редактируемом типе данных.

После добавления полей настройки триггера (набора действий, которые осуществляются при наступлении какого-нибудь события из списка событий) сохраняются. Таким образом можно создать набор триггеров, привязанную к определенному типу данных.

Форма "Операции"

Все операции в платформе Motiware Melody One подразделяются на два типа: системные и пользовательские.

Как правило системные операции отображаются в виде кнопок на панели управления и присутствуют там по умолчанию. Для настройки системных операций необходимо перейти в раздел «Операции», на панели управления развернуть выпадающий список «Дополнительно» и выбрать в нём пункт «Создать системные операции» и получить доступ к списку операций, которые добавляются постановкой значка в соответствующем поле (квадратике).

Кроме системных операций могут встречаться и пользовательские операции, которые представляют собой определенный код выполняемый при нажатии на соответствующую кнопку. К примеру, такой пользовательской операцией в системе электронного документооборота может быть отправка документа на согласование или визирование.

Для настройки пользовательских операций необходимо перейти в раздел «Операции» (см. рис. 4.7.2.6.4.) и панели управления нажать кнопку «Добавить» - (1).

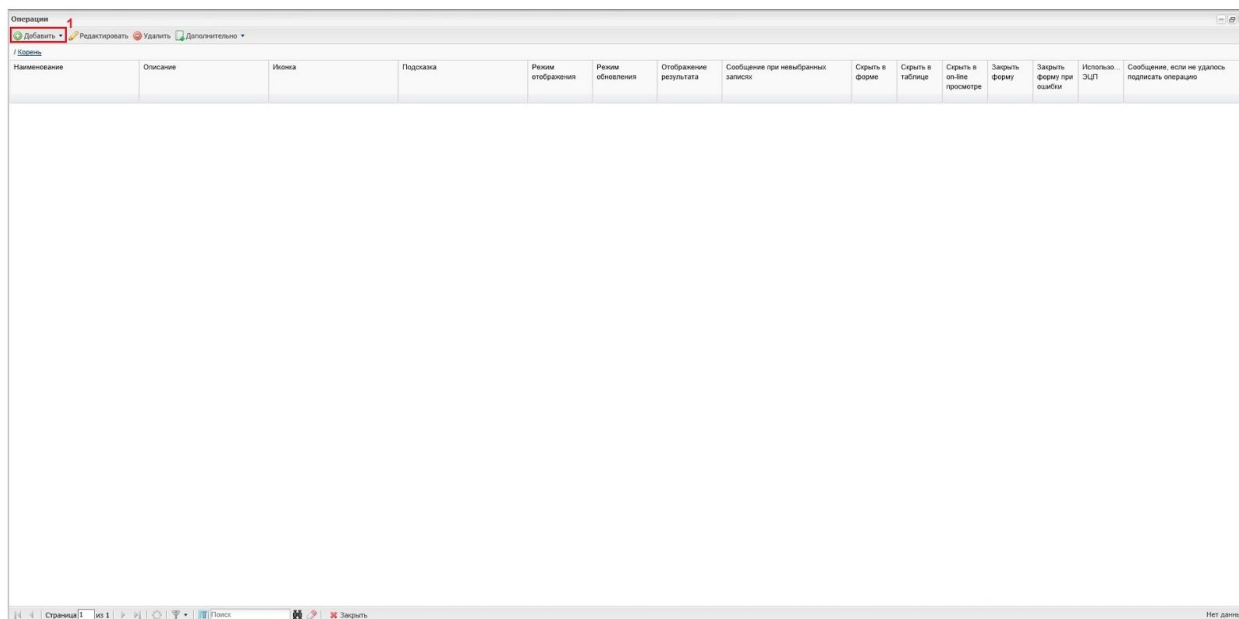


Рис. 4.7.2.6.4. Добавление операции

При этом форме создания пользовательской операции присутствуют следующие вкладки: “Общее”, “Параметры”, “Дополнительные настройки”, “Параметры”, “Отчет”. “Диалоговое окно”, “Копирование полей”.

Вкладка “Общее”, как показано на рисунке 4.7.2.6.5. содержит следующие поля:

1. Наименование - (1), в котором указывается имя операции и которое должно быть уникальным;
2. Описание - (2), текстовое описание операции;
3. Иконка - (3), изображение (16x16), которое будет размещено на кнопке вызова операции;
4. Подсказка - (4), короткое текстовое сообщение, которое будет высвечиваться при наведении курсора на кнопку вызова операции;
5. Режим отображения - (5), которое задает внешний вид средства вызова операции:
 - выпадающий список;
 - кнопка;
 - не выбрано;
6. Режим обновления - (6), поле которое определяет, для каких записей будет выполняться операция:
 - Все записи;
 - Все выделенные записи;
 - Только выделенные записи;
 - Не выбрано;
7. Отображение результата - (7), поле в котором задается способ отображения результата выполнения операции:
 - Ссылка;
 - Сообщение;
 - Фрейм.
8. Сообщение при невыбранных записях - (8), устанавливается текст сообщения, которое выведется при невыбранных записях;
9. Скрыть в форме - (9), поле в котором при установке флажка, операция не будет отображаться при отображении записи в виде формы;
10. Скрыть в таблице - (10), поле в котором при установке флажка, операция не будет отображаться в списке (таблице) записей;
11. Скрыть в online-просмотре - (11), поле в котором при установке флажка, операция не будет отображаться на панели управления в режиме online-просмотра;

12. Закрыть форму - (12), в случае установки флажка в этом поле, после выполнения операции, форма, в которой она была вызвана, закрывается;
13. Закрыть форму при ошибке - (13), в случае установки флажка, при возникновении ошибки при выполнении операции, форма, в которой она была вызвана, закрывается;
14. Использовать ЭЦП - (14), в случае установки флажка, становится доступным использование электронной цифровой подписи;
15. Сообщение, если не удалось выполнить операцию - (15), текстовое поле в которое заносится сообщение, которое будет выведено, если операция не выполниться должным образом;
16. Время ожидания результата - (16), поле устанавливающее время ожидания результата выполнения операции в секундах;
17. Сообщение, если выполнение займёт больше времени, чем указано в пункте 16 “Время ожидания результата, сек” - (17), поле в котором указывается сообщение, которое будет выведено на экран в случае превышения указанного времени ожидания результата;
18. Сообщение о подтверждении выполнения операции - (18), поле в которое заносится текст сообщения, которое выведется на экран для подтверждения выполнения операции;
19. Фильтр - (19), поле для настройки фильтра, который будет определять, для каких записей будет доступно использование операции;
20. ID - (20), идентификатор операции;
21. Вызов извне - (21), поле установка флага в котором, делает возможным вызов. операции извне;
22. Тип операции - (22)
23. Функция - (23), поле позволяет создать/настроить функцию, которая выполнится в результате вызова операции (настройка функции аналогична настройке функции в при описании формы “Триггеры”);
24. Встроенная операция, Позволяет указать операцию, которая станет составной частью данной операции.
25. DataTypeId - (25), значение поля устанавливается автоматически.
26. SheetId - (26), поле в котором можно выбрать таблицы, для которых будет доступно использование операции.
27. Шаблоны - (27), поле для выбора шаблона.
28. Мониторинг заполняемости - (28).
29. Постоперация - (29), поле в котором устанавливается операция, которая будет выполнена после завершения настраиваемой операции.

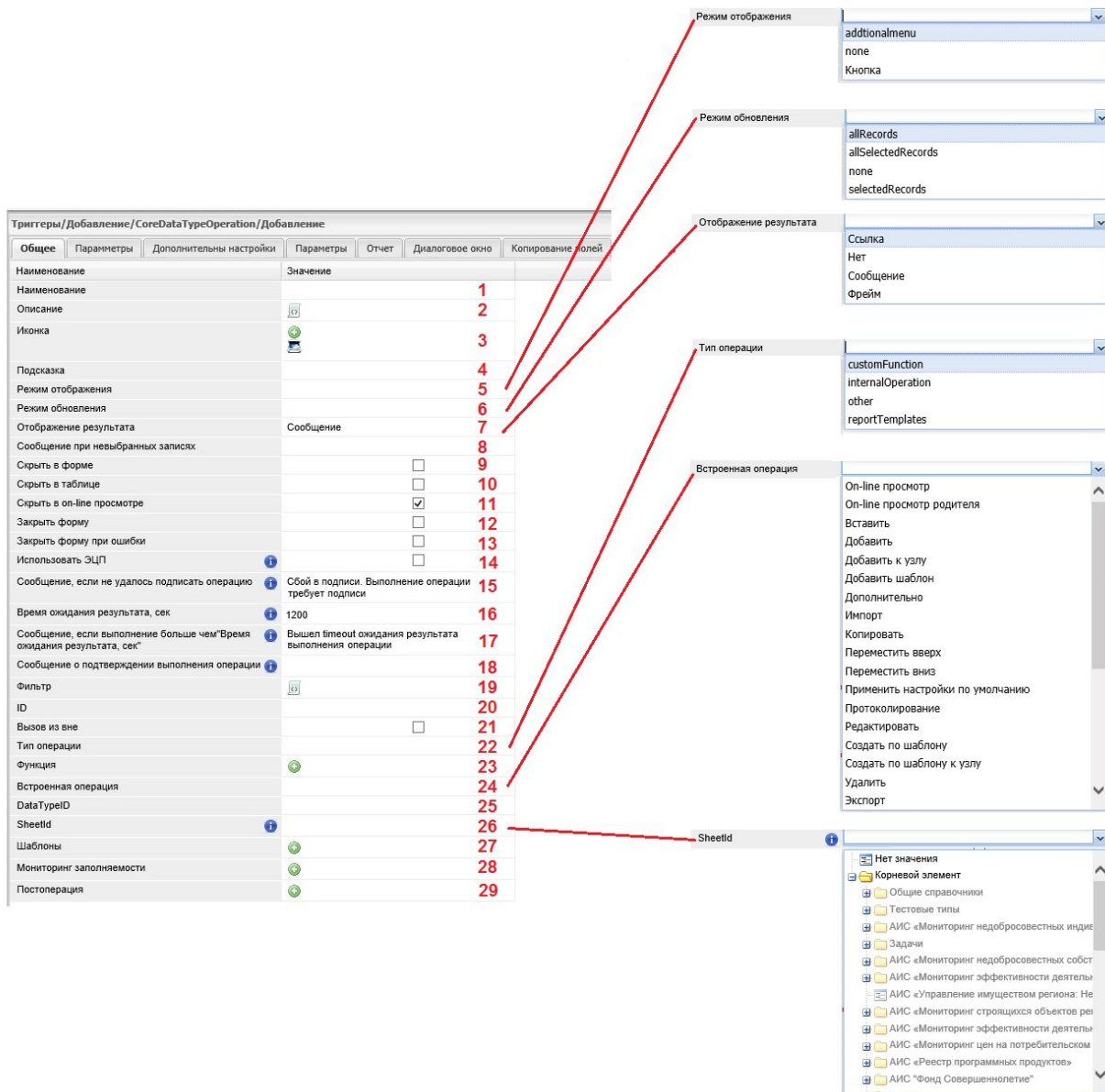


Рис. 4.7.2.6.5. Операции вкладка "Общие"

После настройки конкретной операции необходимо сохранить сделанные изменения при помощи кнопки "Сохранить".

Форма "Форматы"

Форма "Форматы" позволяет настраивать визуальную (цветовую) индикацию записей в таблицах. Существует два типа цветового форматирования: это подсветка фона записи и изменение цвета шрифта записи. Система позволяет настраивать условия при которых фон или шрифт записи будет отображаться определенным цветом. Это

бывает необходимо для индикации какого-либо состояния, непосредственно связанного с бизнес-процессом или бизнес-логикой. Например, в таблице со списком мероприятий, мероприятия с участием руководителя выделяются красным цветом шрифта, мероприятия с участием заместителей руководителя - зеленым, а все остальные - черным. Или розовым цветом фона подсвечиваются новые мероприятия (еще не просмотренные пользователем), желтым - просмотренные, зеленым - измененные. Изменение цвета происходит после выполнения какой-либо операции.

Для того, чтобы задать новый формат отображения записей, необходимо нажать на кнопку “Добавить” - (1) на вкладке “Форматы”, как изображено на рисунке 4.7.2.6.6.

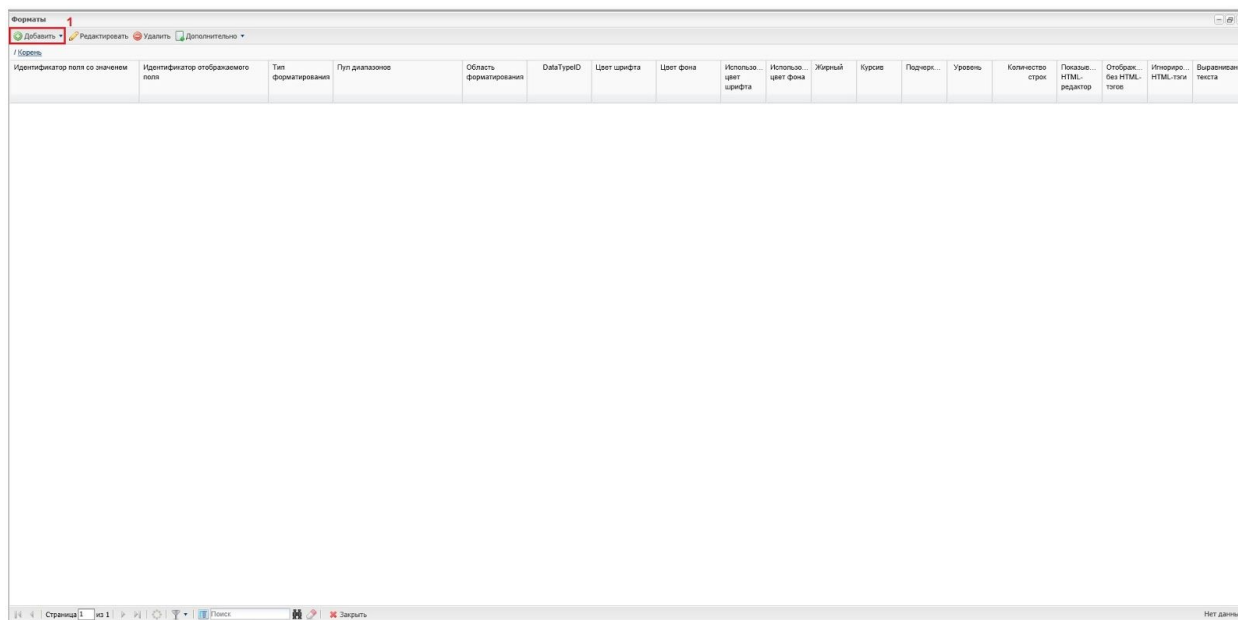


Рис. 4.7.2.6.7. Добавление формата отображения записей

При добавлении нового формата отображения записей открывается форма “Форматы/Добавление” рис. 4.7.2.6.7.

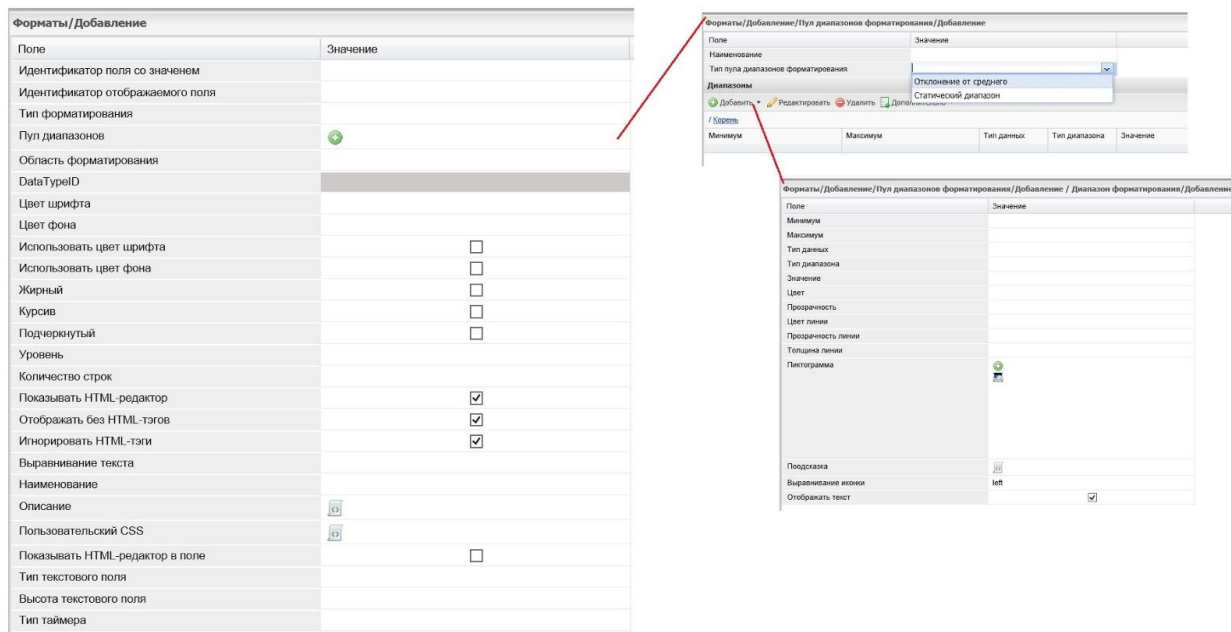


Рис. 4.7.2.6.7. Форма "Форматы"

При настройке цветового отображения во вкладке "Форматы" указывается идентификатор поля по которому настраивается условие, выбирается тип форматирования цвет текста или цвет фона, определяется область форматирования запись (во всей записи или строке) или поле (отдельно взятое поле или одна ячейка). Условия подсветки задаются в поле "Пул диапазона" в котором можно выбрать "Отклонение от среднего" или "Статический диапазон". Затем выполняется операция "Добавить" или "Редактировать", при этом открывается форма в которой задаются условия и диапазон форматирования. Например, если в поле тип данных задано - "Целое число" и оно принимает значение - 1 (в поле "Значение" задано целое число "1"), то цвет будет красным.

В свою очередь можно произвести настройку отображения, задав для значений минимум и максимум, т.е. меньше или больше определенного значения. Если в типе данных выбирается логический тип, указывается "True" (истинно) или "False" (ложно).

Кроме того, во вкладке "Дополнительно" (см. рис. 4.7.2.6.1.) присутствуют операции "Роли" (предназначена для настройки прав доступа), "Изменение кодировки полей", "Копирование полей" (из родительских типов данных в дочерние), "Индексирование и поиск" (позволяет переиндексировать записи), "Пересохранить данные" (при внесении изменений, например формулы вычисления значений, позволяет пересчитать и пересохранить все имеющиеся данные).

4.7.2.7. Вкладка "Ссылки"

На вкладке "Ссылки" (см. рис. 4.7.2.7.1.) можно настроить обратные ссылки, т.е. отображение всех типов данных на которые ссылается наш первый тип данных.

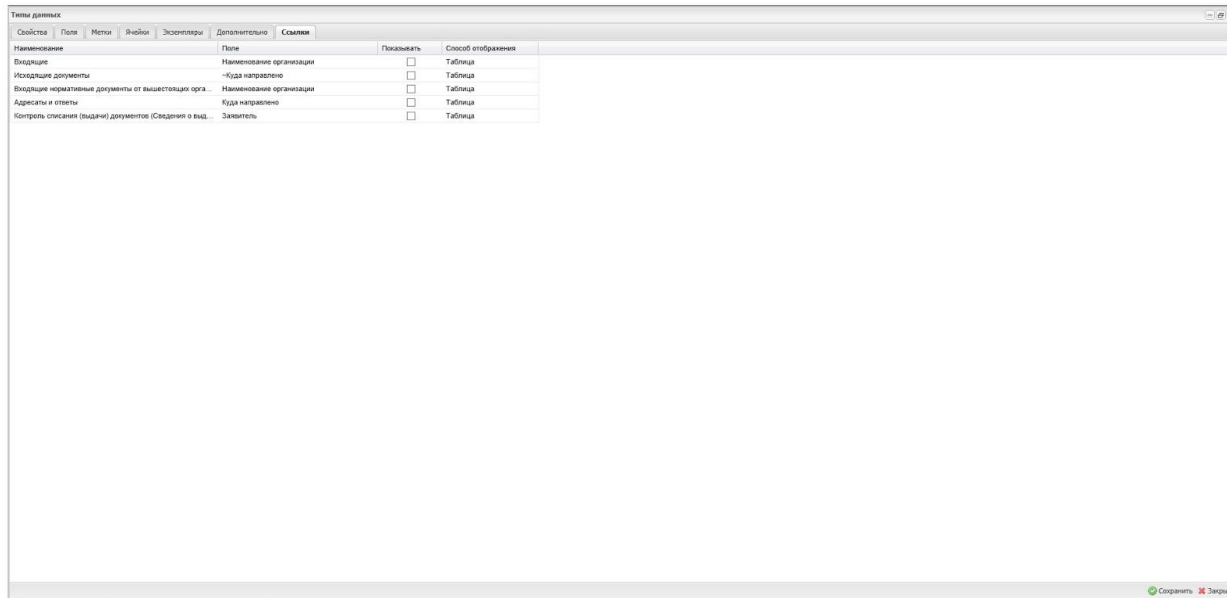


Рис. 4.7.2.7.1. Вкладка "Ссылки"

Кроме того обратную ссылку можно настроить (см. рис. 4.7.2.7.2.) на вкладке "Свойства" в Тип данных/Поле/Добавление - (1).

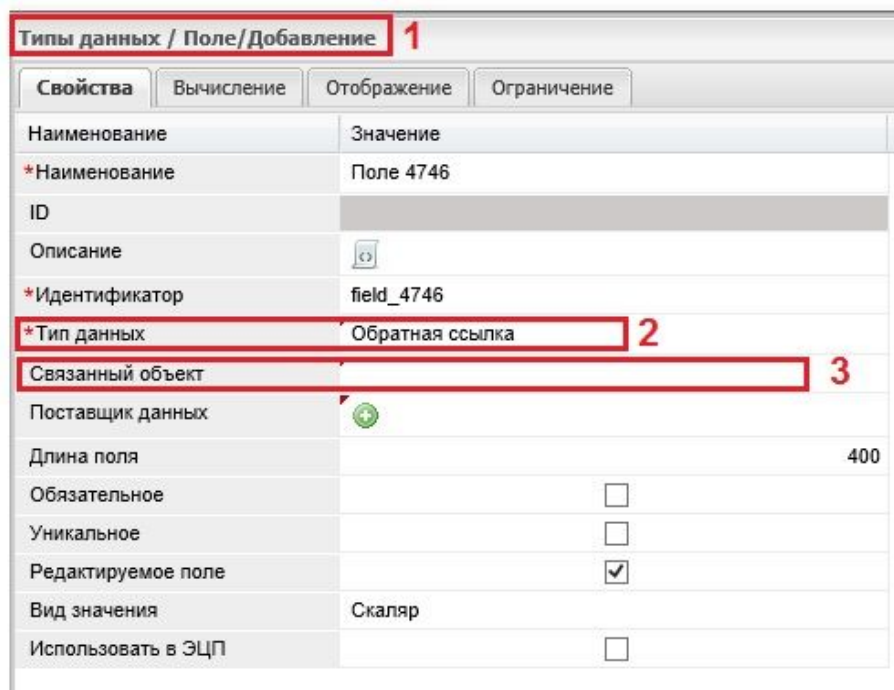


Рис. 4.7.7.2. Задание обратной ссылки

Для этого необходимо:

- в поле “Тип данных” задать значение "обратная ссылка" - (2);
- в поле “Связанный объект” - (3) выбрать в перечне объектов, тот объект который ссылается на этот объект и отобразить его или в виде таблицы или в виде ссылки;
- в поле “Показывать”, которое расположено на вкладке “Ссылки” (см. рис.4.7.2.7.1.) установить флаг напротив требуемого объекта.

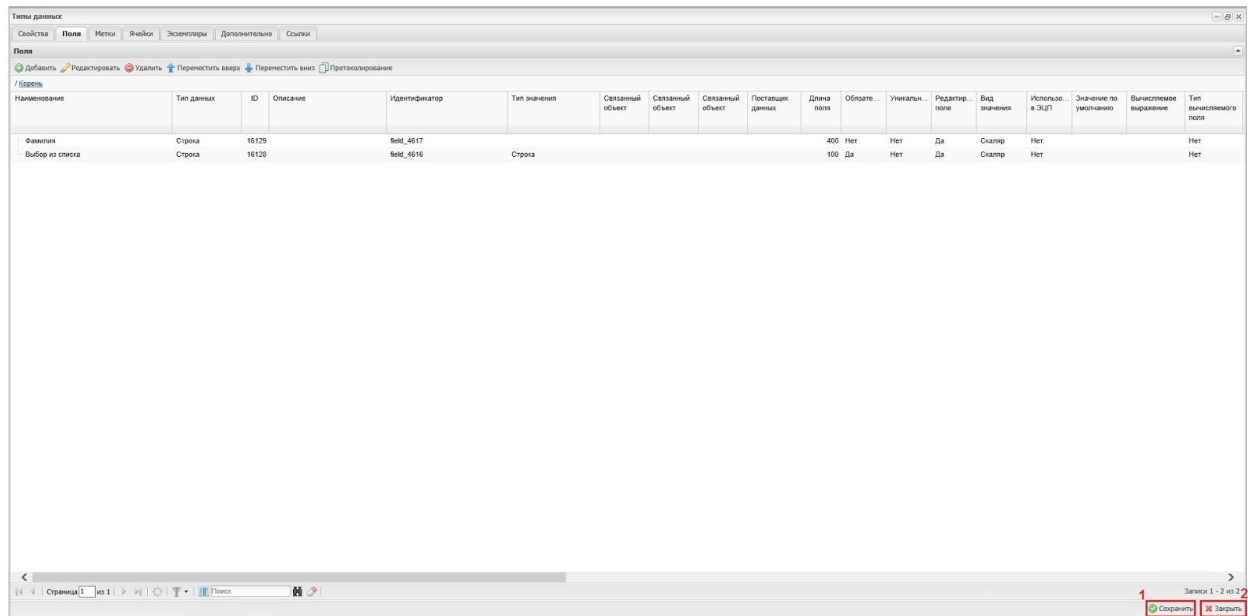


Рис. 4.7.7.3. Сохранение созданного (отредактированного) типа данных

После выполнения всех действий по созданию или редактированию конкретного типа данных необходимо сохранить все изменения, как показано на рисунке 4.2.7.3. при помощи кнопки “Сохранить” - (1) и закрыть, нажав на кнопку “Заккрыть” - (2).

4.8. Общие принципы создания моделей типов данных

При формировании рабочих моделей типов данных эффективно использовать следующие основные принципы:

1. Разумная нормализация данных. Например, если в исходной форме есть поле “ФИО”, то нужно разбивать это поле на три поля – “Фамилия”, “Имя”, “Отчество”. В тоже время поле “Дата рождения” разбивать на “Год”, “Месяц”, “День” - нерационально.
2. Типизация данных. Во всех случаях, когда исходя из семантики данных, есть возможность применения соответствующих типизованных полей, использования строковых полей нужно избегать. Например, поле “Год” – это целое число, соответственно, и поле должно быть задано целым числом.

3. Использование строковых полей вместо текстовых. Во всех случаях, когда известно, что размер данных в поле не превысит предел в 1000 символов (в текущей реализации), нужно применять поле типа “Строка”.
4. Использование списков и справочников. Во всех случаях, когда известно, что значение в данном поле может быть одним из нескольких допустимых, необходимо использовать списки или справочники. Списки используются для небольших перечней значений, о которых на этапе проектирования точно известно, что они изменяться не будут, и представляют собой значения одного из базовых типов. Например, поле “Квартал” имеет смысл сделать строковым со списком из четырех значений: первый квартал, второй квартал, третий квартал, четвертый квартал, и атрибуту “Только из списка” задать значение “Да”. В том случае, когда перечень значений заранее не известен, либо поле ссылается на комплексный объект, необходимо использовать справочники, т.е. поле должно быть типа “Ссылка”. Например, поле “Предприятие” имеет смысл сделать ссылкой на справочник “Юридические лица”.
5. Для полей, которые необходимы для ввода, нужно задавать атрибут “Обязательное”.
6. Для всех полей, для которых можно вычислить исходное значение, нужно задавать атрибут “Значение по умолчанию”. Например, поле “Дата заявки” имеет смысл инициализировать текущей датой.
7. Для всех полей, для которых можно вычислить значение на основе значений других полей, нужно задавать атрибуты “Режим вычисления” и “Вычисляемое выражение”.
8. Для всех полей, которые должны инициализироваться автоматически и не должны изменяться пользователем, нужно задавать “Нет” в атрибуте “Редактируемое”. Например, поле “Дата заявки” инициализируется текущей датой, и пользователь не должен иметь возможности его изменить.

4.9. Использование диаграмм классов UML

Статическая модель может быть хорошо иллюстрирована классической ER-диаграммой, либо диаграммой классов UML.

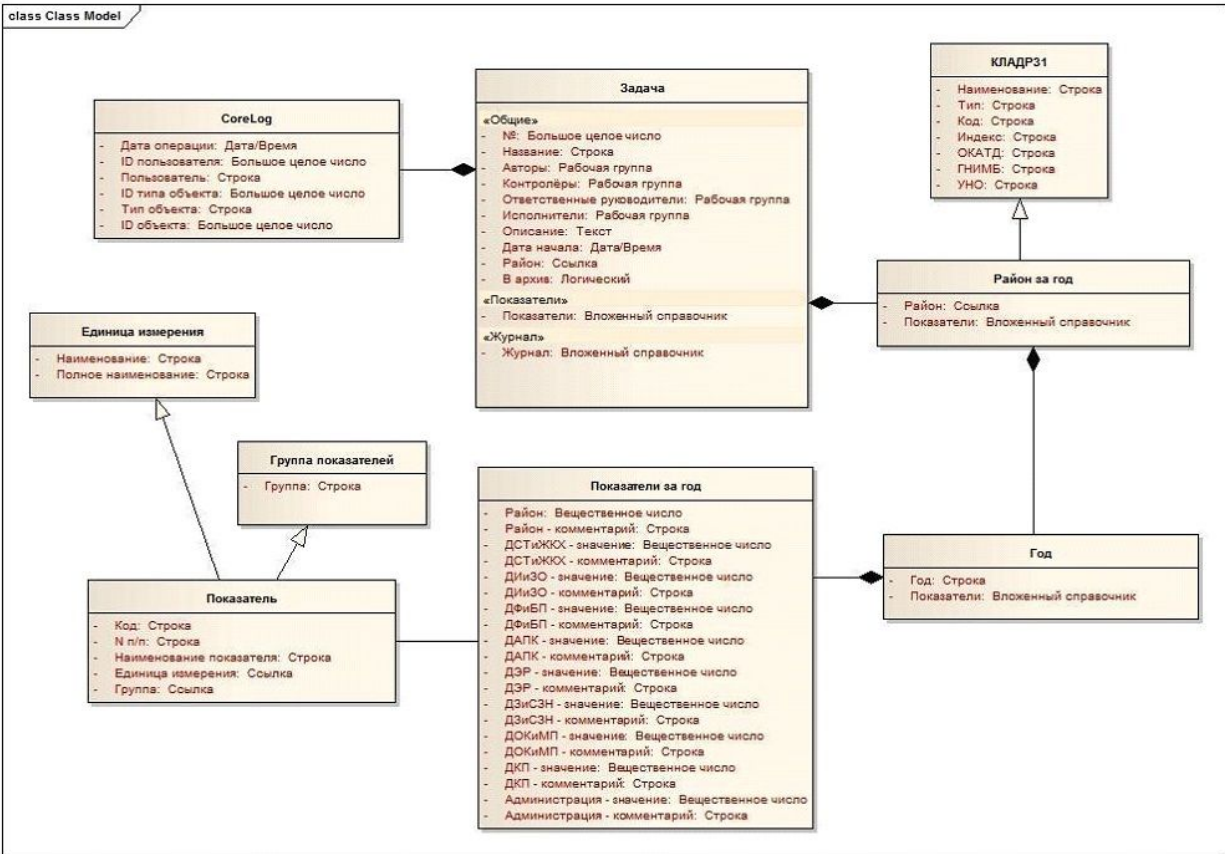

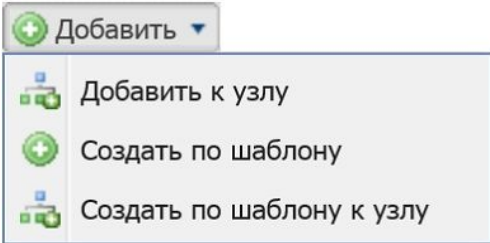

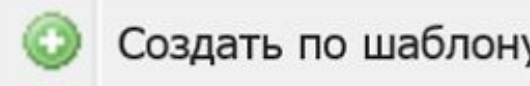
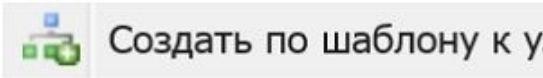















Рис. 4.9.1. Диаграмма классов модуля сбора готовых показателей с районов и городских округов в АИС «Оценка эффективности деятельности органов местного самоуправления»








На рисунке 4.9.1. представлена «Диаграмма классов модуля сбора готовых показателей с районов и городских округов в АИС «Оценка эффективности деятельности органов местного самоуправления»». Все прикладные сущности представляются в системе в виде типов данных. Свойства сущностей определяются полями типов данных, которые в свою очередь обладают типизацией.

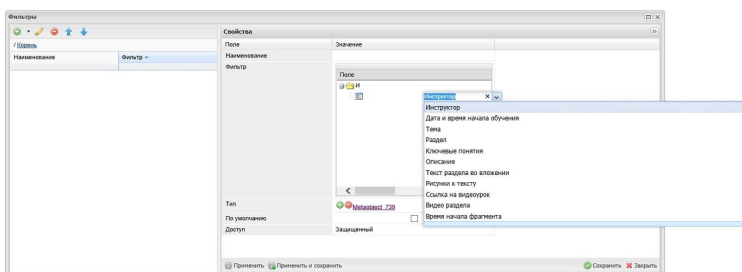
ПРИЛОЖЕНИЯ

Приложение 1. Кнопки для вызова системных операций





	Добавляет объект
	Кнопки, которые группируются с кнопкой добавить
	Добавляет объект к узлу
	Создает объект по шаблону
	Создает объект по шаблону и добавляет объект к узлу
	Редактирование объекта (формы, таблицы, записи)

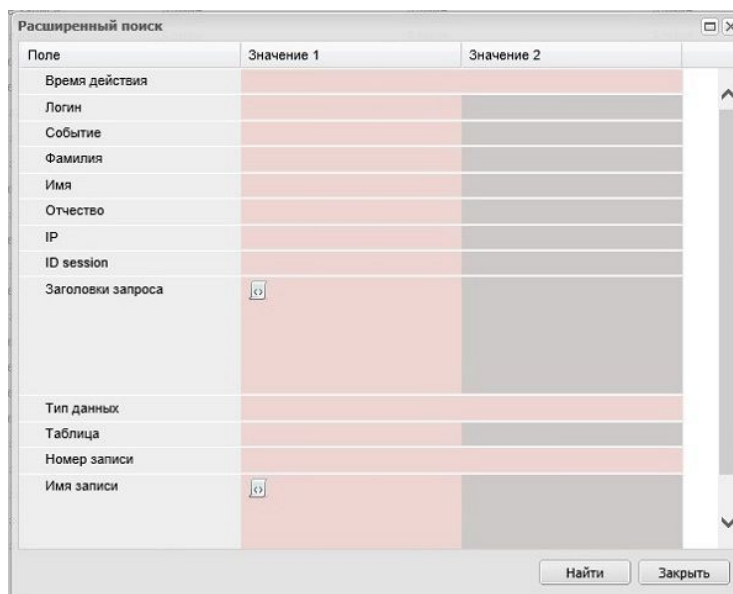
 Удалить	<p>Удаляет объект (таблицу, файл и т.п.)</p>
 Дополнительно ▾	<p>Предоставляет дополнительные возможности, которые отражаются в выпадающем списке</p>
 Дополнительно ▾  Копировать  Вставить  Импорт  Экспорт  Добавить шаблон	<p>Кнопки, которые группируются с кнопкой дополнительно</p>
 Копировать	<p>Копирование объекта</p>
 Вставить	<p>Добавление объекта</p>
 Импорт	<p>Импорт объекта</p>
 Экспорт	<p>Экспорт объекта</p>

 Добавить шаблон	Добавление шаблона
 Сохранить	Сохранение информации в окне рабочей области (в форме, в таблице и т.п.)
 Заккрыть	Закрытие окна (таблицы, формы)
 Выгрузить в Excel	Выгрузка таблицы в Excel
	Обновить информацию в окне
	Очистить окно
	Настройка фильтра
















Настройка фильтра







 Применить	Применить настроенную операцию фильтрации без сохранения
 Применить и сохранить	Применить настроенную операцию фильтрации и сохранить все параметры фильтрации
<input type="text" value="Поиск"/>	Поле ввода информации для поиска
	Искать
	Расширенный поиск



Окно для запуска расширенного поиска

 Страница 1 из 1	Панель навигации
	Сканирование документа
 Данные	Загрузка данных в таблицу
	Добавление географических координат
<div style="text-align: center;">  </div> <p>Выбор места расположения</p>	
	Каталог (тип данных)
	Запуск окна расширенного редактора

	<p>Вызов окна подсказки</p>
	<p>Добавить файл, таблицу</p>
	<p>Удалить файл, таблицу</p>
	<p>Открыть файл для просмотра</p>
 <p>Переместить вверх</p>	<p>Переместить выделенную строку в таблице на одну позицию вверх</p>
 <p>Переместить вниз</p>	<p>Переместить выделенную строку в таблице на одну позицию вниз</p>
 <p>Применить настройки</p>	<p>Сбросить настройки текущей таблицы для всех пользователей</p>
 <p>Протоколирование</p>	<p>Выполнить протоколирование</p>
 <p>Запустить отчет</p>	<p>Запуск формирования отчета</p>

 Редактор отчета	Вызов редактора отчетов
 Просмотр отчета	Просмотр отчета
 Автозаполнение	Выполнить автозаполнение
 Запустить процесс	Запуск процесса (ETL)
	Вход в систему
 Выход	Выход из системы

Приложение 2. Системные типы данных

Ветка типа данных	Описание содержимого
Core	ядро системы, эта ветка содержит основные базовые типы данных
Core / CoreAccess / CoreRole	ветка содержит перечень ролей без привязки к конкретному назначению роли
Core / CoreCounters	ветка содержит перечень имеющихся в системе настраиваемых счетчиков
Core / CoreCounters / CoreCounter	тип данных для настройки счетчиков
CoreCounters / CoreCounterRule	тип данных для настройки правил работы счетчиков
Core / CoreCounters / Закладки	тип данных для удобства отображения счетчиков и правил
Core / CoreCustomFunction	тип данных, содержащий все системные функции для кнопок, триггеров и других системных типов. В экземпляре содержатся всевозможные скрипты написанные на языке Python или Groovy
Core / CoreCustomOperation	ветка содержащая тип данных для пользовательских операций
Core / CoreCustomOperation / CoreDataTypeOperation	тип данных, содержащий набор системных пользовательских операций, например, такие как “Добавить к узлу” и “Редактировать”
Core / CoreData	ветка для хранения разнотипных данных
Core / CoreData / Папки объектов	тип данных, позволяющей группировать какие либо данные по папкам

Core / CoreData / Поставщик данных	объект хранящий информацию о способе получения данных
Core / CoreExport	тип данных, содержащий информацию о способе осуществления импорта данных из таблиц созданных в excel.
Core / CoreInterface	тип данных для отображения интерфейсных данных системы (отображение меню, отображение таблиц, отображение формы)
Core / CoreInterface / CoreFormat	ветка для хранения данных об отображении формата полей как в форме так и в таблице (фон, подсветка полей, шрифт и т. д.)
Core / CoreInterface / CoreFormat / Диапазон форматирования	позволяет настраивать форматирование в зависимости от диапазона
Core / CoreInterface / CoreFormat / Пул диапазонов форматирования	настройка формата в зависимости от пула, на пример использования формата Требуется настроить ранжирование от "0" к "100" при этом "0" имеет цвет красный, "100" - зеленый, а 55 будет иметь цвет между красным и зеленым, ближе к желтому (присвоится расчетно)
Core / CoreInterface / CoreFormat / Формат поля	типы данных, определяющие формат поля
Core / CoreInterface / CoreFormat / Формат поля / CoreIntermediateFieldFormat	промежуточный тип данных между CoreDataView и CoreFieldFormat
Core / CoreInterface / CoreMainMenu	тип данных настройки меню
Core / CoreInterface / CoreMasterDetailViewLevel	тип данных для отображения уровней пункта меню
Core / CoreInterface / CoreMenuCounters	тип данных для добавлении счетчиков пункта меню

Core / CoreInterface / CoreMenuRole	тип данных для добавления ролей пунктов меню
Core / CoreInterface / CoreMenuSettings	тип данных, содержащий информацию о самом объекте "Пункт меню"
Core / CoreInterface / Представление	ветка для хранения типов данных о представлении данных
Core /CoreInterface / Представление / CoreDepartmentUserView	тип данных определяющий отображение подразделений и пользователей
Core / CoreInterface / Представление / CoreFormView	тип данных определяющий поведение всех форм
Core /CoreInterface /Представление /CoreGridView	тип данных табличного отображения (представления) данных
Core / CoreInterface / Представление / CoreGridWithFoldersView	тип данных определяющий отображение в левой части экрана только тех пользовательских папок, которые он сам себе добавил
Core / CoreInterface / Представление / CoreMasterDetailView	тип данных определяющий отображение ссылки на объект не на прямую а через другие типы данных, данное отображение осуществляется не древовидно, а путем смены окон.
Core /CoreObjects	узел, содержит перечень объектов системы которые являются и системными и "пользовательскими" одновременно.
Core / CoreObjects / CoreDataViewOperation - Core /CoreObjects / CoreEnumeration	тип данных для хранения всех имеющихся в системе перечислений
Core / CoreObjects / CoreEnumerationInt	тип данных для хранения перечислений в которых для записи используется не строка а целое число
Core / CoreObjects / CoreLog	тип данных системы протоколирования

Core / CoreObjects / CoreLogParents	тип данных для связи при протоколировании вложенных объектов
Core / CoreObjects / Метка	атомарный тип вида “метка” для формирования самого типа данных
Core / CoreObjects / Поле	атомарный тип вида “поле” для формирования самого типа данных
Core / CoreObjects / Тип данных	непосредственно тип данных
Core / CoreObjects / Шаблоны объектов	хранение настроек для шаблонных записей
Core / CoreObjects / Экземпляры	атомарный тип вида “экземпляр” для формирования самого типа данных
Core / CoreObjects / Ячейка	атомарный тип вида “ячейка” для формирования самого типа данных
Core / CoreTrigger	перечень возможных триггеров для объектов
Core / CoreUsers	ветка, содержащая информацию о настройках пользователя как объекта
Core / CoreUsers / ChangePassword	настройка смены пароля у типа данных “пользователь”
CoreUsers / CoreContact	настройка полей контактной информации для типа данных “пользователь”
Core / CoreUsers / Адрес	настройка поля адреса для типа данных “пользователь”
Core / CoreUsers / Закладки	настройка закладок для для типа данных “пользователь”
Core / CoreUsers / Заместители	настройка данных по заместителям для типа данных “пользователь”

Core / CoreUsers / Подразделение	настройка данных подразделения для типа данных “пользователь”
Core / CoreUsers / Пользователь	тип данных “пользователь”
Core / CoreUsers / Протоколирование действий пользователя	настройка данных для протоколирования действий пользователя
Core / CoreWorkCalendars	узел для хранения настроек календаря и периодов для работы с ним
Core / CoreWorkCalendars / CoreWorkCalendar	тип данных рабочего календаря
Core / CoreWorkCalendars / CoreWorkCalendarException	тип данных
Core / CoreWorkCalendars/ CoreWorkPeriod	тип данных для хранения рабочего периода
Core/CoreWorkCalendars/ CoreWorkPeriodPart	тип данных для хранения частей периодов
Core / Закладки	тип данных для хранения закладок системных типов данных
DataFlow (движение данных)	узел для хранения информации о процессах
DataFlow/ AbstractProcess	тип данных “абстрактный процесс”
DataFlow/AbstractProcess/ CounterProcess	типы данных процесса для нумераторов, используется для сброса счетчика
DataFlow / AbstractProcess / CustomProcess	тип данных для вызова кода, написанного на языке Python
DataFlow / AbstractProcess / ETLProcess	тип данных ETL процесса

DataFlow / AbstractProcess / JBPMProcess	тип данных JBPM процесса
DataFlow / AbstractProcess / ReportProcess	тип данных, предназначенный для запуска отчетов
DataFlow / ETLAbstractActivity	тип данных активностей для ETL процесса
DataFlow / ETLAbstractActivity / ETLInputAbstractActivity	типы данных для получения данных из других сущностей
DataFlow / ETLAbstractActivity / ETLM3ApiCallAbstractActivity	тип данных для API из системы оперативного управления компанией “Мотив”
DataFlow / ETLAbstractActivity / ETLSelfStoreAbstractActivity	тип данных для работы с сохранением полученных либо преобразованных типов данных
DataFlow / ETLAbstractActivity / ETLSelfStoreAbstractActivity / ETLSelfStoreGetTopicalityActivity	тип данных для получение актуальности
DataFlow / ETLAbstractActivity / ETLSelfStoreAbstractActivity / ETLSelfStoreSaveActivity	тип данных для сохранения типов данных (с сохранением ссылок)
DataFlow / ETLAbstractActivity / ETLSelfStoreAbstractActivity / ETLSelfStoreSetTopicalityActivity	тип данных для установки актуальности активности
DataFlow / ETLAbstractActivity / ETLTransformAbstractActivity	тип данных процесса для изменения данных
DataFlow / ETLAbstractActivity / ETLTransformAbstractActivity / ETLTransformReplaceStringActivity	тип данных для преобразования путем замены

DataFlow/ETLAbstractActivity/ETLTransformAbstractActivity/ETLTransformRowToColumnActivity	тип данных для замены строк на столбцы и наоборот
DataFlow/ETLAbstractActivity/ETLTransformAbstractActivity / ETLTransformStreamEncodingActivity	тип данных для изменения кодировки
DataFlow / ETLAbstractDBConnection	тип данных для хранения настроек подключения к различным базам данных
DataFlow/ ETLAbstractDBConnection / FirebirdDBConnection	тип данных для подключения к базе данных firebird
DataFlow / HType	узел для хранения вспомогательных типов для ETL активностей
DataFlow / Узлы Jbpm процессов	узлы активностей Jbpm процессов
Forms	узел хранения “жестких” форм настроек
Forms / motiw4.properties.xml	тип данных для подключения текущей площадки к самой базе данных
Forms / Системные операции	типы данных содержащие поля форм для системных операций; сами же системные кнопки хранятся в Системные типы / Core / CoreData, используется для быстрого создания набора системных кнопок для какого либо типа данных
InternalTypes / Поле	тип данных для настроек перечня самых популярных свойств поля (так сказать “краткая” настройка поля”
SYS_REPORTS / Выполненные отчеты	тип данных ранее запущенных отчетов с прикрепленным полученного файла по результатам его (отчета) выполнения

SYS_REPORTS/ Разделы карточек	типы данных для настройки карточки шаблона отчета (для выполненных отчетов)
Отчеты	узел хранения настроек отчетов
Отчеты / Источники данных	виды источников данных
Отчеты / Параметры отчетов	тип данных для ввода параметров отчета
Отчеты / Разделы карточек	тип данных используемый в шаблоне отчетов
Отчеты / Тип диаграмм	тип используемой диаграммы
Отчеты / Тип отчета	тип выходного файла отчета
Отчеты / Шаблоны отчетов	тип шаблона отчета
Планировщик	узел хранения данных по планировщику
Планировщик / ShedulerTask	тип хранимых данных задач
Планировщик / CronExpression Планировщик / ShedulerTas	тип данных для настроек запуска и ссылка на тот процесс, который будет выполняться (на один из пяти процессов, находящихся в типе данных AbstractProcess)
Планировщик / Timer	тип данных для установки таймера
Планировщик / Закладки	тип данных для хранения данных планировщика
CommonTypes	ветка содержит перечень вспомогательных типов; эти типы предназначены для формирования других типов. Например, тип данных "Графические типы" служит для добавления к системным кнопкам картинок

Приложение 3. Типы триггеров

В платформе Motiware Melody One используются следующие системные триггеры:

- **afterAutoCalculete** – в результате срабатывания триггера, после завершения выполнения операции Автоматического вычисления (AutoCalculete), будет выполнено действие либо совокупность действий заданных в триггере
- **afterCalculete** – в результате срабатывания триггера, после завершения выполнения операции Вычисления (Calculete), будет выполнено действие либо совокупность действий заданных в триггере
- **afterCreate** – в результате срабатывания триггера, после завершения выполнения операции Создания записи(Create), будет выполнено действие либо совокупность действий заданных в триггере
- **afterDelete** – в результате срабатывания триггера, после завершения выполнения операции Удаления записи (Delete), будет выполнено действие либо совокупность действий заданных в триггере
- **afterInsert** – в результате срабатывания триггера, после завершения выполнения операции Вставки записи (Insert), будет выполнено действие либо совокупность действий заданных в триггере
- **afterLoad** – в результате срабатывания триггера, после завершения выполнения операции Загрузки (Load), будет выполнено действие либо совокупность действий заданных в триггере
- **afterSave** – в результате срабатывания триггера, после завершения выполнения операции Сохранения (Save), будет выполнено действие либо совокупность действий заданных в триггере
- **afterUpdate** – в результате срабатывания триггера, после завершения выполнения операции Обновления (Update), будет выполнено действие либо совокупность действий заданных в триггере
- **beforeAutoCalculete** – в результате срабатывания триггера, до выполнения операции Автоматического вычисления (AutoCalculete), будет выполнено действие либо совокупность действий заданных в триггере. После этого будет выполнена операция Автоматического вычисления (AutoCalculete).
- **beforeCreate** – в результате срабатывания триггера, до выполнения операции Создания записи (Create), будет выполнено действие либо совокупность действий заданных в триггере. После этого будет выполнена операция Создания записи (Create).

- **beforeDelete** – в результате срабатывания триггера, до выполнения операции Удаления записи (Delete), будет выполнено действие либо совокупность действий заданных в триггере. После этого будет выполнена операция Удаления записи (Delete).
- **beforeInsert** – в результате срабатывания триггера, до выполнения операции Вставки записи (Insert), будет выполнено действие либо совокупность действий заданных в триггере. После этого будет выполнена операция Вставки записи (Insert).
- **beforeSave** – в результате срабатывания триггера, до выполнения операции Сохранения (Save), будет выполнено действие либо совокупность действий заданных в триггере. После этого будет выполнена операция Сохранения (Save).
- **beforeUpdate** – в результате срабатывания триггера, до выполнения операции Обновления (Update), будет выполнено действие либо совокупность действий заданных в триггере. После этого будет выполнена операция Обновления (Update).

Приложение 4. Описание функций

1. **setLinkValue**
2. **2 getValue**
3. **getValueDepFieldFromWGUser**
4. **setLinkValues**
5. **getAggregationCount**
6. **getAggregationSum**
7. **vl**
8. **vf**
9. **fields**
10. **getFieldValueFromLink**
11. **setValueLinkByLevel**
12. **ifExec**
13. **isPositive**
14. **PeriodOffset**
15. **dateOffset**
16. **getTermByFormat**
17. **getSumValueFromTreeGridByCondition**
18. **getValueFromParentDOSlow**
19. **getValueLinkFromAggregation**
20. **initWorkGroup**
21. **getAggregationSumL2**
22. **getValueByLinktoCriteria**

23. getValueFromSubdictByFieldMaxmin

24. getMaxCoastValueFromTreeGrid

25. getMessageActionByUser

26. getValueFromPath

27. getValueFromOwner

28. getLinkToParent

29. trustCalc

30. calcAfterEdit

31. linkValueVirtual

32. countByPath

33. linkRecords

34. coreCalculationByCondition

35. valueByFilter

36. getPhotoByUser

37. getBackLinks

38. getMaxValue()

1. setLinkValue

setLinkValue(fieldName, lookupValue, fieldCache=None) Функция для инициализации поля типа ссылка значением.

fieldName - поле из справочника, где надо найти value

lookupValue - значение

fieldCache - поле для отображения, если равно пустой строке, то вычисляется кэш по умолчанию. Если None - то найденное значение

2. getValue

getValue(iden, id, rowId, field) Для получения значения поля, зная id записи

iden - Идентификатор типа данных

id - id записи

rowId - номер строки

field - идентификатор поля

3. **getValueDepFieldFromWGUser**

getValueDepFieldFromWGUser(fieldWG, field, delimiter=', ') Для получения значения поля департамента, для пользователей из рабочей группы

fieldWG - Идентификатор поля с рабочей группой

field- Идентификатор поля из департамента

delimiter - (опционально) разделитель для записей, если несколько уникальных результатов

4. **setLinkValues**

setLinkValues(fieldName, lookupValues, fieldCache=None) Функция для инициализации поля типа ссылка значением.

fieldName - поле из справочника, где надо найти любые из lookupValues

lookupValues - список значений

fieldCache - поле для отображения, если равно пустой строке, то вычисляется кэш по умолчанию. Если None - то найденное значение

Пример 1: setLinkValues("Name", ["Name1", "Имя 2", "Name 3"]) - ссылка будет проинициализирована 3-мя ссылками: Name1, Имя 2 и Name 3

Пример 2: setLinkValues("Name", [vf("Role"), "Имя 2", "Name3"]) - ссылка будет проинициализирована 3-мя ссылками: значениям поля Role, Имя 2 и Name 3

5. **getAggregationCount**

getAggregationCount(aggregationField, aggregationLookupField, findCompare, condition="findValue is None or findValue >= findCompare") Возвращает количество значений агрегации по условию.

aggregationField - поле-агрегация у главного объекта

aggregationLookupField - поле самой агрегации, из которого берется значение для условия и устанавливается в findField

findCompare - значение для сравнения с findValue

condition - условие

6. **getAggregationSum**

getAggregationSum(aggregationField, aggregationLookupField, findCompare, aggregationValueField, condition="findValue is None or findValue >= findCompare")

Возвращает сумму значений агрегации по условию.

aggregationField - поле-агрегация у главного объекта

aggregationLookupField - поле самой агрегации, из которого берется значение для условия и устанавливается в findField

findCompare - значение для сравнения с findValue

aggregationValueField - значения, которое будет суммироваться по условию

condition - условие

7. vl

vl (labelIdentifier) - возвращает значение по идентификатору метки

labelIdentifier - идентификатор метки

8. vf

vf(fieldIdentifier) - возвращает значение (список значений) поля

fieldIdentifier - идентификатор поля

Примеры:

vf('Data.name') - возвращает значение поля name по ссылке в поле Date

setLinkValue("DepartmentName",vf("Owner.supper.DepartmentName").get(0),"DepartmentName") - инициализирует поле типа "Ссылка" ссылкой на департамент пользователя, ссылка на которого указана в поле Owner.

9. fields

fields(rowNumber, fieldIdentifier) - возвращает значение по номеру метки и по идентификатору поля

rowNumber - порядковый номер метки

fieldIdentifier - идентификатор поля

10. getFieldValueFromLink

getFieldValueFromLink(rowId, field, link, fieldInLink) - возвращает значение поля по ссылке

rowId - номер метки

field - идентификатор поля типа ссылка

link - идентификатор типа данных, на которое ссылается field

fieldInLink - идентификатор поля, с которого будет брать значение

Вместо данной функции рекомендуем использовать новую - vf

11. setValueLinkByLevel

setValueLinkByLevel(fieldid, link, fieldlink, level) - возвращает ссылку (или значение ссылки)

на элемент справочника по уровню fieldid - Идентификатор поля ссылки link -

Идентификатор метаобъекта fieldlink - Поле отображение результата ссылки, или само

значение ссылки level - уровень в иерархии Примеры: setValueLinkByLevel('Settlement',

'Kladr', 'Name', 0) в результате - ссылка на корневой элемент выбранной записи для 'Settlement' (т.е. область/край) setValueLinkByLevel('Settlement', 'Kladr', 'Name', 1) в результате - ссылка элемент 2-го уровня для выбранной записи для 'Settlement' (т.е. район)

12. ifExec

ifExec(condition, resultThen, resultElse) - если condition == true - то результат resultThen, иначе resultElse

condition- условие

resultThen- строка, с кодом для значения по ветке true

resultElse- строка, с кодом для значения по ветке false

13. isPositive

isPositive(value, defaultValue = 0) - если value > 0, то возвращается value, иначе defaultValue

value - число, проверяемое на больше 0

defaultValue - (необязательное) значение, если value <= 0

14. PeriodOffset

PeriodOffset(StartDate, EndDate, Calendar, WorkPeriod, Direction) возвращает разницу рабочих периодов (WorkPeriod - дни, часы) между EndDate и StartDate, по календарю Calendar.

StartDate - идентификатор поля, на начала периода

EndDate - дата конца периода

Calendar - (опционально) идентификатор поля на календарь рабочего времени, или id календаря, если не задан - используется первый календарь

WorkPeriod - (опционально) идентификатор поля на рабочий период, или id рабочего периода, если не задан - используется "Основной период" календаря

Direction - (опционально)направление округления, 0 - математическое округление, 1 - округление в большую сторону, другое - округление в меньшую сторону

15. dateOffset

dateOffset(StartDate, Calendar, WorkPeriod, PeriodCount, Direction) возвращает дату смещенную от StartDate, по календарю Calendar, с рабочими периодами WorkPeriod, на PeriodCount периодов, в направлении Direction

StartDate - идентификатор поля на начала периода

Calendar - (опционально) идентификатор поля на календарь рабочего времени, или id календаря, если не задан - используется первый календарь

WorkPeriod - (опционально) идентификатор поля на рабочий период, или id рабочего периода, если не задан - используется "Основной период" календаря

PeriodCount - (опционально) количество периодов, на который нужно получить смещение

Direction - (опционально) направление перемещения (не используется)

16. **getTermByFormat**

getTermByFormat(date1, date2, format) возвращает разницу дат (date1 - date2) по заданному формату (формат пока не разработан и нужно передавать "days" - в днях).

date1 - дата начала

date2- дата окончания

format- формат

17. **getSumValueFromTreeGridByCondition**

getSumValueFromTreeGridByCondition(fieldInSubdict, fieldName, fieldNameCompare, findCompare) возвращает сумму значений для заданного поля вложенного справочника по условию.

fieldInSubdict- поле по которому будет суммироваться значения

fieldName - вложенный справочник

fieldNameCompare - поле вложенного справочника для сравнения

findCompare - значение для сравнения с fieldNameCompare

18. **getValueFromParentDOSlow**

getValueFromParentDOSlow(field) - возвращает значение поля в родительском объекте.

field - идентификатор поля

19. **getValueLinkFromAggregation**

getValueLinkFromAggregation(ownerMo, ownerField, mo, field) - формирует ссылки из всех полей ссылок из агрегации

ownerMo - идентификатор текущего типа данных

ownerField - идентификатор поля с агрегацией

mo - идентификатор типа данных агрегации

field - идентификатор поля в агрегации

20. **initWorkGroup**

initWorkGroup(users, departemnts, fieldCache="login", clear=True) - инициализирует поле, типа рабочая группа

users - массив пользователей, заключенные в одиночные кавычки, поиск пользователей осуществляется по полю login

departemnts - массив департаментов, заключенные в одиночные кавычки, поиск которых будет по имени.

fieldCache (не обязательное поле, по умолчанию login) - поле, отображаемое в качестве имени пользователя

clear (не обязательное поле, по умолчанию True) - очищать текущую рабочую группу.

Примеры:

```
initWorkGroup([], [u"Департамент"])
```

в рабочую группу попадает департамент "Департамент"

```
initWorkGroup(["vit"], [u"Департамент"])
```

в рабочую группу попадает департамент "Департамент" и пользователь с логином vit

```
initWorkGroup(["vit", "admin"], [u"Департамент"], "Name")
```

в рабочую группу попадает департамент "Департамент" и пользователи с логином vit и admin. В качестве отображаемого имени будет имена пользователя

```
initWorkGroup(["vit"], [u"Департамент"], "login", False)
```

в рабочую группу попадает департамент "Департамент" и пользователи с логином vit + те кто был добавлен в ручную.

```
initWorkGroup([UserManager.getCurrentUser().getRecord().getRow(0).get("login")], [])
```

инициализация рабочей группы текущим пользователем

21. **getAggregationSumL2**

```
getAggregationSumL2(link, aggregationField, aggregationLookupField, aggregationValueField, "findValue >= dataRecord.getRow(0).get('DateMonthStart') and findValue <= dataRecord.getRow(0).get('DateMonthEnd')")
```

 Возвращает сумму значений агрегации второго уровня по условию.

link - ссылка главного объекта

aggregationField - поле-агрегация у ссылаемого объекта

aggregationLookupField - поле самой агрегации, из которого берется значение для условия и устанавливается в findField

aggregationValueField - значения, которое будет суммироваться по условию

condition - условие

22. **getValueByLinktoCriteria**

```
getValueByLinktoCriteria(molden, fieldId, filter, fieldInLink, delimiter = ',')
```

Получение значений из ссылаемых на текущую запись ссылок

molDen - Идентификатор ссылаемого объекта

fieldId - Поле типа "Ссылка" или "Ссылка произвольно типа" на текущий объект

filter - Дополнительный фильтр записей

fieldInLink - Поле, с которого берутся значения

delimiter - (можно не указывать) параметр разделитель между результатами

23. **getValueFromSubdictByFieldMaxmin**

getValueFromSubdictByFieldMaxmin(fieldInSubdict, fieldName, fieldNameCompare, flagMaxMin, d=now()) Возвращает значение агрегации по условию min или max.

fieldInSubdict - результирующее поле агрегации

fieldName - агрегация

fieldNameCompare - поле агрегации, по которому производится сравнение

flagMaxMin, - min или max

d - дата до которой производится сравнение (для поля типа Date)

24. **getMaxCoastValueFromTreeGrid**

getMaxCoastValueFromTreeGrid(sheetId, ownerMetaobjectId, field, find, gde, to) Возвращает значение агрегации по условию max, а также поле gde == field

sheetId - id таблицы агрегации, для текущей записи можно получить metaobject.getMetafieldByIdentifier(<Идентификатор поля Агрегации>).getLinkObject()

ownerMetaobjectId - id типа данных, для которой эта агрегация, для текущей записи можно получить metaobject.getVersionId()

field - идентификатор поле, значение которого будет максимальным

find - значение сравниваемое на равенство в поле gde

gde - идентификатор поля, по которому определяется условия включения записи

to - идентификатор поля результата

25. **getMessageActionByUser**

getMessageActionByUser(id, key) - Получение текста сообщения для действия задач (фото, пользователь, сообщение...)

id - текущий пользователь

key - ключ записи с файлом таблицы "Графические файлы"

26. **getValueFromPath**

getValueFromPath(fieldIden, paths, isComplex, delimiter = ', ', filter = None)

Получает значение из "Агрегации", "Агрегации произвольного типа", "Ссылки", "Ссылки произвольного типа" или "Рабочая группа"

fieldIden - идентификатор поля

paths - список путей к полю результата, разделитель точка

isComplex - ключ, если результат сложное поле - true, иначе false

delimiter - (Опционально) для простых полей - разделитель

filter - (Опционально) строка фильтра для "Агрегации" и "Агрегации произвольного типа"

27. **getValueFromOwner**

getValueFromOwner(ownerDataRecord, field)

Получает значение из "Агрегации" или "Агрегации произвольного типа"

ownerDataRecord - родительская запись

field - идентификатор поля родительской записи

28. **getLinkToParent**

getLinkToParent(level = 1)

Инициализирует "Ссылку" и "Ссылку произвольного типа" на одного из родителей

level - (опционально) уровень глубины поиска родителя, по умолчанию - берется первый родитель. Например если объект на 2-м уровне, то нужно передавать число ≥ 2

29. **trustCalc**

trustCalc(value)

Переданное значение не будет установлено, если поле поменялось

30. **calcAfterEdit**

calcAfterEdit(field, fun)

Вычислять выражение после изменения поля field

field - идентификатор поля, при изменении которого запускает вычисление fun - строка-функция, вычисляемое выражение для текущего поля

31. **linkValueVirtual**

linkValueVirtual(linkField, value, defaultValue = "")

Получает значение по ссылке, можно использовать в виртуальных полях

linkField - идентификатор поля с ссылкой value - идентификатор поля значения

defaultValue - значения по умолчанию

32. **countByPath**

countByPath(paths, filter = None, defaultValue = 0):

Считает количество элементов составного объекта, находящемуся по пути path

paths - путь к составному объекту от текущего, разделенные точкой (.) filter - дополнительный фильтр для справочника defaultValue- значение по умолчанию

Пример:

```
- countByPath('Plan.Works', "" {"_constructor":"AdvancedCriteria", "operator":"and",  
"criteria":[{"fieldName":"PercentDone", "operator":"notEqual", "value":"100", "value2":""}]}"" ) ==  
0
```

True - все работы завершены

False - есть не завершённые

33. linkRecords

```
linkRecords(paths, field, maxCount = 3, delimiter = "  
", filter = None, defaultValue = "")
```

формирует ссылки на записи справочника paths

paths - путь к составному объекту от текущего, разделенные точкой (.) field - Поле, которое будет отображаться в качестве результата maxCount - Кол-во записей на строке delimiter - разделитель строк filter - дополнительный фильтр для справочника defaultValue- значение по умолчанию

34. coreCalculationByCondition

```
coreCalculationByCondition(recordKey, fieldKey = 'Role', value = None, fieldRez = 'Value')
```

Вызовов функции заданной в типе данных CoreCalculationByCondition, вызов берется первой записи, сортировка осуществляется по полю Pos

recordKey- Ключ поиска записи в CoreCalculationByCondition fieldKey- Поле ключа для дополнительной фильтрации, если не задана, берется поле Role value - Значение дополнительной фильтрации fieldKey, если не задана, берется роли текущего пользователя fieldRez - поле результата, если Expression - то выполняется python выражение записанное в этом поле

35. valueByFilter

```
valueByFilter(key)
```

Есть тип данных CoreCustomValueByFilter, у него поля: Ключ (Строка), Значение (Строка), Значение ссылки (произвольная ссылка), вес (Целое), Тип данных (Ссылка на тип данных), Фильтр (Текст).

В типе данных CoreCustomValueByFilter добавляется записи, с ключом key, тип данных на который нужно задать фильтр, и в каждом поле Текст - фильтр принятия результата. Если подходящих условий для записи несколько - берется значение с большим весом.

фильтр может быть json фильтром, или выражением на groovy, в качестве параметров передается metaobject, dataRecord, session, filterRecord
key - ключ в CoreCustomValueByFilter

36. **getPhotoByUser**

getPhotoByUser(id, userIden="User", messagelden="Message", datelden="DateCreate", photolden="Photo")

<p>Формирует сообщение как лента действий мотива.

- id картинки по умолчанию - userIden идентификатор поля, ссылка на пользователя - messagelden идентификатор поля сообщения - datelden идентификатор поля даты сообщения - photolden идентификатор поля в карточке пользователя, тип файл, 1-й файл - картинка пользователя

37. **getBackLinks**

getBackLinks(map) Возвращает ссылку на запись обратной ссылки

map - словарь - Тип данных : Поле

Примеры:

getBackLinks({'DocVnutr':'LinkDoc'})

38. **getMaxValue()**

getMaxValue()

Возвращает максимальное значение текущего поля в рамках агрегации

Примеры:

Long(getMaxValue()).longValue() + 1

Приложение 5. Виды таблиц

Таблица	Элемент	Содержание
Обычная таблица - DataTreeGrid	Верхний toolbar	Содержит кнопки, либо кнопки с выпадающим списком. Есть деление на системные и пользовательские. Если системные не приходят с сервера, отображаются по умолчанию (Добавить, Редактировать, Удалить и т.д.). Для CommonSettingsAdmin также добавляется кнопка "Применить настройки", при нажатии для всех пользователей сбрасываются настройки текущей таблицы.
	Путь текущей записи	Содержит путь выбранной записи, всегда начинается с "Корень", путь состоит из имен первой колонки (Отключается настройкой).
	Колонки	С возможностью объединения колонок. Колонки перетаскиваются, скрываются, выбирается сортировка. Все состояние сохраняется на сервере.
	Данные	Отображаются в виде дерева (слева появляется знак "+",

		<p>по нажатию на который под ним добавляются вложенные записи).</p> <p>Подсветка идет так же на клиенте (CoreFormat).</p> <p>Render'ы на отдельные типы элементов, кроме того определены (например для подписи появляется иконка проверки ЭЦП). DnD для изменения позиции записи, а также для изменения родителя.</p> <p>При двойном клике, в зависимости от настроек, вызывается операция, если не задано - операция редактирования.</p> <p>Также может отображаться ссылка на открытие записей (\$wnd.showFormView например, полный перечень и описание позже).</p> <p>Первая колонка может быть также ссылкой на открытие записи в отдельном окне.</p>
	<p>Нижний toolbar</p>	<p>Содержит постраничное управление. Вызов редактора фильтров, вызов редактора поиска, field поиска.</p> <p>Также нижний toolbar содержит специфичные кнопки, например для ETLProcess запуск процесса.</p>

Таблица выбора ссылки - LinkTreeGrid	Все свойства DataTreeGrid	Сам грид расположен в окне, в котором в нижней правой части есть кнопки "Выбрать" и "Отмена".
Таблица выбора ссылки произвольного типа - LinkOnRecordsSetTypeTreeGrid	Состоит из 2-х панелей, в левой части - дерево выбора таблиц, в разрезе типов данных.	
	В правой отображается выбранная таблица, используется DataTreeGrid.	
	В окне, в нижней правой части, есть кнопки "Выбрать" и "Заккрыть".	
Таблица агрегации произвольного типа - DataTreeGridChildNodesOfParentType	Все свойства DataTreeGrid.	
	Переопределено поведение кнопок верхнего Toolbar'a, из-за того что данные не одного типа, а не определённого	
	Переопределено поведение при DnD, в условиях разного типа.	
Таблица Журнала - LogTreeGrid	Все свойства DataTreeGrid	

	Все данные для таблицы берутся из поискового запроса.	
Таблица папок - CoreObjectFolder	Состоит из 2-х панелей, в левой части состоит из объектов типа "CoreObjectFolder".	"Все свойства DataTreeGrid. В верхний toolbar добавлена кнопка скрытия папки."
	Правая часть содержит данные, пред определенного для папки типа.	Все свойства DataTreeGrid.
Таблица выбора пользователей - CoreDepartmentUserView	"Состоит из 2-х панелей, в левой части состоит из объектов типа ""CoreObjectFolder"", а также ""CoreDepartment"". Отображение содержит все свойства для CoreObjectFolder".	"Содержит колонки объекта CoreObjectFolder, записи CoreDepartment добавлены с id = -1*id, для отделения того что выбрали папку или департамент".
	В правой отображается таблица по выбранному департаменту или папки, используется DataTreeGrid.	
	В окне, в нижней правой части, есть кнопки "Выбрать" и "Заккрыть".	В зависимости от того что было выбрано, будет добавлена либо папка, либо департамент, либо пользователь.

Многоуровневая таблица - CoreMasterDetailView	"В настройках в виде JSON хранится настройка пути объектов. Каждый объект отображается в виде таблицы (CoreGridView). Отображение может быть нескольких уровней, но отображается не более 2-х."	
	При выборе записи в левой таблице - правая таблица отображается в зависимости от настройки связи.	
	При выборе записи в правой таблице, если есть еще пути, Правая таблица переносится влево, а справа открываются записи.	
	В нижней части грида появляются кнопки "лева" и "права", для отображения скрытых таблиц из-за переходов.	
Таблица с формой - CoreGridWithFormView	Все свойства DataTreeGrid	
	В зависимости от настройки, открывается запись на чтение, в виде формы.	

Таблица меню - MenuTreeGrid	Все свойства DataTreeGrid	
	В восточной части можно открыть грид агрегации выбранной записи. Настраивается не в представлении, а в настройках меню.	
Многоуровневая дерево и таблица - CoreMasterTreeDetailView	"В настройках в виде JSON хранится настройка пути объектов и объект (DataTreeGrid), который необходимо отобразить при клике по дереву."	
Таблица результат - ETLProcess'a, DataTreeGridETLProcess	Все свойства DataTreeGrid, кроме некоторых кнопок.	
Грид редактирования ролей	Состоит из checkbox настройки ролей.	
Грид отображения ссылки - LinkValueCollectionGrid	Все свойства DataTreeGrid	Дополнительная кнопка "Выбрать".
Грид обратной ссылки - LinkReverseGrid	Все свойства DataTreeGrid	Дополнительная кнопка "Выбрать", удалить не удаляет, а убирает ссылку.

ГЛОССАРИЙ

А

Авторизация - предоставление определённому лицу или группе лиц прав на выполнение определённых действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий.

Администратор программных средств (ПС) управляет использованием ПС ординарными пользователями и осуществляет сопровождение ПС, не связанное с модификацией программ. Например, он может регулировать права доступа к ПС между ординарными пользователями, поддерживать связь с поставщиками ПС или выполнять определенные действия, чтобы поддерживать ПС в рабочем состоянии, если оно включено как часть в другую систему.

Архитектура информационной системы -

1. концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы. Описание (модель) основного расположения и взаимосвязей частей системы (физического либо концептуального объекта или сущности).
2. конструкция или структура, которая отображает взаимосвязи между всеми элементами внутреннего подчинения, системами или действиями;
3. совокупность артефактов конструкции или описательных представлений.

Атрибут - это свойство объекта, характеризующее его экземпляр, т.е. атрибут - порция информации, определяющая характерное свойство некоторого объекта.

Атрибут сущности - описывает данные о сущности, которые нужно сохранить. У каждой сущности ноль или более атрибутов, описывающих ее, и каждый атрибут описывает в точности одну сущность. Каждый экземпляр сущности (строка таблицы) имеет в точности одно значение, возможно, равное NULL, для каждого из своих атрибутов. Значение атрибута может быть числом, строкой символов, датой, временем или другим базовым значением данных.

Аутентификация — это процедура проверки подлинности данных, например, проверки соответствия введённого пользователем пароля к учётной записи паролю в базе данных, или проверка цифровой подписи письма по ключу шифрования, или проверка контрольной суммы файла на соответствие заявленной автором этого файла.

Г

Главное меню (см. меню) - меню, набор позиций, используемый пользователем чаще всего. Обычно главное меню занимает в интерфейсе одно из наиболее удобных мест. В

большинстве интерфейсов состав главного меню можно менять при помощи настроек приложения. Кроме того главное меню может "всплывать" при нажатии на определённую кнопку либо сразу отображаться при запуске приложения. Для некоторых команд меню могут использоваться специальные сочетания клавиш.

Группа - это именованная совокупность пользователей. Объединение субъектов в группы облегчает администрирование и, как правило, строится на основе формальной или фактической структуры организации. Каждый пользователь может входить в несколько групп. Таким образом "группы" - это логические конструкции, с помощью которых можно объединить учётные записи пользователей для какой-то общей цели.

Д

Данные - идеализированное представление окружающей действительности в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе.

Диаграмма (Diagram) - это графическое представление множества элементов. Чаще всего она изображается в виде связного графа с вершинами (сущностями) и ребрами (отношениями) и представляет собой некоторую проекцию системы.

Диаграмма классов (Class diagram) - статическая структурная диаграмма, описывающая структуру системы, демонстрирующая классы системы, их атрибуты, методы и зависимости между классами.

Диаграмма классов в терминологии UML - диаграмма, на которой показан набор классов (и некоторых других сущностей), не имеющих явного отношения к проектированию БД, а также связей между этими классами. Кроме того, диаграмма классов может включать комментарии и ограничения.

Доступ (Access) возможность получения каких-либо ресурсов системы в зависимости от прав пользователя.

И

Интерфейс (Interface) - средства для обеспечения физического или функционального взаимодействия между различными учётными элементами

Интерфейс пользователя (см. Пользовательский интерфейс)

Информационный объект (ИО) - является информационным описанием некоторой сущности (см. **Сущность**). Информация об ИО представляется совокупностью экземпляров записей данных.

К

Категории пользователей -

- Ординарный пользователь ПС (end-user) - использует ПС для решения своих задач (в своей предметной области). Он может не знать многих деталей работы компьютера или принципов программирования.
- Администратор ПС (system administrator) - управляет использованием ПС ординарными пользователями и осуществляет сопровождение ПС, не связанное с модификацией программ.

Класс - именованное описание совокупности объектов с общими атрибутами, операциями, связями и семантикой.

Корневой каталог - каталог, прямо или косвенно включающий в себя все прочие каталоги и файлы файловой системы. В корневом каталоге (root) содержится вся иерархия системы.

М

Меню - элемент интерфейса пользователя, позволяющий выбрать одну из нескольких перечисленных опций программы.

Мощность отношения (мультипликатор) - означает число связей между каждым экземпляром класса (объектом) в начале линии с экземпляром класса в её конце.

Различают следующие типичные случаи:

- ноль или один экземпляр,
- один экземпляр,
- ноль или более экземпляров,
- один или более экземпляров.

Мультипликатор (см. мощность отношения)

Н

Нотация - система условных обозначений, принятая в какой-либо области.

О

Объект - это предмет или явление, имеющее четко определенное поведение и обладающее состоянием, поведением и индивидуальностью.

П

Панель инструментов – это область графического интерфейса, содержащая элементы управления, выполненные чаще всего в виде кнопок, щелчком на которые можно вызвать необходимую команду или меню.

Пароль – это чувствительная к регистру строка, содержащая определенное количество символов. Допустимыми символами для паролей являются буквы, цифры и специальные символы.

Платформа - совокупность подходов и программных средств, предназначенных для построения информационной системы

Поле — некая характеристика с уникальным именем и набором атрибутов, таких как, например: id, название или тип поля. Поля предназначены для хранения информации об объекте. Значения полей можно отобразить, отфильтровать или изменить в зависимости от стоящих задач.

Пользовательская документация программных средств (User Documentation) - объясняет пользователям, как они должны действовать, чтобы применить разрабатываемое ПС. К таковой документации относятся документы, которыми должен руководствоваться пользователь при инсталляции ПС (при установке ПС с соответствующей настройкой на среду применения ПС), при применении ПС для решения своих задач и при управлении ПС (например, когда разрабатываемое ПС будет взаимодействовать с другими системами). Эти документы частично затрагивают вопросы сопровождения ПС, но не касаются вопросов, связанных с модификацией программ.

Пользователь (User) - лицо, потребляющее ИТ-услугу.

Пользовательский интерфейс – это совокупность информационной модели проблемной области, средств и способов взаимодействия пользователя с информационной моделью, как и компонентов, обеспечивающих формирование

информационной модели в процессе работы программной системы. Под информационной моделью понимается условное представление проблемной области, формируемое с помощью компьютерных (визуальных и звуковых) объектов, отражающих состав и взаимодействие реальных компонентов проблемной области.

Процедура - установленный способ осуществления деятельности или процесса.

Процесс - связанная последовательность действий, активностей, изменений, осуществляемая для достижения поставленных целей или удовлетворения потребностей.

Р

Рабочая область (Client Area) - прямоугольная часть окна, где прикладная программа показывает на экране выводимую информацию, такую как текст или графика.

Регламент - правила, описывающие выполнение выделенного процесса деятельности.

Роль - совокупностью прав доступа на объекты компьютерной системы. Задание ролей позволяет определить более четкие и понятные для пользователей компьютерной системы правила разграничения доступа.

Ролевое разграничение - основной идеей управления доступом на основе ролей является идея о связывании разрешений доступа с ролями, назначаемыми каждому пользователю.

С

Сервер (Server) - программный продукт или аппаратное средство (компьютер), предоставляющий какие-либо услуги (например, выдачу данных по запросу) другому компьютеру или пользователю. Обычно на сервере хранятся данные, и выполняется их обработка. На сервере приложений не обязательно должны храниться данные, а их обработка не обязательно выполняется на файл-сервере.

Сервер базы данных - содержит базу данных, сетевую операционную систему, сетевую систему управления базами данных (СУБД) для обеспечения многопользовательских запросов.

СУБД (Система Управления Базами Данных) - система управления базами данных. Программное обеспечение, которое обеспечивает ввод, хранение и обработку больших объемов информации (баз данных).

Сущность - это реальный объект, процесс, явление или событие, информация о котором должна сохраняться и быть доступна. Сущность - понятие семантическое.

Т

Тип данных - набор характеристик (полей), а именно: значений, триггеров, операций и экземпляров с определенными свойствами, которые определяют назначение объекта. Тип данных описывает поведение и представление записей с полями.

Тип поля – это характеристика, указывающая на способ представления и хранения данных в памяти, способ отображения на экране и при печати, особенности ввода и редактирования.

Трехуровневая архитектура (Three-Tier) - архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиента, сервера приложений (к которому подключено клиентское приложение) и сервера баз данных (с которым работает сервер приложений).

Триггер (Trigger) - хранимая на сервере процедура (код, последовательность действий, операций), которая запускается автоматически при наступлении определенных событий (действий).

Триггеры делятся на два типа в зависимости от времени срабатывания:

- До выполнения действия, вызвавшего срабатывание триггера (с префиксом before).
- После завершения выполнения действия, вызвавшего срабатывание триггера (с префиксом after).

У

Управление доступом на основе ролей (Role Based Access Control) - развитие политики избирательного управления доступом, при этом права доступа субъектов системы на объекты группируются с учётом специфики их применения, образуя роли.

Учётная запись пользователя – это запись, которая содержит сведения, необходимые для идентификации пользователя при подключении к системе, как и информацию для авторизации и учёта. Учётная запись может содержать анкетные данные пользователя.

Ф

Флажок (Check box) - элемент графического пользовательского интерфейса Motiware Melody One, позволяющий пользователю управлять параметром в зависимости от настройки, с двумя или тремя состояниями. Два состояния - разрешено, - запрещено. Три состояния: - разрешено, - запрещено, - не определено.

Ю

Юзабилити — показатель количества операторских ошибок, скорости взаимодействия с продуктом, скорости обучения навыкам взаимодействия и субъективной удовлетворенности определенных пользователей продукта, достигающих определенных целей/мотивов в определенной среде.

Е

ETL (Extract, Transform, Load) - один из основных процессов в управлении хранилищами данных, который включает в себя:

- извлечение данных из внешних источников;
- трансформация и очистка для обеспечения соответствия нуждам бизнес-модели;
- загрузка их в хранилище данных.

С точки зрения процесса ETL, архитектуру хранилища данных можно представить в виде трёх компонентов:

- источник данных: содержит структурированные данные в виде таблиц, совокупности таблиц или просто файла (данные в котором разделены символами-разделителями);
- промежуточная область: содержит вспомогательные таблицы, создаваемые временно и исключительно для организации процесса выгрузки.
- получатель данных: хранилище данных или база данных, в которую должны быть помещены извлечённые данные.

ETL - процесс следует рассматривать не только как процесс переноса данных из одного приложения в другое, но и как инструмент подготовки данных к анализу.

D

Drag&Drop - функция перемещения элементов при помощи мыши. Позволяет быстро и просто отредактировать порядок расположения объектов.

U

UML (Unified Modeling Language) - унифицированный язык моделирования, язык графического описания для объектного моделирования в области разработки ПО. UML является языком широкого профиля, это - открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.