

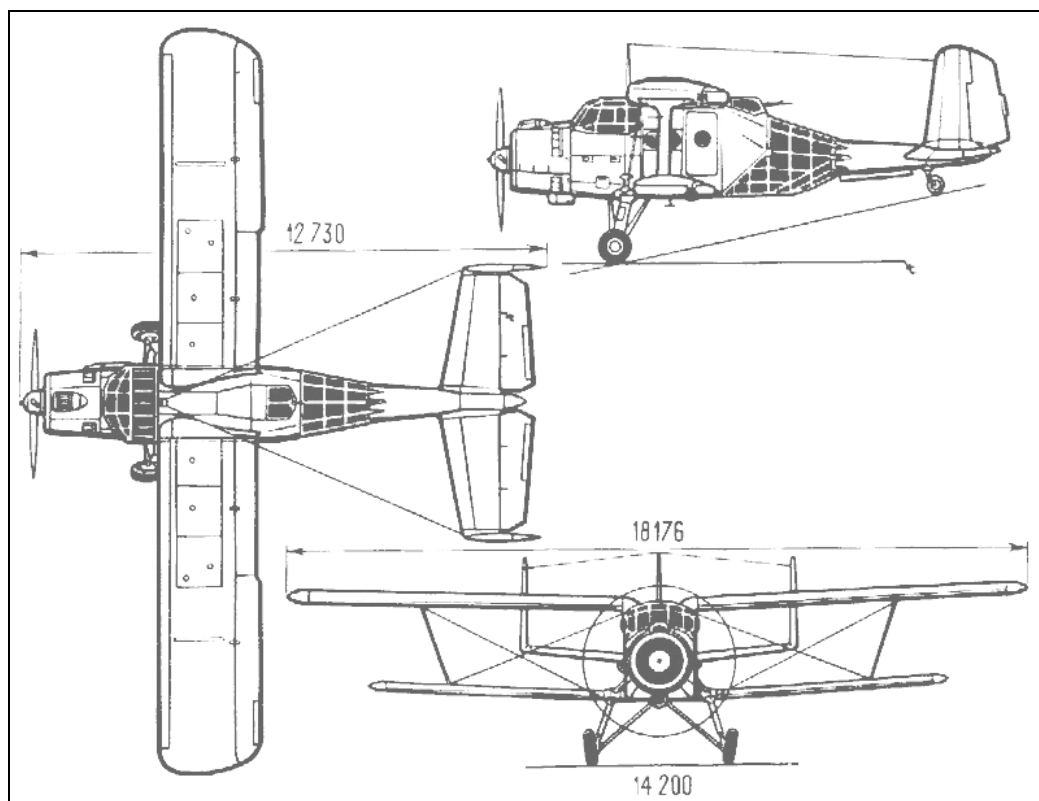


DEVELOPER'S GUIDE

MINWEB 2

Конструктор веб порталов

Руководство разработчика



MINIWEB 2 Конструктор веб порталов

Книга представляет собой справочное руководство, для разработчиков web-проектов построенные на базе технологий MiniWeb2 компании Silentium.

Представляемая система постоянно совершенствуется, поэтому новые возможности описываются дополнительными приложениями к документации и специальными инструкциями.

Copyright Silentium Company. Все права защищены. Никакая часть настоящей книги не может быть воспроизведена или в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, а так же через другие носители и информационные системы, если на то нет письменного разрешения собственника программного продукта.

Программисты-разработчики

Симонов Денис Павлович
Микитюк Сергей Владимирович
Строков Максим Владимирович

Документация

Эйрих Маргарита Владимировна

Компания-разработчик

Компания "Силентиум"

Тел. (4912) 25-17-50

390011 г. Рязань Куйбышевское шоссе, 21 офис 11

Сайт: www.silentium.ru Электронная почта: mail@silentium.ru

Рязань 2004-2014 г.

Содержание

1. ОБЩЕЕ ОПИСАНИЕ.....	7
1.2. Состав пакета.....	8
1.3. Инструкции по установке.....	9
2. СЕРВЕРНАЯ ЧАСТЬ.....	10
3. КЛИЕНТСКАЯ ЧАСТЬ.....	11
4. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС.....	12
4.1. Проекты.....	12
4.2. Метаданные.....	14
4.2.1 Подключения.....	14
4.2.2 Переменные.....	16
4.2.3 Действия.....	17
4.2.4 Источники.....	17
4.2.5 SQL Скрипты.....	25
4.2.6 Диалоги.....	26
4.2.7 Отчёты.....	28
4.2.8 Шаблоны отчётов.....	31
4.2.9 Fly шаблоны.....	33
4.3 Доступ.....	34
4.3.1 Настройка.....	35
4.3.2 Пользователи.....	35
4.3.4 Типы сущностей.....	37
4.3.6 Сущности.....	38
4.3.7 Разрешения.....	39
7. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ.....	40
8. СПЕЦИФИКАЦИИ.....	41
8.1 Переключатели полей метаисточника.....	41
8.2 Переключатели полей диалогов.....	42
9. ЭПИЛОГ.....	43

1. Общее описание.

Программное обеспечение MiniWeb2 состоит из набора функций и объектов, позволяющих эффективно разрабатывать прикладные системы, работающие в среде web, где помимо обширного набора готовых блоков существует пользовательский интерфейс для настройки работы приложения.

По сути это фреймворк со всеми необходимыми элементами. Отличительной особенностью предлагаемого инструмента является то, что он рассчитан на обработку больших объемов информации, размещающихся в базе данных с целью их эффективного использования, а так же дальнейшего анализа.



MiniWeb2 не накладывает ограничений на структуру базы данных, за исключением готовых таблиц для сохранения информации о пользователях и их правах.

Использовать MiniWeb2 можно как целиком, так и частично. Второй вариант требует глубокого знания внутренних механизмов. В простых случаях для начала работы достаточно изучить поставляемые демонстрационные примеры. Представляемое программное обеспечение постоянно развивается, поэтому документация может несколько отставать от последних версий платформы.

Помимо стандартных возможностей, реализованных в представляемой системе, так же предлагается использование методики построения приложения с помощью обычных web-технологий. Так как на некоторых этапах требуется написание дополнительного кода, то для разработки рекомендуются следующие языки

- ✓ PHP
- ✓ JavaScript

Возможно, что совместно могут использоваться и другие языки и технологии. Однако, в таком случае ошибки, возникающие при их использовании, не вносятся в список багов, и, как следствие работа по их устранению вестись в дальнейшем не планируется.

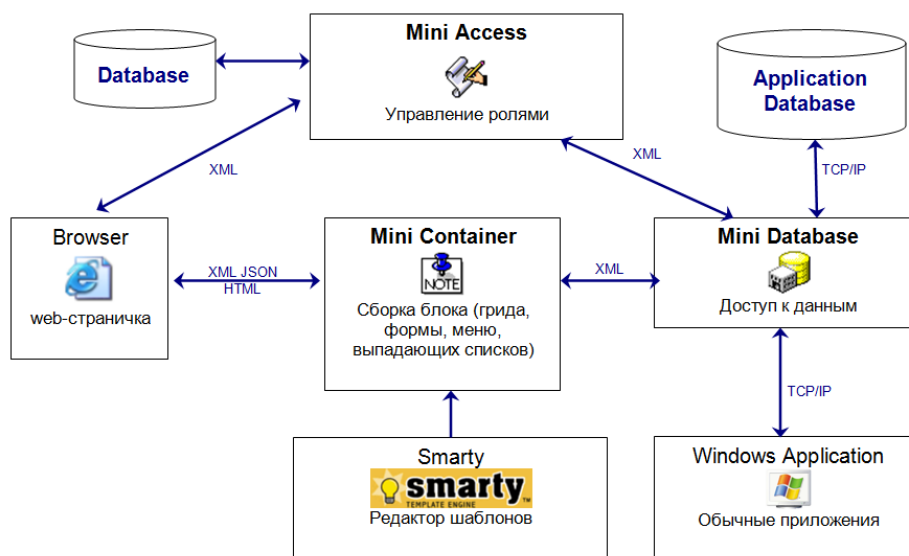


Рис. 1.1 Схема взаимодействия блоков и подсистем на уровне логики.

1.2. Состав пакета.

Пакет MiniWeb2 состоит из следующих основных блоков:

- ✓ Серверная часть
- ✓ Клиентская часть
- ✓ Пользовательский интерфейс
- ✓ Дополнительные средства

Серверная часть представляет собой web-службу предоставляющую по запросу непосредственно информацию из базы данных предметной области, а так же о настройках и состоянии системы относительно проекта, с которым работает web-страница.

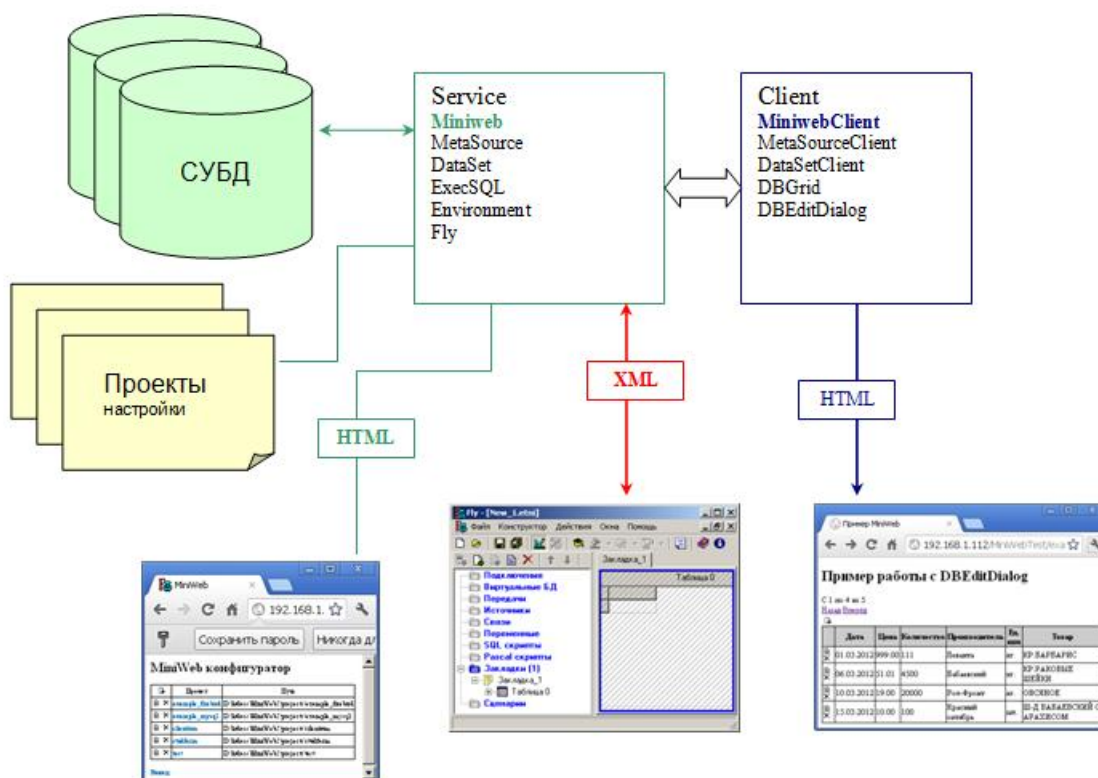


Рис. 1.2 Общая архитектура компонентов MiniWeb2

Система может быть разнесена на три сервера. Можно на два сервера, можно сделать все на одном. Здесь не указаны библиотеки Smarty, но они могут подключаться, так же как равно и аналогичные шаблонизаторы.



Вы можете подключать в проекте любые требуемые вам готовые библиотеки сторонних производителей.

1.3. Инструкции по установке.

Требования к Apache

Для работы переадресации в файле конфигурации httpd.conf раскомментировать строку

```
LoadModule rewrite_module modules/mod_rewrite.so
```

Требования к PHP

Версия 5.3.3 или выше

Для работоспособности необходимо включить следующие расширения (для OS Linux расширения файлов другие)

- ✓ php_mbstring.dll (работа с юникодом)
- ✓ php_curl.dll (библиотека CURL)
- ✓ php_interbase.dll (драйвер для работы с Interbase/Firebird)
- ✓ php_mysqli.dll (драйвер для работы с MySQL)
- ✓ extension=php_pdo_firebird.dll (драйвер PDO для работы с Interbase/Firebird)
- ✓ extension=php_pdo_mysql.dll (драйвер PDO для работы с MySQL)

Установка

- ✓ Скачать фреймворк MiniWeb <http://server/miniweb2/service/help/downloadMiniWeb> (Помощь/Загрузки)
- ✓ Распаковать в нужную папку
- ✓ Для работы встроенных примеров необходимо создать базы данных

Создание баз для примеров Firebird

- ✓ Создать базу данных MW_ROLES с кодировкой WIN1251
- ✓ Настроить алиасы в файле firebird/aliases.conf
- ✓ Создать базу данных STORE (аналогично предыдущей)
- ✓ Перейти в проект firebird_examples
- ✓ Перейти к настройкам подключения Метаданные / Подключения
- ✓ Поправить настройки подключения в соответствии с расположением сервера
- ✓ Убедиться, что подключения проходят
- ✓ Перейти к скриптам Метаданные / Скрипты
- ✓ Выполнить скрипты \$create_database, \$mw_roles

Создание баз для примеров MySQL

- ✓ Создать базу данных MW_ROLES с кодировкой cp1251
- ✓ Создать базу данных STORE с кодировкой cp1251
- ✓ Перейти в проект mysql_examples
- ✓ Перейти к настройкам подключения Метаданные / Подключения
- ✓ Поправить настройки подключения в соответствии с расположением сервера
- ✓ Убедиться, что подключения проходят
- ✓ Перейти к скриптам Метаданные / Скрипты
- ✓ Выполнить скрипты \$create_database, \$mw_roles

2. Серверная часть.

Активная часть MiniWeb2 представлена в виде двух независимых наборов функций (классов), которые могут работать на разных серверах.

- ✓ Серверная часть
- ✓ Клиентская часть

Miniweb Service может взаимодействовать напрямую с Windows-приложениями. Можно устанавливать MiniWeb2 частично. Однако это требует понимания внутренних процессов, и мы рекомендуем ставить весь пакет целиком вне зависимости, какую часть планируется использовать.

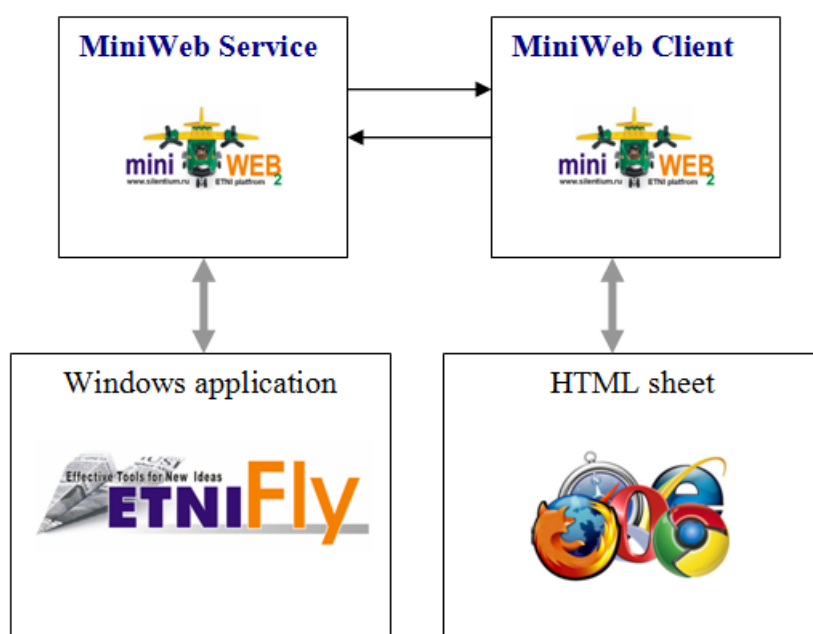


Рис. 2.2 Состав MiniWeb2.

MiniWeb Service реализует следующие основные функции:

- ✓ Визуализация интерфейса настройки проектов и системы в целом
- ✓ Предоставление данных из источников (метаисточников) в формате XML и JSON
- ✓ Модификация данных
- ✓ Запуск скриптов к базе данных
- ✓ Обеспечение авторизации пользователей
- ✓ Проверка прав доступа к объектам системы (источники, скрипты, управление, производные элементы)
- ✓ Хранение переменных сессии
- ✓ Генерация ETNI Fly файлов на основе шаблонов
- ✓ Получение и сохранение данных из ETNI Fly
- ✓ Генерация отчетов из настроенных шаблонов

Сервисная часть расположена в папке "service" и содержит каталоги:

✓	Controllers	Контроллеры для связи и визуальной настройки
✓	Images	Графика для оформления интереса пользователя по настройке
✓	Log	Файлы информации о возникающих ошибках
✓	Lib	Базовые библиотеки на PHP
✓	Plugins	Подключаемые библиотеки (авторизации и другие)
✓	Scripts	Базовые библиотеки JavaScript
✓	Styles	Библиотека стилей
✓	Templates	Шаблоны Smarty для визуализации интерфейса настройки
✓	Templates_c	Временная папка шаблонизатора Smarty

3. Клиентская часть.

Клиентская часть содержит в себе библиотеки на PHP и на JavaScript и позволяет не только использовать встроенные функции, но предоставляет возможность написания собственного слоя, базирующегося на контроллерах, написанных специально для создаваемого проекта. Кроме контроллеров предусмотрена работа с собственными шаблонами, стилями, вспомогательными скриптами, библиотеками сторонних фирм.

MiniWeb Client реализует следующие основные функции:

- ✓ Трансляция HTML-запросов в JSON их отправка в MiniWeb Service и обратно
- ✓ Предоставление набора библиотек для написания собственных контроллеров
- ✓ Предоставление базовых контроллеров обработки информации
- ✓ Автоматическое формирование окон редактирования данных
- ✓ Кэширование прав доступа для увеличения скорости проверки прав
- ✓ Связь с шаблонизатором Smarty для модулей проекта
- ✓ Генерация HTML-кода для активных элементов страницы

Клиентская часть MiniWeb2 расположена в папке "client" и содержит каталоги:

✓	Controllers	Базовые контроллеры для наследования в проектах
✓	Images	Графика для демонстрационных примера
✓	Log	Файлы информации о возникающих ошибках
✓	Lib	Библиотеки взаимодействия с сервером на PHP
✓	Plugins	Подключаемые библиотеки (грид на основе Smarty)
✓	Scripts	Библиотеки JavaScript для проектов
✓	Styles	Стили (диалог редактирования и другие)
✓	Views	HTML шаблоны (без использования Smarty)

Для сохранения информации о проекте используется папка project, внутри которой располагается папка с названием проекта, которая содержит директории:

✓	Baseexport	Шаблоны и настройка отчетов
✓	Fly_templates	Шаблоны ETNI Fly
✓	Scripts	SQL-скрипты данных
✓	Sources	Метаисточники проекта

А так же в папке проекта расположены файлы

✓	connections.ini	Настройка подключений к базе данных
✓	environment.ini	Переменные среды проекта
✓	fly.ini	Конфигурация заливки данных из файлов ETNI Fly
✓	security.ini	Настройка плагина ролей

4. Пользовательский интерфейс.

Разделы главного меню имеют следующий перечень пунктов, который меняется в зависимости от состояния (выбора проекта, входа в систему):

- ✓ Проекты
- ✓ Метаданные
- ✓ Доступ
- ✓ Помощь
- ✓ Выход

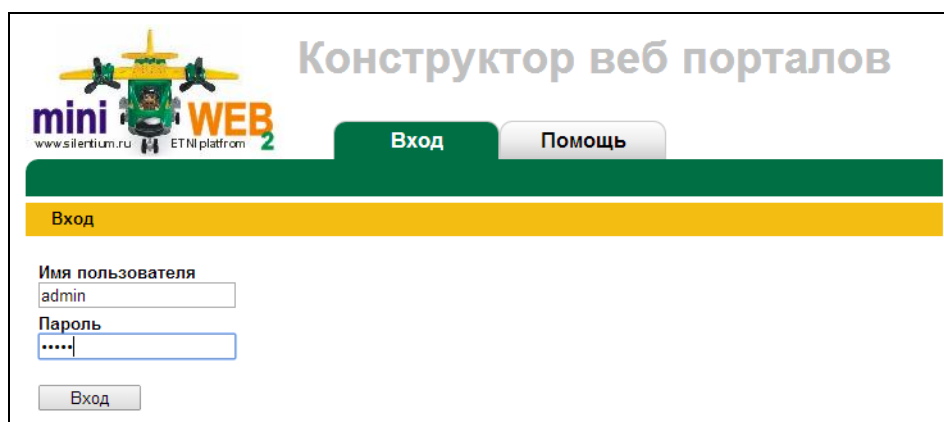


Рис. 4.1 Вход в систему настройки MiniWeb2.

4.1. Проекты.

В разделе “Проекты”, помимо выбора текущего проекта можно выполнить дополнительные действия

- ✓ Проекты
- ✓ Настройки
- ✓ Изменить пароль
- ✓ Тесты
- ✓ Очистить логи
- ✓ Очистить кэш Smarty

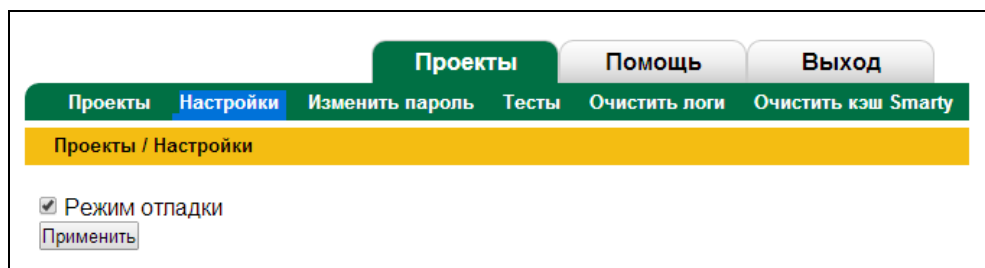


Рис. 4.1.0 Настройка MiniWeb.

В настоящее время для настройки вынесен только один параметр, но далее набор настроек будет постепенно увеличиваться.

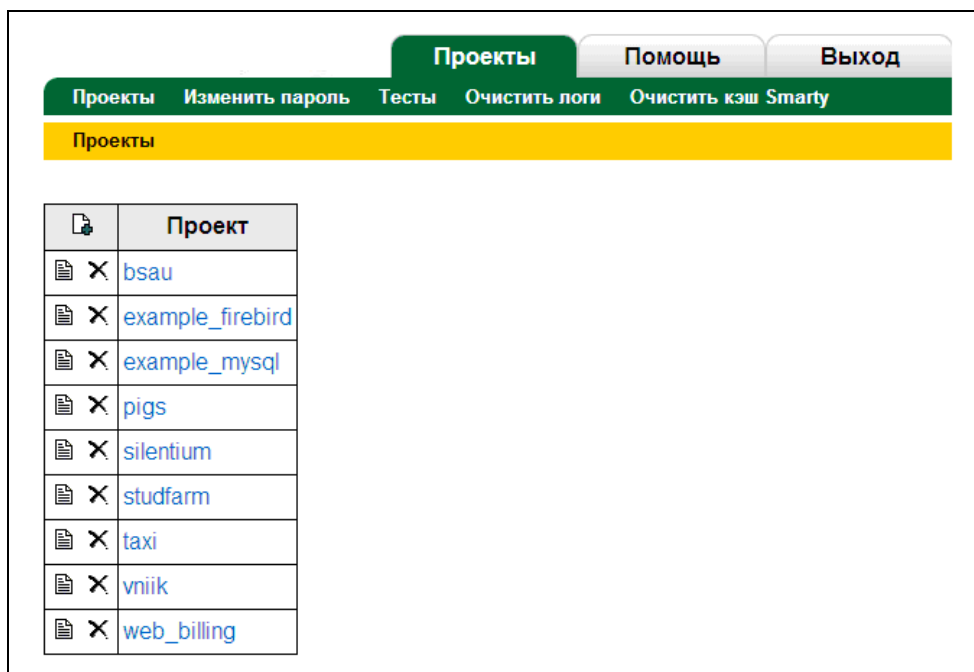


Рис. 4.1.1 Пример выбора текущего проекта из существующих.

Для выбора проекта достаточно просто щелкнуть на его названии. Так же можно отредактировать его название, удалить его с помощью кнопок расположенных слева в списке проектов. Добавление происходит с помощью кнопки в левом верхнем углу таблицы.

Смена пароля – это стандартная процедура. Достаточно дважды ввести новый пароль в появившемся диалоге. После установки новой версии MimiWeb2 или смены версии PHP, других библиотек настоятельно рекомендуется провести тесты, которые дают результаты в следующих категориях:

- ✓ Успешные тесты
- ✓ Пропущенные тесты
- ✓ Неудачные тесты
- ✓ Ошибки

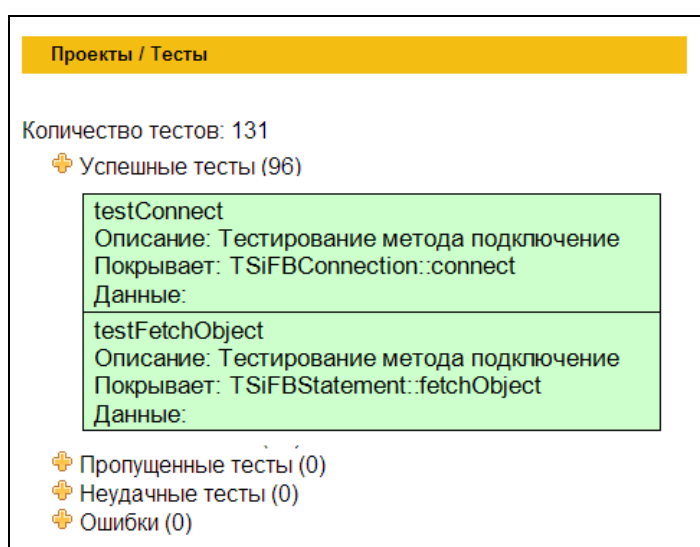


Рис. 4.1.2 Иллюстрация прохождения тестов.

В процессе возникновения сообщений об ошибке все они попадают в специальный лог-файл, который можно очистить, выбрав указанный выше пункт меню. Аналогично работает очистка кэша Smarty, отвечающего за подстановку шаблонов. Особенно рекомендуется делать такую очистку при переносе проекта на другой хостинг провайдера.



При переносе системы (проектов) на другой хостинг настоятельно рекомендуется проверить выполнение тестов, а так же провести очистку кэша Smarty.

Для того чтобы двигаться дальше обязательно нужно выбрать необходимый проект (либо создать, если такого еще не существует).

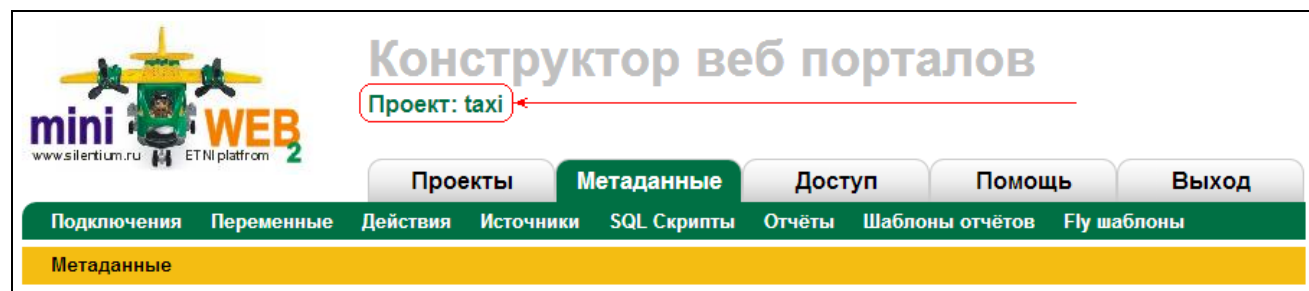


Рис. 4.1.3 Выбранный текущий проект.

Выбранный проект можно увидеть в шапке над главным меню. Так же, обратите внимание после выбора проекта перечень доступных меню становится гораздо разнообразнее.

4.2. Метаданные.

Метаданные – это то, что присуще именно выбранному проекту. Это – специфика, которая отделена от исходного кода программ. Особенно это касается конструкций на языке SQL-запросов с тем, чтобы далее они не смешивались с PHP, с JavaScript, с тегами HTML.

4.2.1 Подключения.

Метаданные / Подключения								
	Алиас	Тип	Сервер	Порт	База данных	Пользователь	Кодировка	Драйвер PDO
	taxi	MySQL	127.0.0.1	3306	taxi	root	cp1251	Да

Рис. 4.2.1 Подключения к базам данных.

Подключений к базам данных может быть несколько. Они могут располагаться на разных серверах, иметь различные настройки. При добавлении нового подключения указываются следующие поля:

- ✓ Алиас
- ✓ Тип
- ✓ Сервер
- ✓ Порт
- ✓ База данных
- ✓ Пользователь
- ✓ Пароль
- ✓ Кодировка

На рисунке ниже представлено окно настройки подключения к базе данных. Формируется алиас подключения к базе данных. С помощью этого алиаса далее происходит работа источников, SQL скриптов и так далее.

Тип подключения может быть выбран для следующих баз данных:

- ✓ mySQL
- ✓ Intebase
- ✓ Postgres

Сервер, на котором располагается база данных, может быть отличен от того, на котором развернута система MiniWeb2. Его необходимо указать в поле “Сервер”. Значение может быть прописано как Localhost (или 127.0.0.1) для случая, когда все располагается на том же сервере. Если указывается мнемоническое название – это может приводить к задержкам в работе системы.

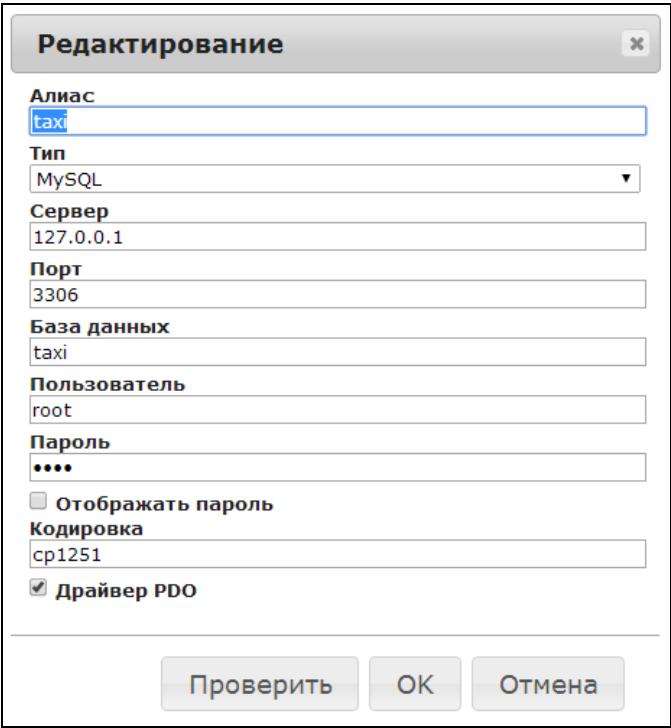







Рис. 4.2.2 Пример добавления нового подключения.

Остальные опции стандартные и не требуют пояснений. За исключением “Драйвер PDO”. Эта опция переключает подключаемые к PHP библиотеки и может быть полезна при возникновении проблем с базой данных.

Возможны следующие манипуляции с подключениями

Таблица 4.2.1 Действия с подключениями

Кнопка	Описание
	Добавить новое подключение
	Настроить подключение
	Протестировать подключение
	Вывести перечень таблиц
	Удалить подключение

4.2.2 Переменные.

Переменные требуются для построения систем высокого уровня и для пробных модулей могут вовсе не потребоваться. Однако следует знать, что они существуют и могут быть полезны для получения дополнительной информации внутри запросов источников, а так же при получении доступа к внутренним механизмам контроллеров, которые описаны в этом руководстве.

Метаданные / Переменные							
	Имя	Действие	Имя поля	Значение	Тип	По-умолчанию	Указано
 X	%PERSON_CODE_POST%				Локальная	0	Нет
 X	%DISCIPLINE_CODE_DEPARTMENT%				Локальная	0	Нет
 X	%STUDENT_CODE_PROFILE%				Локальная		Нет
 X	%ENTRYYEAR%			2012	Серверная		Да
 X	%CURRENT_CODE_USER%				Локальная		Нет
 X	%DOC_CODE_PERSON%				Локальная		Нет
 X	%DOC_CODE_POST%				Локальная		Нет
 X	%WAY_CODE_WAYTYPE%				Локальная		Нет

Рис. 4.2.3 Переменные проекта.



Переменные так же можно использовать в качестве констант приложения.



По типу переменные могут быть следующие

- ✓ Локальные
- ✓ Серверные
- ✓ Динамические

Если переменная локальная, то ее значение обрабатывается только в рамках сессии, в которой работает текущий пользователь (грубо говоря, на локальном компьютере). Значение серверной переменной прописывается глобально, и ее изменение сказывается на работе всех пользователей.

Если переменная динамическая, то ее значение каждый раз считывается заново, то есть запускается связанное с переменной действие. Действия описаны в следующей главе, их набор указан в спецификации к данному руководству. Перечень действий постоянно пополняется.

Таблица 4.2.2 Манипуляции с переменными

Кнопка	Описание
	Добавить переменную
	Редактировать переменную
X	Удалить переменную



Значение по умолчанию особенно важно для локальных переменных. Разработчику следует следить за ситуацией, когда начинается работа с переменной на локальном компьютере.

4.2.3 Действия.





Метаданные / Действия				
	Наименование	Тип	Выходные поля	Описание
 X	Текущее время	GET_CURRENT_TIME	VALUE	Определение текущего поля
 X	Текущий пользователь	GET_CURRENT_USER_NAME	VALUE	Возвращает имя авторизованного пользователя
 X	Ввести значение	INPUT_STRING	TEXT	Ввод текста от пользователя



Рис. 4.2.4 Доступные для переменных действия.

Действия, приведенные здесь в качестве примера, скорее всего, будут переименованы в дальнейшей реализации. Не следует опираться на приведенные названия.



На момент написания данной документации действия еще не реализованы. Их полный перечень смотрите в спецификациях, которые будут так же представлены на страничках интерфейса.

Таблица 4.2.3 Манипуляции над действиями

Кнопка	Описание
	Добавить действие
	Редактировать действие
X	Удалить действие

Редактирование действия X

Имя

Тип

Выходные поля

Описание

Рис. 4.2.5 Настройка действия.

4.2.4 Источники.

Источники или метаисточники, как они обозначены нашей компанией в аналогичных разработках под Windows, играют основную роль в обработке информации. Это и способ доступа к базе данных, описание связей выводимых таблиц, подписывание полей, автоматические окна редактирования записей и так далее. Множество функций реализуется с помощью переключателей, прикрепляемых к определенному полю. Такие настройки позволяют резко сократить цикл разработки и требуют детального описания.

Таблица 4.2.4 Действия с источниками

Кнопка	Описание
	Добавить источник
	Редактировать источник
	Просмотреть данные
	Удалить источник

Действия с источниками стандартные. Их описывать не будем. Важно научиться использовать кнопку “Просмотреть данные”. С ее помощью можно легко проверить работоспособность каждого источника.

Проблемы с проверкой работоспособности могут возникнуть при параметризованных запросах. Это вопрос будет решен в следующих версиях продукта (можно будет поставить параметры по-умолчанию).
































Метаданные / Источники	
	Наименование
  	SRC_COMPANY
  	SRC_DRIVER
  	SRC_HOUSE
  	SRC_HOUSE_CODE
  	SRC_PART
  	SRC_PRICELIST
  	SRC_REGION
  	SRC_SECTOR
  	SRC_STOP
  	SRC_STREET

Рис. 4.2.5 Пример настроенных источников.

Настройка источника базируется на окне, которое имеет закладки

- ✓ Главная
- ✓ Таблицы
- ✓ Запрос
- ✓ Поля
- ✓ Результаты

На главной закладке указывается имя источника (далее по этому имени происходит обмен данными из контроллеров проекта), указывается подключение путем выбора уже настроенного алиаса (описанного в главе “Подключения”). Указывается технология доступа

- ✓ Таблица
- ✓ Запрос
- ✓ Запрос-таблица

По технологии “Таблица” не требуется знания языка SQL. Достаточно указать какие таблицы используются, как они связаны. Это самый простой вариант работы источника. Для технологии “Запрос”, наоборот не требуется связывание таблиц (которое происходит визуально). Достаточно написать произвольный SQL-запрос возвращающий данные. Недостаток такого подхода в том, что

представляемые данные нельзя редактировать. Такая технология хороша для построения отчетов, сбора статистики, проведения анализа данных.

И последняя технология "Запрос-таблица" очень эффективна, но несколько трудна в настройке. При таком подходе данные вводятся в диалоговом окне с помощью связанных таблиц, а вывод данных происходит с помощью запросов.

Настройка метаисточника

Главная Таблицы Запрос Поля Результаты

Наименование
SRC_COMPANY

Подключение
taxi

Технология
☒ Таблица
☐ Запрос
☐ Запрос-таблица

OK Отмена Справка

Рис. 4.2.6 Выбор технологии для источника, указание его названия.

Настройка метаисточника

Главная Таблицы Запрос Поля Результаты

Подключение	Таблица	Алиас	Индекс	Связи
	region	REGION	CODE_REGION	
	street	STREET	CODE_STREET	
	house	HOUSE	CODE_HOUSE	CODE_STREET-STRE
	company	COMPANY	NAME	CODE_HOUSE-HOUSI

OK Отмена Справка

Рис. 4.2.7 Связывание таблиц источников.






Важно согласовывать потоки данных между таблицами и запросами. Если в таблицах не хватает каких-то полей, которые есть в запросе, то их необходимо пометить как "Calculated" в переключателях к полям.

Так же можно порекомендовать для такой технологии сначала сформировать связи таблиц, описать поля и уже, потом автоматически сгенерировать требуемый запрос (чтобы не писать его вручную на начальной стадии). Далее запрос можно легко довести до нужного состояния.

Далее опишем, как можно быстро связать несколько таблиц. Связывание происходит по методу один к одному. То есть одной записи настраиваемой таблицы сопоставляется одна запись в другой таблице. Если записей находится несколько, то берется первая попавшаяся. Разумеется, такого быть не должно и следить за этим должен сам разработчик.

Таблица 4.2.6 Действия с полями таблиц

Кнопка	Описание
	Добавить
	Редактировать
	Удалить
	Поднять на строку выше
	Опустить на строку ниже

Для начала надо добавить все участвующие в связке таблицы. Не обязательно сразу добавлять все. Рекомендуем начинать с двух, проверять работоспособность и добавлять еще. Важен порядок следования таблиц. Последней идет главная таблица, по которой и будет происходить навигация, добавление, редактирования, удаление. Все то, что выполняет обычный пользователь. Это – основа понимания работы источников. Редактируется только последняя таблица в списке. Остальные – это только для реализации вычисляемых полей, присутствующих в ней.

И так, при добавлении в список таблиц, появляется окно представленное ниже. Первое поле “Подключение” указывать не надо оно берется из первой закладки источника. Это поле может быть полезно при построении гетерогенных связей, когда таблицы находятся в различных базах данных. Таблицу соответственно выбираем из выпадающего списка, в котором перечисляются все существующие таблицы в базе данных. Таблица может быть связана не один, а несколько раз. Для этих целей используется алиас таблицы, который может проставляться по нажатию кнопки расположенной справа от поля ввода алиаса.

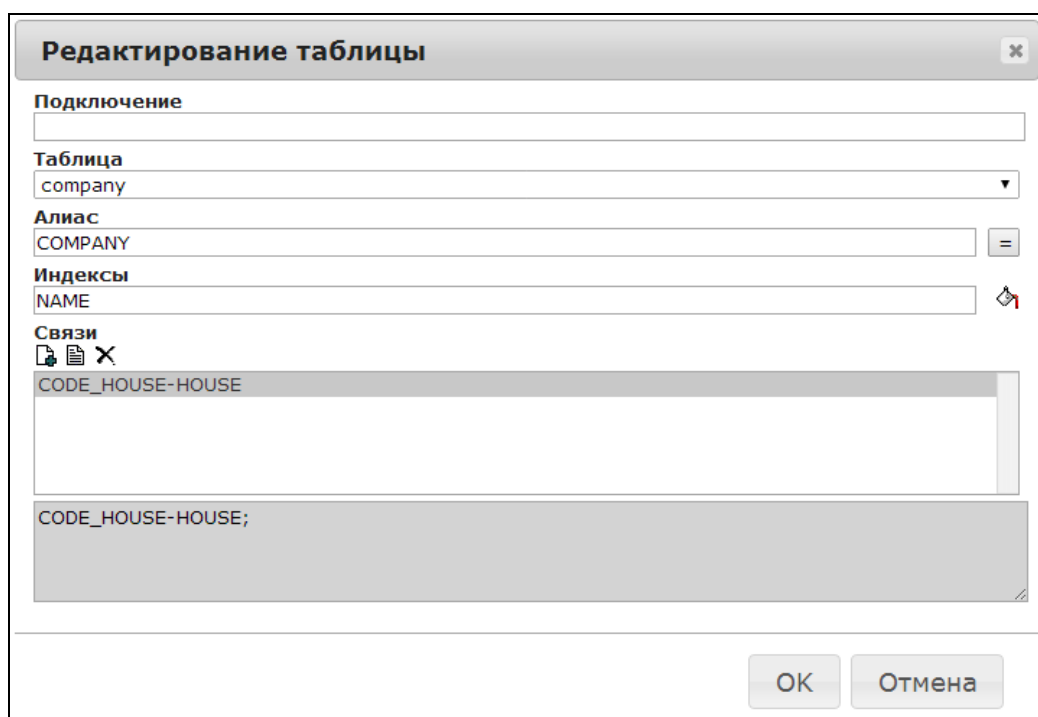




Рис. 4.2.8 Редактирование настройки строки таблицы.

В поле “Индекс” можно указать любое поле из существующих в этой таблице. Но для связанных таблиц указывается именно то, по которому указывается связь. Далее это учитывается. Индекс можно проставить путем ввода символов с клавиатуры, а можно перейти в режим выбора поля из списка по кнопке справа от поля “Заполнить перечень полей в качестве индекса”

Таблица 4.2.5 Действия с настройкой таблицы.

Кнопка	Описание
	Проставить алиас по названию таблицы
	Заполнить перечень полей в качестве индекса

Связи строятся для таблиц визуально, но их синтаксис можно контролировать в самом нижнем поле окна редактирования, обозначенного серым цветом. Связь идет от главной таблицы к подчиненной. То есть таблицы надо настраивать, начиная с последней таблицы и выше по списку. Сама связь добавляется очень просто. Нужно просто указать на какую таблицу ссылаемся, и какое поле за это отвечает. Не правда ли кажется что информации недостаточно для связывания? Надо по идее четыре пункта

- ✓ Имя главной таблицы
- ✓ Поле главной таблицы
- ✓ Имя подчиненной таблицы
- ✓ Поле подчиненной таблицы

Так вот, имя главной таблицы берется в предыдущем окне (на этой таблице находимся), а поле подчиненной таблицы берется из поля индекс. Остается ввести только соотношение поле-таблица. Именно это и представлено на рисунке ниже.

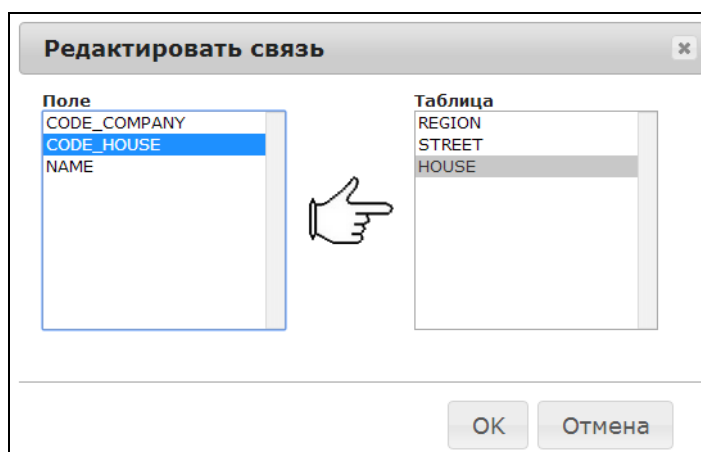


Рис. 4.2.9 Формирование связи между таблицами.

Настройка SQL-запросов сводится к внесению кода внутри поля “Текст SQL-запроса”. Тут же его можно запустить и проверить его работоспособность. Поля при этом не подписываются и не прячутся (согласно описанию на следующей закладке).

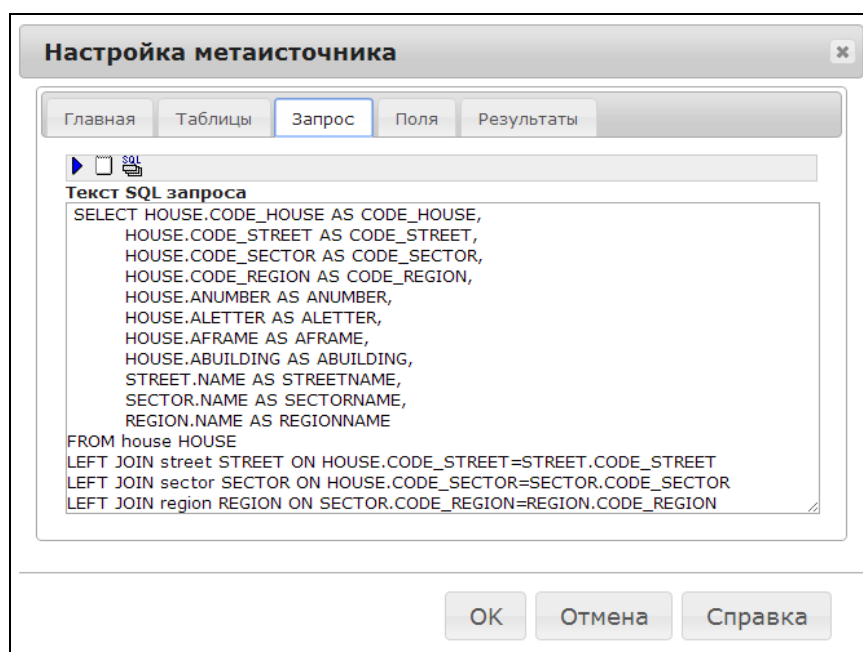





Рис. 4.2.10 Формирование запроса для источника.

Таблица 4.2.6 Настройка SQL-запроса.

Кнопка	Описание
	Пробный запуск запроса
	Просмотр генерации дополнительных SQL-инструкции
	Автоматическая генерация SQL-кода для связанных таблиц

На рисунке ниже приводится пример генерации дополнительных скриптов. Их просмотр позволяет быстро диагностировать возникающие проблемы с навигацией, поиском, модификацией записей. Можно скопировать эти запросы и попробовать запустить в какой-нибудь другой среде для локализации проблемы.

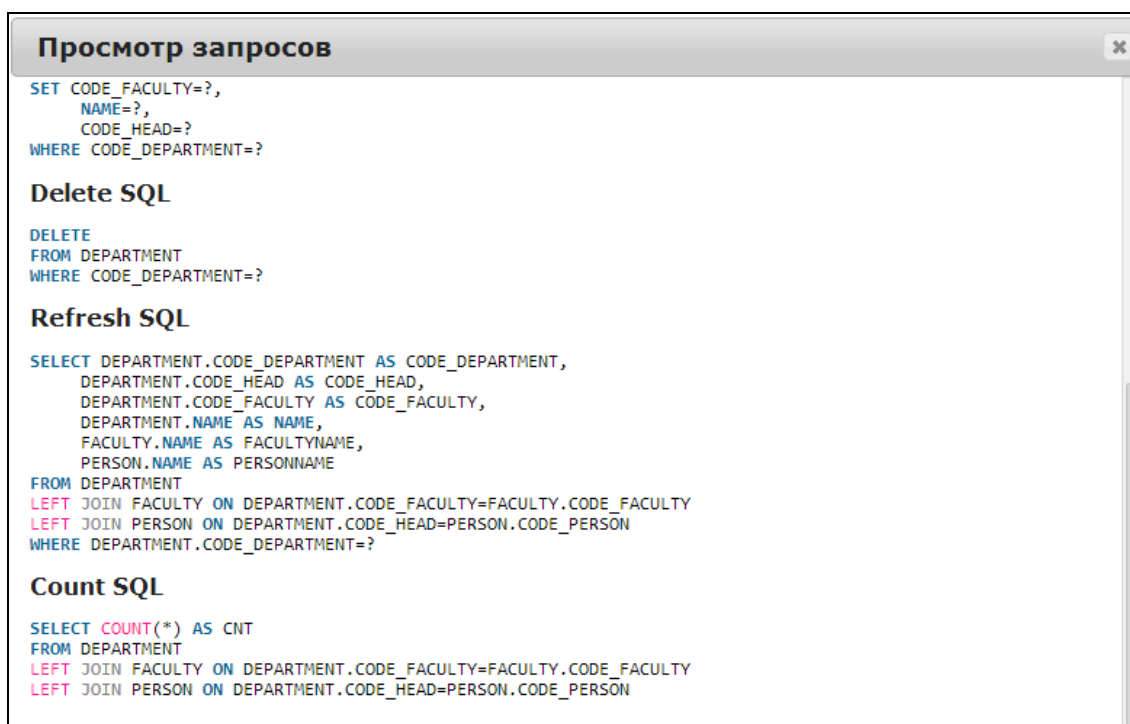


Рис. 4.2.11 Просмотр генерации дополнительных SQL-инструкций.

Автоматическая сборка SQL-запроса базируется на настройках, сделанных на предыдущей закладке по связыванию таблиц. При этом предыдущий текст SQL-запроса полностью затирается. Если что-то пошло не так всегда можно нажать кнопку отмены, расположенную в нижней части окна редактирования.



Автоматическая сборка SQL-запроса полностью удалит его предыдущий вариант, который мог быть написан вручную.

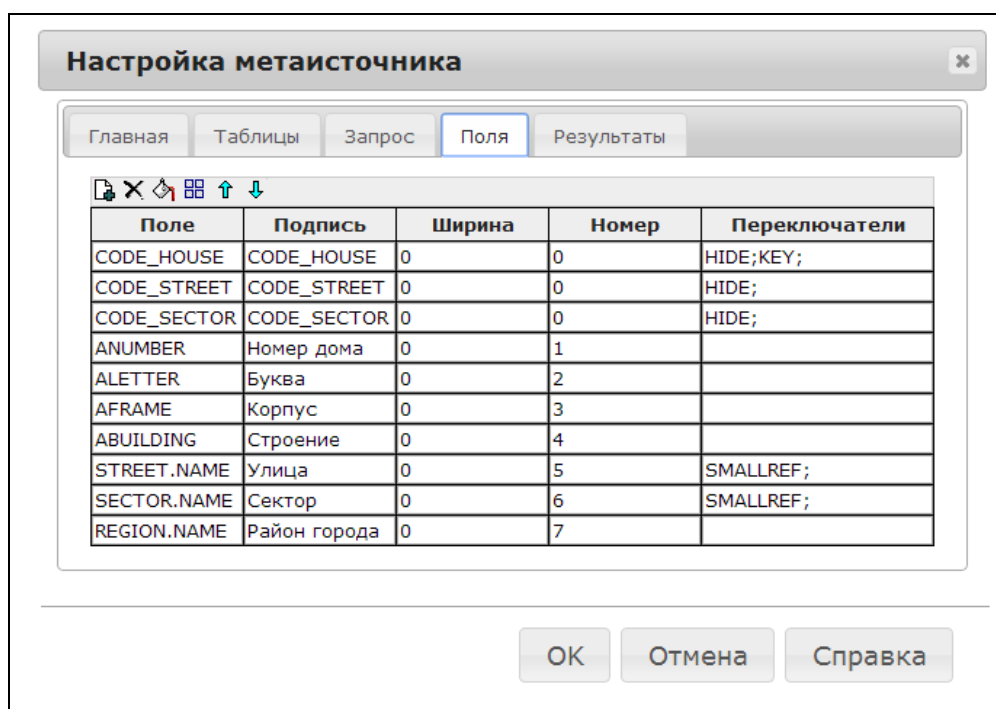


Рис. 4.2.12 Настройка полей источника.

Чтобы не добавлять каждое поле вручную их можно занести одним махом по кнопке "Заполнить поля". Сразу добавляется их полный список. Можно добавлять по одному "Добавить поле". Это требуется делать для полей, находящихся не в главной таблице, а в соседних (связанных). При этом пишется через точку: Таблица. Поле. Более правильно сказать, что пишется алиас таблицы, точка, название поля. Например: COMPANY.NAME. Далее внутри системы эта точка обрезается и поле может быть доступно как "COMPANYNAME". Поля можно быстро удалить по команде "Очистить поля". Так же можно изменить порядок их следования кнопками "Поднять" и "Опустить".

Таблица 4.2.7 Функции для формирования полей.

Кнопка	Описание
	Добавить поле
	Удалить поле
	Заполнить поля
	Очистить поля
	Поднять на строку выше
	Опустить на строку ниже

В правом столбце напротив каждого поля можно поставить переключатель, либо несколько из них. Эти переключатели диктуют поведение источника и стандартных функций, ассоциированных с ним в проекте. Чтобы быстро получить их список необходимо выбрать кнопку "Справка", расположенную в нижнем ряду окна.

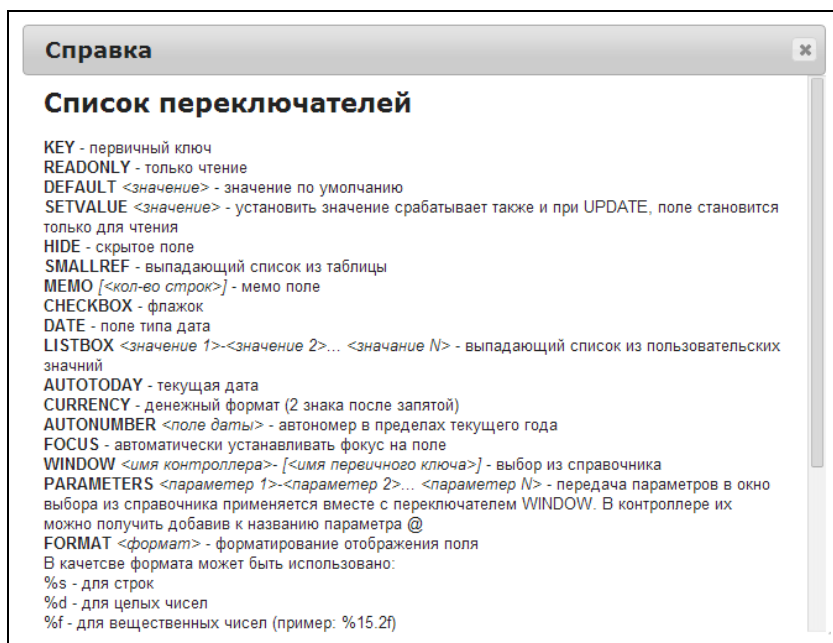


Рис. 4.2.13 Перечень переключателей для полей метаисточника.



Полный перечень переключателей необходимо смотреть в разделе "Спецификации". Их перечень периодически пополняется новыми.

Перечень колонок описания полей следующий

- ✓ Поле
- ✓ Подпись
- ✓ Ширина
- ✓ Номер
- ✓ Переключатели

Поле прописывается, так как оно есть физически в таблице. Поля связанных таблиц пишутся через точку. Подпись поля идет обычно на русском языке и отображается потом в гриде и окне редактирования, может состоять из нескольких слов. Ширина указывается в пикселях для поля редактирования. Номер дает информацию о том, в какой строке выводить. Таким образом, порядок следования полей в диалоговом окне и в гриде может быть разным.

Теперь, когда все настроено (либо сделано в первом приближении) самое время проверить, работает ли то, что разработчик настраивает. Для этого предназначена закладка "Результаты". Здесь можно видеть то, как сформируется грид на странице сайта.

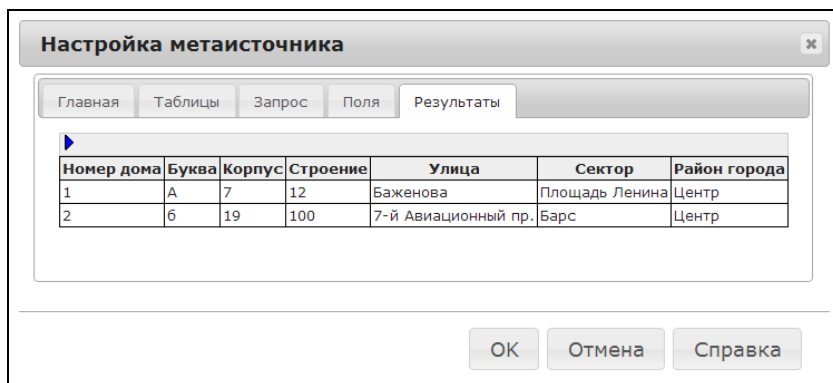



Рис. 4.2.14 Тестовый запуск источника.

Таблица 4.2.8 Функции проверки результатов.

Кнопка	Описание
	Запуск источника

4.2.5 SQL Скрипты.

Отличие скриптов от источника в том, что они обычно не возвращают данные, но делают некоторые манипуляции над базой данных. Важно передать в скрипты параметры и запустить их в контроллере внутри проекта.

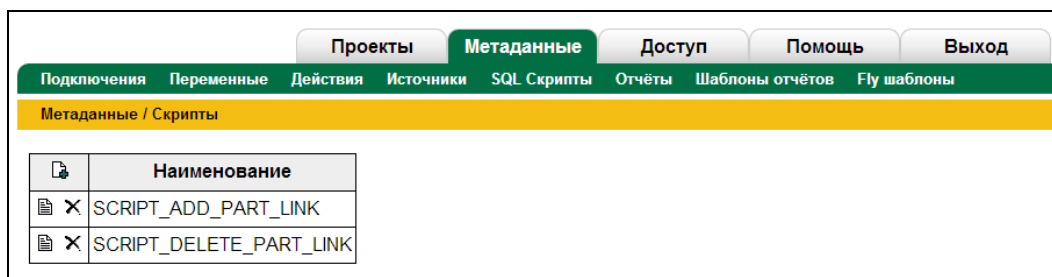




Рис. 4.2.15 Настройка скриптов проекта.

У каждого скрипта есть свое название. Скрипты можно добавлять, удалять, редактировать так как и другие объекты системы. При добавлении новое название скрипта ставится автоматически, например "SCRIPT_1", "SCRIPT_2". Далее это название можно легко изменить.

Таблица 4.2.9 Настройка SQL скриптов проекта.

Кнопка	Описание
	Добавление скрипта
	Редактирование настройки скрипта
	Удаление скрипта

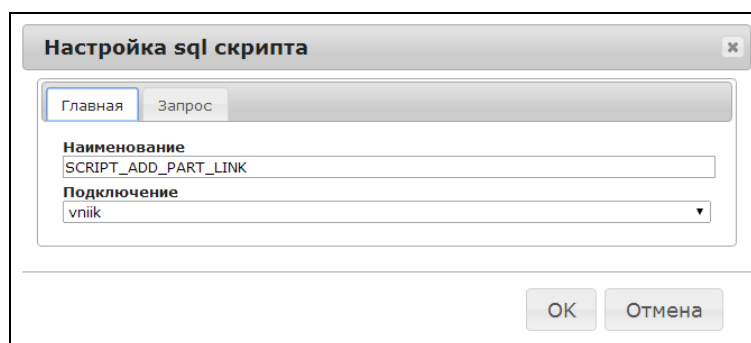


Рис. 4.2.16 Настройка скрипта.

Все что нужно сделать на первой закладке – указать, какое подключение использовать для работы скрипта. И, при желании, проставить осмысленное название на английском языке, что должно помочь в понимании смысла работы SQL-инструкции, которая заполняется на второй закладке.

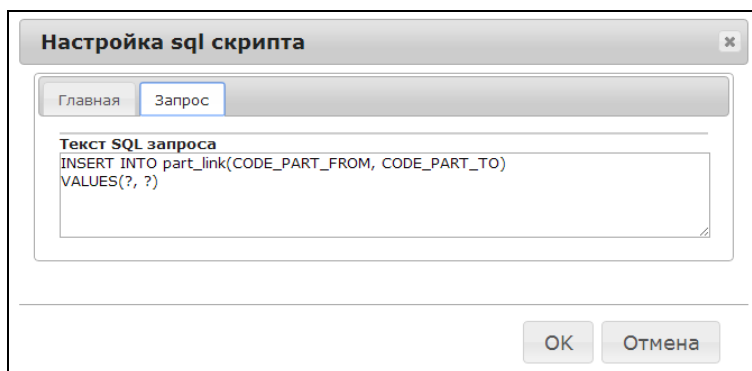


Рис. 4.2.17 Непосредственно написание скрипта.

4.2.6 Диалоги.

В процессе построения приложений часто есть необходимость организации стандартных диалогов, как например, ввод строки текста пользователем, выбор из предоставляемого списка, ввод даты и так далее. Все эти диалоги могут быть настроены в представляемой системе. Диалог, это всплывающее окно (хотя если капнуть в реализацию, оно им фактически не является), у которого есть кнопка "Ок" и кнопка "Отмена". Пока пользователь не обработал такое окно (не ввел данные) дальнейшая работа невозможна. Такие окна называют еще модальными.

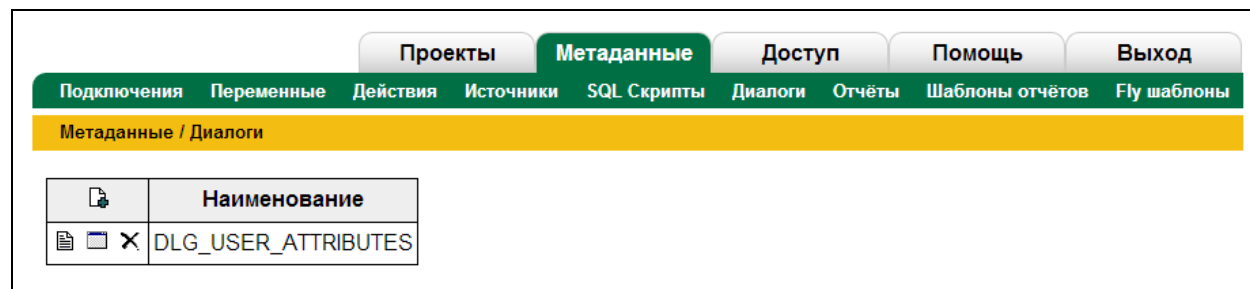


Рис. 4.2.18 Настройка стандартных диалогов.

Каждое такое окно настраивается в разделе "Диалоги". Настроенный единожды диалог имеет свой идентификатор "Наименование" и может вызываться в различных местах проекта.

Как видно из таблицы, приведенной ниже, действия довольно стандартные:

- ✓ Добавить
- ✓ Редактировать
- ✓ Удалить

Таблица 4.2.10 Настройка диалогов.

Кнопка	Описание
	Добавление диалога
	Редактирование настройки диалога
	Удаление диалога

Вызов диалога происходит из настроек проекта, а так же непосредственно из исходного кода, как показано на рисунке ниже. Наиболее сложное в нем – это функция обратного вызова, которая вызывается по окончании работы диалога. Такие макароны необходимы в связи с асинхронностью выполнения функций в браузерах.

```

$this->dbgridPlugin->addToolButtonCallUserDialogGrid(
    'parameters.gif',      // картинка кнопки
    'Атрибуты',           // подсказка кнопки
    'DLG_USER_ATTRIBUTES', // имя диалога
    'Атрибуты',           // подпись диалога
    'users/applyUserAttr', // адрес обратного вызова
    'function() {dbgrid.refresh();}', // функция обратного вызова
    array('CODE_USER',     // поля значения которых надо взять из грида
          'CODE_DEPARTURE',
          'CODE_POST'),
    array()                // свои параметры
);

```

Рис. 4.2.19 Вызов диалога из исходно кода проекта.

Настройка основных параметров довольно простая. Указывается идентификатор, описание, подписи и алиас подключения к базе данных.

Рис. 4.2.20 Редактирование настройки диалога.

Что же касается полей диалога, то их может быть много и их поведение зависит от настраиваемых переключателей, представленных в таблице ниже.

Таблица 4.2.11 Настройка переключателей полей.

Кнопка	Описание
READONLY	Только чтение. Выводимое поле недоступно для изменения.
DEFAULT <значение>	Значение по умолчанию. Перед началом ввода данных это поле принимает значение, указанное через пробел после переключателя
SMALLREF <Имя таблицы>-<Поле для отображения>-<Поле первичного ключа>	Выпадающий список из таблицы. По указанному на первой закладке алиасу система извлекает из таблицы поле для отображения и выбранное значение передается в виде кода из поля первичного ключа. В следующих версиях планируется организовать ввод SQL-запроса вместо имени таблицы (будет работать и то и другое)
MEMO [<кол-во строк>]	Текстовое поле на несколько строк. Количество строк указывается через пробел.
CHECKBOX	Флажок. Поле для ввода логического значения Да/Нет
DATE	Поле типа дата
LISTBOX <значение 1>-<значение 2>...<значение N>	Выпадающий список из фиксированных значений. Требуемые значения необходимо перечислить через дефис ('-')
FOCUS	Автоматически устанавливать фокус на это поле
WINDOW <имя контроллера>-<имя первичного ключа>	Выбор из справочника. Обработывает фрагмент php-кода имеющий имя контроллера. Обычно это имя файла (без расширения PHP)
FILENAME <имя файла>	Загрузка файла с переименованием. Чаще всего используется в виде

	<имя папки>/[<поле первичного ключа>]
PAGE <наименование закладки>	Закладка, на которой находится поле в диалоге редактирования

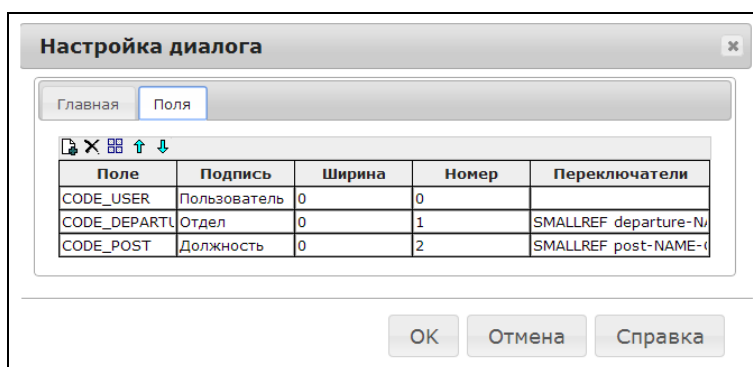


Рис. 4.2.21 Редактирование настройки диалога.

Таблица 4.2.12 Настройка полей диалога.

Кнопка	Описание
	Добавление поле
	Удаление поле
	Удалить все поля
	Переместить поле вверх
	Переместить поле вниз

Ниже на рисунке показана работа диалога на основе двух настроенных полей в виде выпадающего списка. Система сама собирает внутри SQL-запросы, извлекает данные, готовит результат и передает его в нужное место в виде первичного ключа.

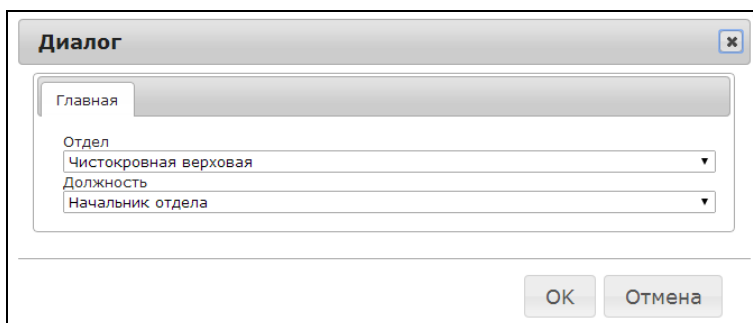


Рис. 4.2.22 Пример диалога в действии.



Диалоги были добавлены сравнительно недавно в MiniWeb2, поэтому все необходимые вызовы еще не реализованы. Однако они появятся в самое ближайшее время. Читайте о них в аннотациях к релизам.

4.2.7 Отчёты.

Отчеты – наиболее сложная часть построения системы. В простом варианте – это шаблоны, подключение данных, вывод. Но могут быть более сложные случаи с группировкой, подчиненными источниками и так далее. Сложные ситуации могут быть разрешены программным путем. Но для большинства случаев достаточно обеспечить связывание источников и подавляющее большинство отчетов не потребует дополнительного программирования.

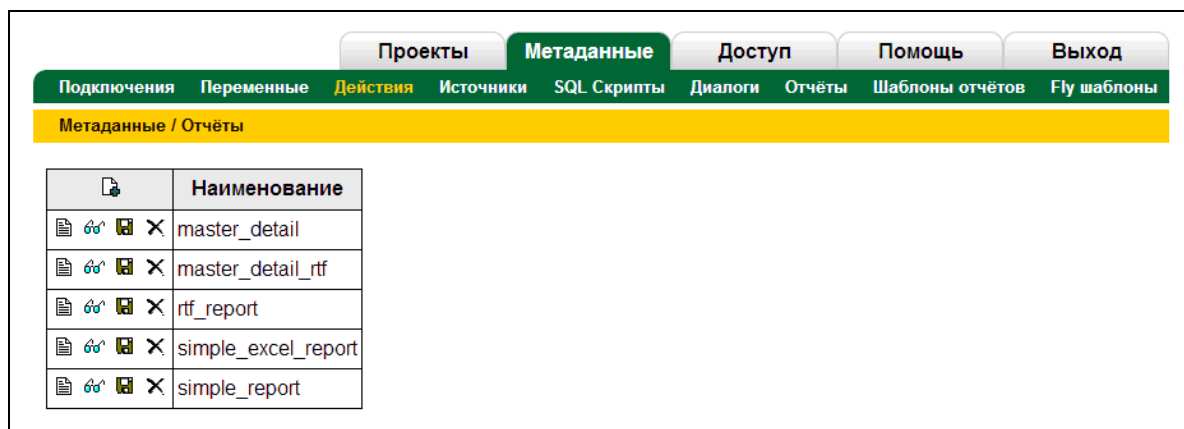


Рис. 4.2.18 Отчеты проекта.

Таблица 4.2.13 Настройка отчета.

Кнопка	Описание
	Добавление отчета
	Настройка поведения отчета
	Удаление отчета
	Просмотр получаемого отчета
	Скачать получаемый отчет

Просмотр и скачивание отчета – это очень похожие функции. Однако, просмотр не всегда возможен, вот почему мы сделали скачивание помимо просмотра непосредственно в браузере.

Рис. 4.2.19 Настройка главных параметров.

На первой закладке указывается, как идентифицируется отчет – его наименование. Указывается, какой шаблон использовать. Проставляется тип отчета. Типы отчетов на момент выпуска документации (далее будут расширяться):

- ✓ Excel
- ✓ Text
- ✓ RTF

Соответственно каждая технология имеет свои особенности реализации. Как следует из названий, каждый шаблон настраивается соответствующими инструментами MS Excel, MS Word, MS Notepad. Наиболее универсальный вывод в TXT. Это может быть не только текстовый файл, но и INI и HTML, XML, а так же специализированные файлы для издательских систем (Quark Express, PageMaker и другие)

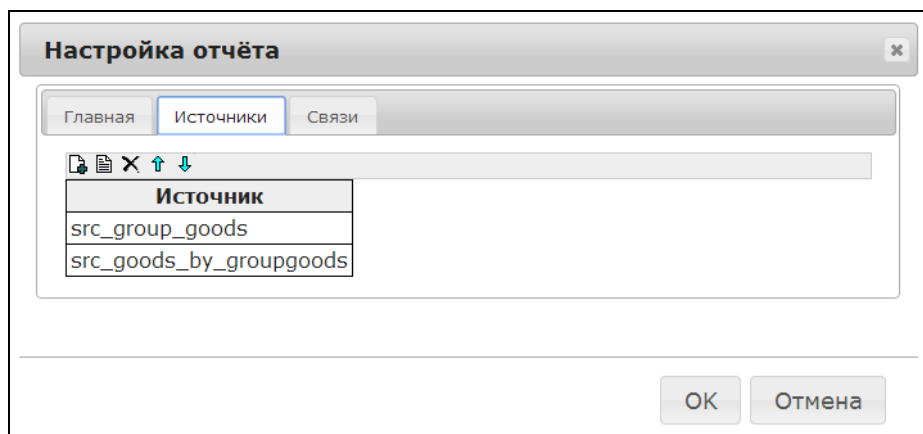


Рис. 4.2.20 Указание перечня источников.

На закладке Источники необходимо проставить используемые источники. К этому моменту источники уже должны быть в системе через соответствующий пункт, описанный в предыдущей главе (4.2.4 Источники). Порядок следования источников так же важен. Его можно поменять соответствующими кнопками.

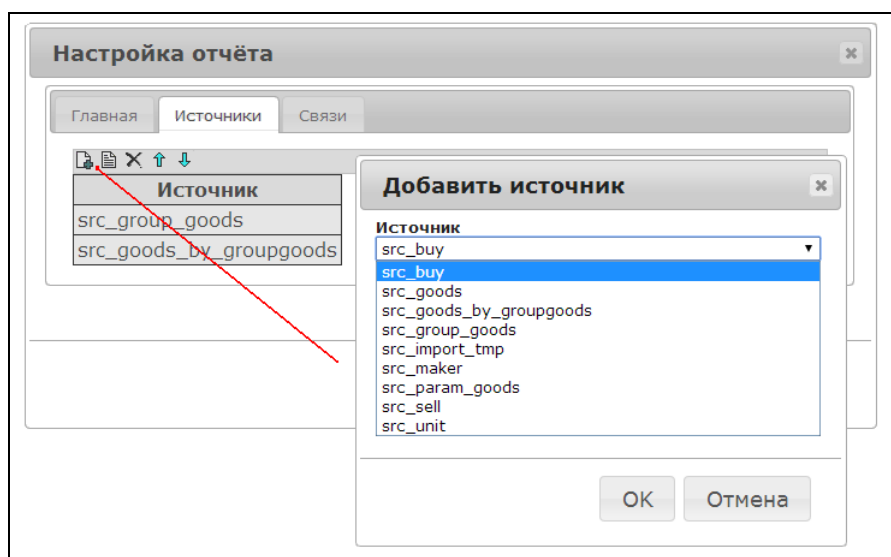


Рис. 4.2.21 Добавление необходимых источников.

Что интересно в связях, так это, что не видно по какому полю следует связывать источники между собой. Для такой связи необходимо прописать в SQL-запросе макроподстановку [Имя_истчника.Имя_поля]Пример макроподстановки приведен в листинге ниже.

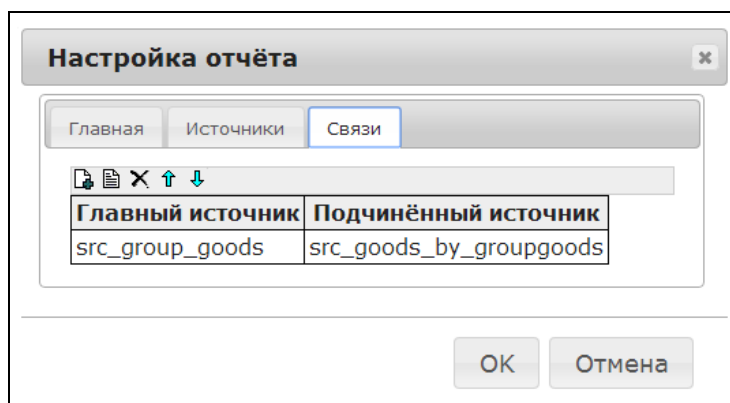


Рис. 4.2.22 Добавление связей источников.

```

SELECT
  CODE_GOODS,
  CODE_GROUP_GOODS,
  NAME
FROM GOODS
WHERE CODE_GROUP_GOODS = [src_group_goods.CODE_GROUP_GOODS]

```


Листинг. 4.1 Исходный код SQL-запроса с макроподстановкой для связи.

В результате всех усилий получается отчет в таком виде, как например, на рисунке 4.2.22. Это вариант с ведуще-детализированным подходом. Описанные здесь возможности можно посмотреть вживую на поставляемых с MiniWeb примерах.



Рис. 4.2.21 Просмотр получаемого отчета.



Чтобы исправить шаблон отчета, его надо сначала скачать в разделе шаблоны по кнопке .

Может показаться, что несколько неудобно создавать источники в одном месте, шаблоны в другом и собирать их все вместе в третьем месте в разделе Отчеты. Мы совершенно с вами согласны. Мы над этим работаем и надеемся в будущем предоставить больше возможностей для редактирования непосредственно на месте

4.2.8 Шаблоны отчётов.

Сами шаблоны отчетов готовятся по определенному синтаксису вне представляемого интерфейса. Их можно подготовить, например, с помощью программы MS Excel. Он формируется обычным способом, затем делается операция “Сохранить как Веб-страница”. Поля, которые проставляются в нужных местах записываются в квадратных скобках как **[Таблица.Имя поля]**.

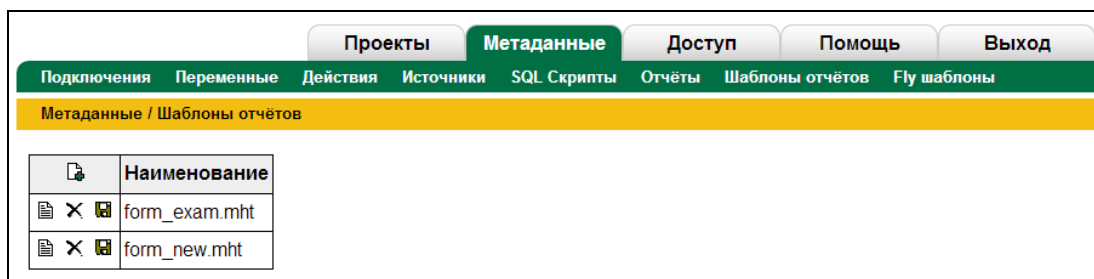


Рис. 4.2.19 Шаблоны отчетов.

Шаблоны, как видно из иллюстраций и таблиц можно загружать на сервер, скачивать для модификации, забрасывать обратно на сервер. Так же удалять, менять их имена.

Таблица 4.2.10 Функции настройки шаблонов отчетов.

Кнопка	Описание
	Добавление шаблона
	Редактирование настройки шаблона
	Удаление шаблона
	Скачать шаблон

Файл шаблона загружается на сервер непосредственно в диалоговом окне, показанном ниже. При заброске происходит переименование файла шаблона на указанное в поле ввода имени. При повторной заброске предыдущие версии шаблонов затираются, поэтому будьте аккуратны в своих действиях. Подобные изменения потенциально могут остановить работу системы по выводу отчетов.



При модификации шаблонов настоятельно рекомендуем сохранять предыдущие их версии. Тогда в случае отказа работы шаблона всегда можно забросить на сервер предыдущий его вариант.

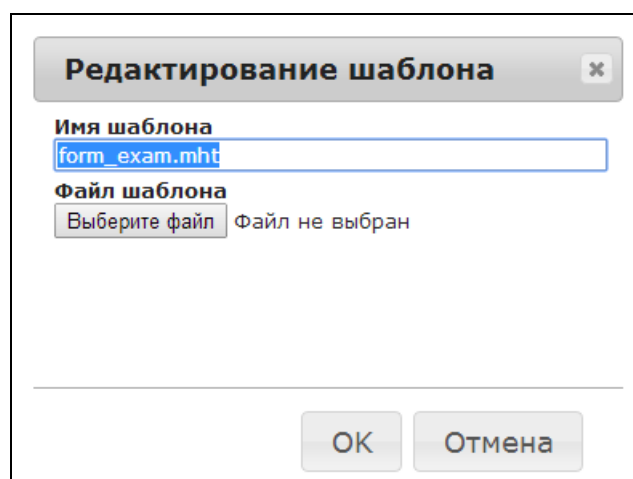


Рис. 4.2.20 Настройка шаблона.

При настройке шаблона необходимо четко придерживаться спецификаций, которые управляют его поведением. Например запись **[roll src_goods]** означает, что происходит прокрутка в этой части шаблона относительно источника **src_goods**, который описан в разделе метаисточников. Надпись **[/roll src_goods]** означает, что скроллинг источника завершен. Последняя инструкция может показаться избыточной, так как источник был указан в начале. Но если вы начинаете работать сразу с несколькими источниками, то может присутствовать вложенность и эта информация необходима.

Ниже приведен пример подготовки отчета средствами MS Excel. Как вы можете убедиться, подготовить его не составляет большого труда. Тем более что здесь же можно сделать визуально все необходимое оформление, вставку картинок и так далее. Особенно это актуально, когда отчеты полностью готовы, но периодически требуется улучшать их внешний вид.

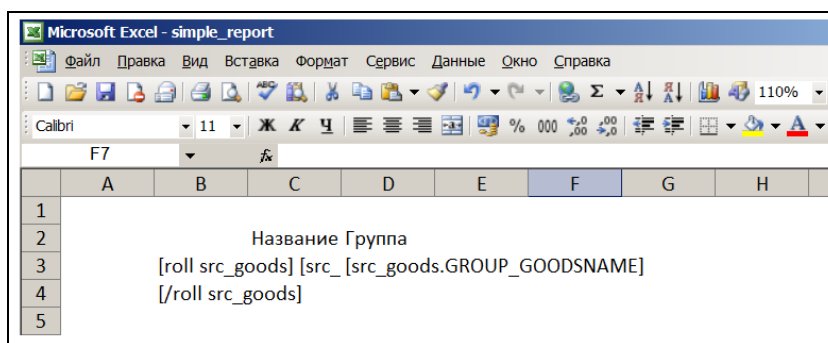


Рис. 4.2.21 Настройка шаблона для MS Excel.

Тестовые шаблоны хороши своей простотой и находят свое применение при подготовке материалов с дельнейшей версткой в виде книг, журналов и так далее. Они просты в подключении, обладают высокой степенью надежности и коротким периодом ввода в эксплуатацию.

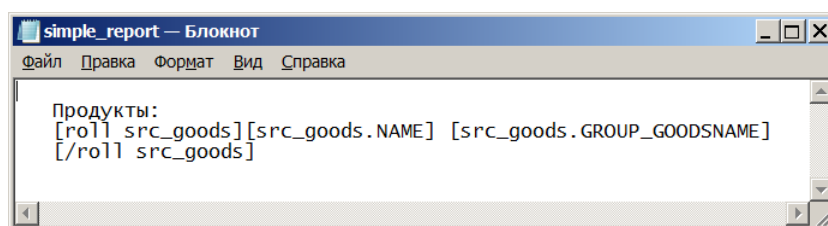


Рис. 4.2.22 Настройка шаблона для обычного текста.

4.2.9 Fly шаблоны.

Fly шаблоны готовятся средствами программы ETNI Fly. Их настройка описывается в отдельном руководстве www.silentium.ru/download/etni_fly_guide.pdf Эти шаблоны обрабатываются специальными механизмами и на выходе дают готовый etni-файл для его открытия на рабочей станции программным обеспечением для MS Windows.

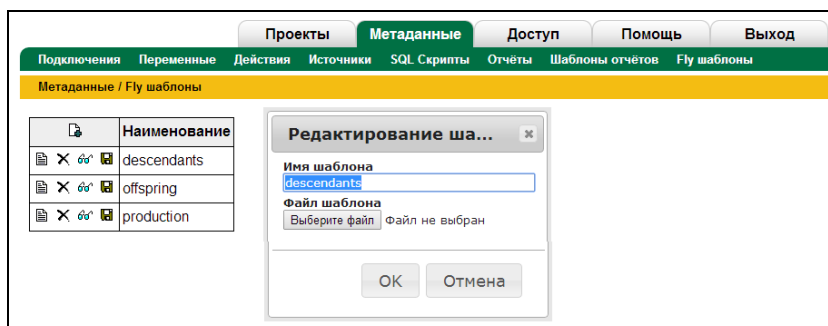







Рис. 4.2.23 Настройка Fly-шаблонов.



Необходимо заранее проинсталлировать программу ETNI Fly. Ее можно скачать по адресу: www.silentium.ru/download/fly_install.exe

Таблица 4.2.11 Функции для работы с Fly шаблонами.

Кнопка	Описание
	Добавление шаблона
	Редактирование настройки шаблона
	Удаление шаблона
	Тестирование работы шаблона
	Скачать шаблон

Здесь доступен полный перечень манипулирования шаблонами, в том числе переименование, скачивание, заброска на сервер и что очень важно – их пробный запуск. При вызове функции ” **Тестирование работы шаблона**” происходят необходимые подстановки параметров, заполнение данными, сжатие получаемого файла и его загрузка на рабочую станцию, так как если бы это выполнил обычный пользователь. Далее этот файл открывается программой ETNI Fly, установленной на компьютере.

4.3 Доступ.

Управление доступом в проекте MiniWeb2 может быть двух вариантов, причем может использоваться как отдельно по каждому виду (упомянутом в списке ниже), так и в комбинированном варианте:

- ✓ Управление доступом извне (внутри кода проекта)
- ✓ Управление доступом с помощью предоставляемой оболочки

Помимо самой настройки разработчику предоставляются отдельные функции авторизации и проверки доступа. Подробнее об этих функциях необходимо смотреть в поставляемых спецификациях.

Понятие сущности

Для идентификации сущности может быть два варианта. Первый просто по имени сущности (например, имя пункта меню или имя метаисточника), либо по двум идентификаторам: имя сущности и параметр. Параметров может быть несколько, но они все равно записываются в одно поле через разделитель. Параметр может выбираться из списка, но может и вводиться напрямую в поле.

К сущности нужно ввести описание – отдельно поле для того чтобы сориентировать системного администратора или опытного пользователя, что это за сущность. В этом плане так же будет подспорьем группа сущностей, например: метаисточник, пункт меню, кнопка, страница на сайте, предприятие, отдел и так далее относительно предметной области.

Так же есть смысл передавать список возможных сущностей извне (списком). Идентификаторы сущностей и их примечания не должны повторяться.

Основные потоки

- ✓ Справочник пользователей
- ✓ Справочник ролей
- ✓ Справочник сущностей
- ✓ Справочник групп сущностей
- ✓ Справочник параметров
- ✓ Предоставление ролей
- ✓ Предоставление сущностей для ролей

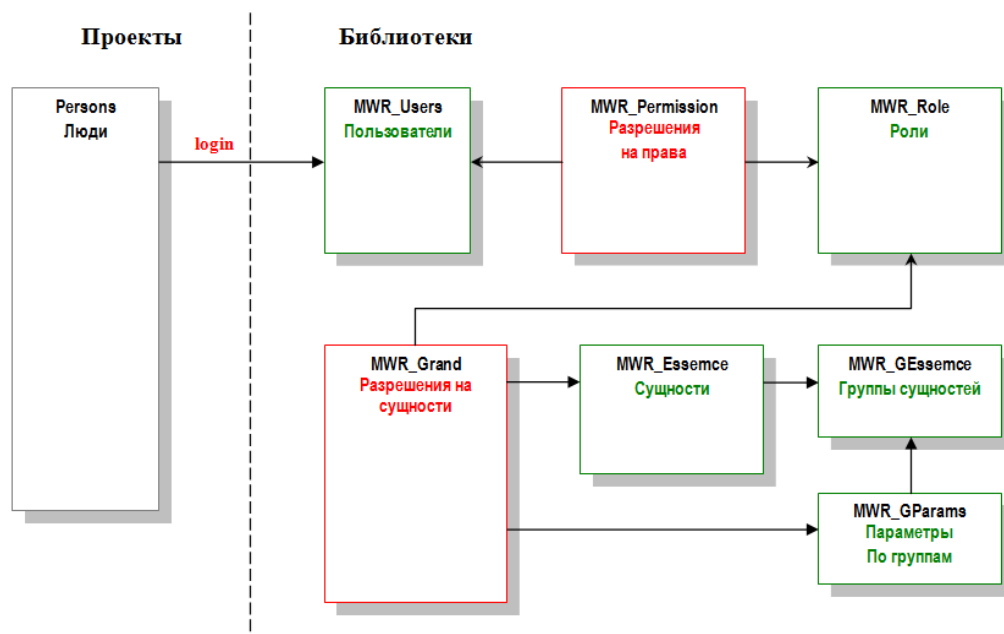


Рис. 4.3.1 Управление доступом MiniWeb2

Внешние функции модуля, вызываемые в исходном коде проекта (внутри контроллеров):

- ✓ Регистрация пользователей
- ✓ Авторизовать пользователя
- ✓ Сообщить, есть ли у пользователя права на сущность с передаваемым параметром

4.3.1 Настройка

Настройка может иметь множество пунктов. На данном этапе, все, что требуется настроить – это указать в какой базе данных находятся физически таблицы для управления доступом. Для этого заранее должен быть настроен такой алиас в разделе "Подключения".

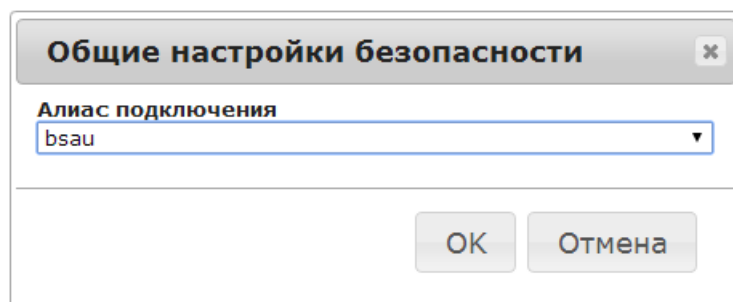







Рис. 4.3.2 Выбор базы данных для настройки ролей.

4.3.2 Пользователи

Пользователей внутри оболочки очень просто добавлять, так как для этого есть все условия. Однако часто бывает необходимо добавлять пользователей программно извне. Для этих целей так же есть набор необходимых функций, описанных в спецификации к данному руководству.

Визуально работа с пользователями достаточно проста и помимо удаления, добавления, редактирования можно выставлять и менять пароль, предоставлять и отбирать роли системы. То есть даются не сами разрешения на сущности системы, а роли, которые в свою очередь обладают такими правами.

Таблица 4.3.1 Функции управления пользователями.

Кнопка	Описание
	Добавление пользователя
	Редактирование пользователя
	Удаление пользователя
	Настройка пароля
	Предоставление ролей

Рекомендуем сначала настраивать роли и уже, затем предоставлять эти роли пользователям. Как видно на рисунке ниже, каждый пользователь должен быть активирован. Такая активация осуществляется методом отправки почтового сообщения и уже, затем переходим по отправленной ссылке. Такой процесс активации, как правило, автоматический.






















Доступ / Пользователи					
	Логин	ФИО	e-mail	Дата регистрации	Активирован
   	administrator		administrator@mail.ru	2013-04-19 18:51:15	Да
   	dean		email@mail.ru	2013-04-19 18:50:12	Да
   	dephead		dephead@mail.ru	2013-04-19 18:50:56	Да
   	methodologist		methodologist@mail.ru	2013-04-19 18:50:34	Да
   	student		email@mail.ru	2013-04-19 18:49:56	Да

Рис. 4.3.3 Управление пользователями системы.

Разрешения

Наименование	Примечание	Разрешение	Делегировать
Администратор	Полный доступ к управлению контентом	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Просмотр контента	Позволяет просматривать сайт	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Редактор информации об институте	Позволяет редактировать общеинститутскую информацию	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Редактор контактов	Позволяет редактировать контактную информацию	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Редактор науки	Позволяет редактировать научные статьи	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Применить

Отмена

Рис. 4.3.4 Предоставление ролей пользователям.



Роли, представленные на рисунке, добавляются на следующей закладке. Управление ими описано в следующей главе.

4.3.3 Роли

Роли в данном месте не имеют особых настроек. Просто это перечень ролей, который можно расширять, модифицировать. Если указано, что роль выставляется по-умолчанию, это значит что при добавлении нового пользователя права на нее включаются автоматически. Это часто требуется для ситуации, когда на портале появляются новые регистрации участников и их права нужно обеспечить немедленно без вмешательства администратора. Следует осторожно обращаться с данной возможностью, так как это может быть причиной несанкционированного доступа.




Доступ / Роли			
	Наименование	По умолчанию	Примечание
 X	Администратор	Нет	
 X	Декан	Нет	
 X	Заведующий кафедрой	Нет	
 X	Методист	Нет	
 X	Студент	Нет	

Рис. 4.3.5 Формирование ролей.

Таблица 4.3.2 Управление перечнем ролей

Кнопка	Описание
	Добавление роли
	Редактирование роли
X	Удаление роли



Роль по-умолчанию важна только для вновь добавляемых пользователей. К старым пользователям это не относится.

4.3.4 Типы сущностей

Типы сущностей, как правило, одни и те же от проекта к проекту, однако при необходимости их можно наращивать с помощью представляемой закладки.









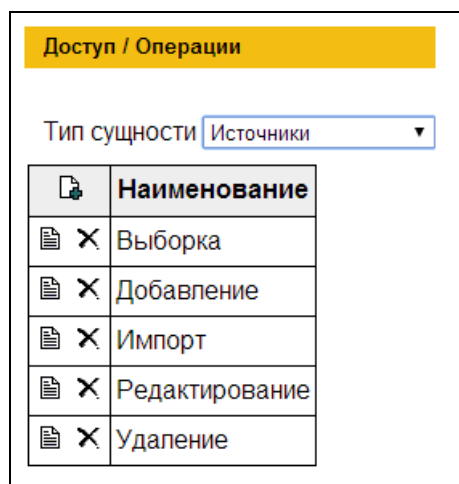
Доступ / Типы сущностей	
	Наименование
 X	SQL скрипты
 X	Источники
 X	Кнопки
 X	Пункты меню
 X	Страницы
 X	Таблицы
 X	Элементы страницы

Рис. 4.3.6 Типы сущностей.

Над сущностями выполняются все те же действия: добавление, редактирование, удаление.

4.3.5 Операции

Для каждого типа сущности свой набор операций, которые с ней можно производить. Например, как указано на рисунке внизу, источник позволяет делать извлечение данных (выборка), добавление записей, удаление записей, импорт данных. Часто операция может быть только одна и как ее лучше назвать как раз указывается в этом месте.



Иконка	Наименование
📄 ✕	Выборка
📄 ✕	Добавление
📄 ✕	Импорт
📄 ✕	Редактирование
📄 ✕	Удаление

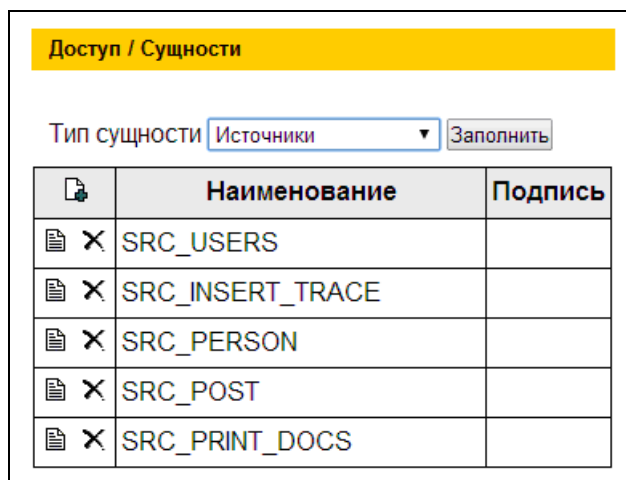
Рис. 4.3.7 Операции.

4.3.6 Сущности

Здесь, как показано на рисунке ниже заносятся все сущности, участвующие в проекте. В результате развития каждого проекта добавляются все новые объекты управления (такие как источники, пункты меню кнопки). Все эти объекты, названные общим понятием сущность, необходимо добавлять для дальнейшего разграничения прав доступа. Их можно:

- ✓ Добавить вручную
- ✓ Заполнять автоматически

Для заполнения автоматически достаточно нажать на кнопку "Заполнить". Появятся, например, новые настроенные источники. Не для всех сущностей возможно автоматическое заполнение. Об этом должен позаботиться разработчик. Если вы добавили тип сущности, которого не было раньше в системе, то MiniWeb может просто не знать, откуда брать обновленный список. На момент написания документации возможно заполнение только источников и SQL-скриптов. Далее такое заполнение будет расширено.



Иконка	Наименование	Подпись
📄 ✕	SRC_USERS	
📄 ✕	SRC_INSERT_TRACE	
📄 ✕	SRC_PERSON	
📄 ✕	SRC_POST	
📄 ✕	SRC_PRINT_DOCS	

Рис. 4.3.8 Сущности.

4.3.7 Разрешения

Непосредственно на этой страничке предоставляются разрешения (права) на доступ к сущностям для той или иной роли. Собственно это пересечение (фильтр) между типом сущности и выбранной ролью. Ниже на рисунке представлена таблица, по строкам которой указываются сущности, а по столбцам операции над ними. В местах пересечения есть управление “Включить” или “Выключить”.

Таблица 4.3.3 Управление разрешениями

Кнопка	Описание
	Включить/выключить разрешение
	Включить разрешение по столбцу / по строке
	Выключить разрешение по столбцу / по строке
	Включить / выключить все

Для удобства так же можно действовать сразу по группам или столбцам. Так же в левом верхнем углу три кнопки действуют на все разрешения.

Доступ / Разрешения

Тип сущности Роль

Имя сущности	Подпись					
		Выборка	Добавление	Импорт	Редактирование	Удаление
SRC_DEAN		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_DEPARTMENT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_DISCIPLINE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_DOCS		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_DOCS_IN		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_DOCS_STUDENT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_DOCTYPE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_FACULTY		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_FIND_BARCODE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_FIND_DEPARTMENT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_FIND_METHODOLOGIST		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_FIND_NEXTWAY		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_FIND_USER		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRC_FIND_WAY		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рис. 4.3.8 Управление разрешениями.



Не забывайте нажимать кнопку применить , расположенную в правом нижнем углу. Так же можно отменить случайно сделанные изменения.

7. Примеры использования.










Простые примеры на PHP	
Простой пример	
Параметризованный запрос	
Постраничное разбиение	
Редактирование данных	
Использование DBGrid	
Использование DBLookup	
Использование DBEditDialog	
Авторизация и проверка прав	
Выполнение SQL скриптов	

Рис. 7.1 Перечень простых примеров использования MiniWeb2.








Примеры со Smarty	
Простой пример	
Примеры работы через базовые контроллеры	
Использование DBEditDialog	
Форма для выбора из справочника DBGrid	
Использование расширенного контроллера таблицы с поддержкой редактирования	
Авторизация и проверка прав	
Использование плагинов для расширения возможностей контроллеров	
Простой пример	
Вывод грида	

Рис. 7.2 Более сложные примеры со Smarty, использование контроллеров.










Диалоги редактирования AJAX	
Диалоги редактирования AJAX	
Окно COMPLEX	
Импорт данных	
Ввод мультимедиа	
Экспорт по шаблону	
Простой экспорт по шаблону	
Простой экспорт в эксель	
Загрузка шаблона на сервер	
Скачивание шаблона	
Ручная сборка отчёта	

Рис. 7.3 Использование Ajax, экспорт данных.

Ниже показан пример самого простого вызова MiniWeb2. Подробнее можно ознакомиться с примерами непосредственно скачивая соответствующие проекты в виде архивов. Исходный код в этих проектах содержит подробные комментарии практически по всем вызываемым строчкам кода


```

include_once dirname(__FILE__) . '/../client/lib/si_classloader.php';
TSiClassLoader::instance();

$miniWeb = new TSiMiniWebClient();
$miniWeb->setServer('http://localhost/miniweb2/service');
$miniWeb->setProject('example_firebird');
$miniWeb->login('admin', 'siadmin');

$sourceUnit = $miniWeb->createMetaSource('src_unit');
$sourceUnit->limit = 10;
$sourceUnit->offset = 1;
$sourceUnit->orderby = 'name';
$sourceUnit->open();

for ($i=0; $i<count($sourceUnit->data); $i++)
    echo $sourceUnit->data[$i]->NAME . '<br>';

```

Листинг. 7.1 Исходный код примера. Примечания здесь опущены.

8. Спецификации.

Здесь указывается полный перечень настроек для ключевых элементов интерфейса, таких как метаисточники, диалоги, действия для внутренних переменных. Эти спецификации постоянно пополняются, поэтому в документации может быть представлен не полный их перечень



Рекомендуем пользоваться подсказками непосредственно в окнах настроек по кнопке "Справка", расположенной в нижней части окна настройки того или иного элемента.

8.1 Переключатели полей метаисточника.

Таблица 4.8.1 Настройка переключателей полей источников

Кнопка	Описание
KEY	Первичный ключ
READONLY	Только чтение
DEFAULT <значение>	Значение по умолчанию
SETVALUE <значение>	Установить значение срабатывает также и при UPDATE, поле становится только для чтения
HIDE	Скрытое поле
SMALLREF	Выпадающий список из таблицы
MEMO [<кол-во строк>]	Текстовое (мемо) поле
CHECKBOX	Флажок для ввода логического значения (Да/Нет)
DATE	Поле типа дата
LISTBOX <значение 1>-<значение 2>...<значение N>	Выпадающий список из пользовательских значений
AUTOTODAY	Текущая дата
CURRENCY	Денежный формат (2 знака после запятой)

AUTONUMBER <поле даты>	Автономер в пределах текущего года
FOCUS	Автоматически устанавливать фокус на поле
WINDOW <имя контроллера>- [<имя первичного ключа>]	Выбор из справочника. Отрабатывает фрагмент php-кода имеющий имя контроллера. Обычно это имя файла (без расширения PHP)
FILENAME <имя файла>	Загрузка файла с переименованием Чаще всего используется в виде <имя папки>/[<поле первичного ключа>]
COMPRESS <тип сжатия>-<параметр сжатия>[-<папка сжатых файлов>]	сжатие загруженных файлов. Поддерживаемые типы сжатия: <ul style="list-style-type: none"> ✓ PICW - сжатие картинки по ширине. В качестве параметра указывается желаемая ширина в пикселях ✓ PICH - сжатие картинки по высоте. В качестве параметра указывается желаемая высота в пикселях Если папка сжатия не указана, то сжимается оригинальный файл. Когда папка указана, то сжатый файл перемещается в неё, а оригинал сохраняется.
PARAMETERS <параметр 1>-<параметр 2>... <параметр N>	Передача параметров в окно выбора из справочника применяется вместе с переключателем WINDOW. В контроллере их можно получить, добавив к названию параметра @
PAGE <наименование закладки>	Закладка, на которой находится поле в диалоге редактирования.
FORMAT <формат>	форматирование отображения поля. В качестве формата может быть использовано: %s для строк, %d для целых чисел, %f для вещественных чисел (пример: %15.2f) Для даты и времени: Y год, m месяц, d число, H часы, i минуты, s секунды Примеры: d.m.Y, H:i:s, Y-m-d

8.2 Переключатели полей диалогов.

Таблица 4.8.2 Настройка переключателей полей диалогов.

Кнопка	Описание
READONLY	Только чтение. Выводимое поле недоступно для изменения.
DEFAULT <значение>	Значение по умолчанию. Перед началом ввода данных это поле принимает значение, указанное через пробел после переключателя
SMALLREF <Имя таблицы>-<Поле для отображения>-<Поле первичного ключа>	Выпадающий список из таблицы. По указанному на первой закладке алиасу система извлекает из таблицы поле для отображения и выбранное значение передается в виде кода из поля первичного ключа. В следующих версиях планируется организовать ввод SQL-запроса вместо имени таблицы (будет работать и то и другое)
MEMO [<кол-во строк>]	Текстовое поле на несколько строк. Количество строк указывается через пробел.
CHECKBOX	Флажок. Поле для ввода логического значения Да/Нет
DATE	Поле типа даты
LISTBOX <значение 1>-<значение 2>...<значение N>	Выпадающий список из фиксированных значений. Требуемые значения необходимо перечислить через дефис ('-')
FOCUS	Автоматически устанавливать фокус на это поле
WINDOW <имя контроллера>- [<имя первичного ключа>]	Выбор из справочника. Отрабатывает фрагмент php-кода имеющий имя контроллера. Обычно это имя файла (без расширения PHP)
FILENAME <имя файла>	Загрузка файла с переименованием. Чаще всего используется в виде <имя папки>/[<поле первичного ключа>]
PAGE <наименование закладки>	Закладка на которой находится поле в диалоге редактирования

9. Эпилог.

Представленное здесь руководство не претендует на художественную ценность а так же содержит в себе множество идиоматических конструкций не воспринимаемых широким кругом читателей.

Однако мы решили что пусть лучше будет документация в подобном виде чем если бы ее не было вообще. Поднимаемые здесь проблемы не могут быть описаны техническим писателем, а у разработчиков никогда нет времени на такие вещи как подготовка документации.



По всем вопросам разработки систем, поддержке существующих комплексов пожалуйста обращайтесь по электронной почте mail@silentium.ru или по телефону **+7-491-225-17-50**

По этим же контактам вы можете заказать разработку необходимого программного обеспечения, организовать тренинг для собственных специалистов.

Компания Silentium
www.silentium.ru