

УТВЕРЖДЕН

SIMBIX.00003-01 33 ЛУ

ООО «ЭЛЕКТРОННОЕ ПРОЕКТИРОВАНИЕ»

(наименование организации разработчика)

SIMBIX.ИНТЕГРАЦИОННЫЙ ПОРТАЛ

(наименование программы)

ВЕРСИЯ 1.0

(версия программы)

SIMBIX.IP

(условное обозначение программы)

РУКОВОДСТВО РАЗРАБОТЧИКА

(наименование документа)

SIMBIX.00003-01 33

(условное обозначение документа)

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

В данном документе приводятся основные принципы и способы разработки проектов на платформе Simbix.Интеграционный портал, а также функциональное назначение программы, ее логическая структура, необходимые программные и технические средства для функционирования программы, вызов и загрузка программы, а также входные и выходные данные.

Перв. применен.

Справ. №

Подпись и дата

Инв. № дубл.

Взам. инв. №

Подпись и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подпись	Дата
Разраб.		Ясиновская Е.Д.		
Провер.				
Т. контр.				
Н. контр.				
Утв.				

Simbix.Интеграционный портал

Руководство разработчика

Лит.	Лист	Листов
	2	36

ООО «Электронное
Проектирование»

СОДЕРЖАНИЕ

1	ОБЩИЕ СВЕДЕНИЯ.....	4
1.1	Обозначение и наименование программы	4
1.2	Программное обеспечение	4
1.3	Языки программирования, на которых написана программа	4
2	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ.....	5
3	УСТАНОВКА SIMBIX AP	7
3.1	Установка	7
3.2	Создание проекта	12
3.3	Разворачивание архива проекта.....	13
3.4	Настройка.....	14
3.5	Проверка работоспособности.....	14
4	МОДЕЛЬ ДАННЫХ.....	15
4.1	Узел.....	15
4.2	Модуль.....	15
4.3	Метод.....	16
5	СОЗДАНИЕ ПЕРВОЙ СТРАНИЦЫ	17
5.1	Создание модуля	17
5.2	Создание типа	18
5.3	Вывод узла на странице. Создание метода.....	20
6	МНОЖЕСТВЕННАЯ ВЫБОРКА	21
7	ВТОРАЯ СТРАНИЦА.....	25
7.1	Выбор размещения метода	25
7.2	Добавление метода к типу "Новость". Вывод данных.	26
8	ПРОЦЕССОР.....	28
8.1	Создание RSS-трансляции. Задача	28
9	ОРГАНИЗАЦИЯ ДЕРЕВА УЗЛОВ	30
9.1	Создание дерева. Лента новостей	30
10	РАБОТА СО ССЫЛКАМИ	33

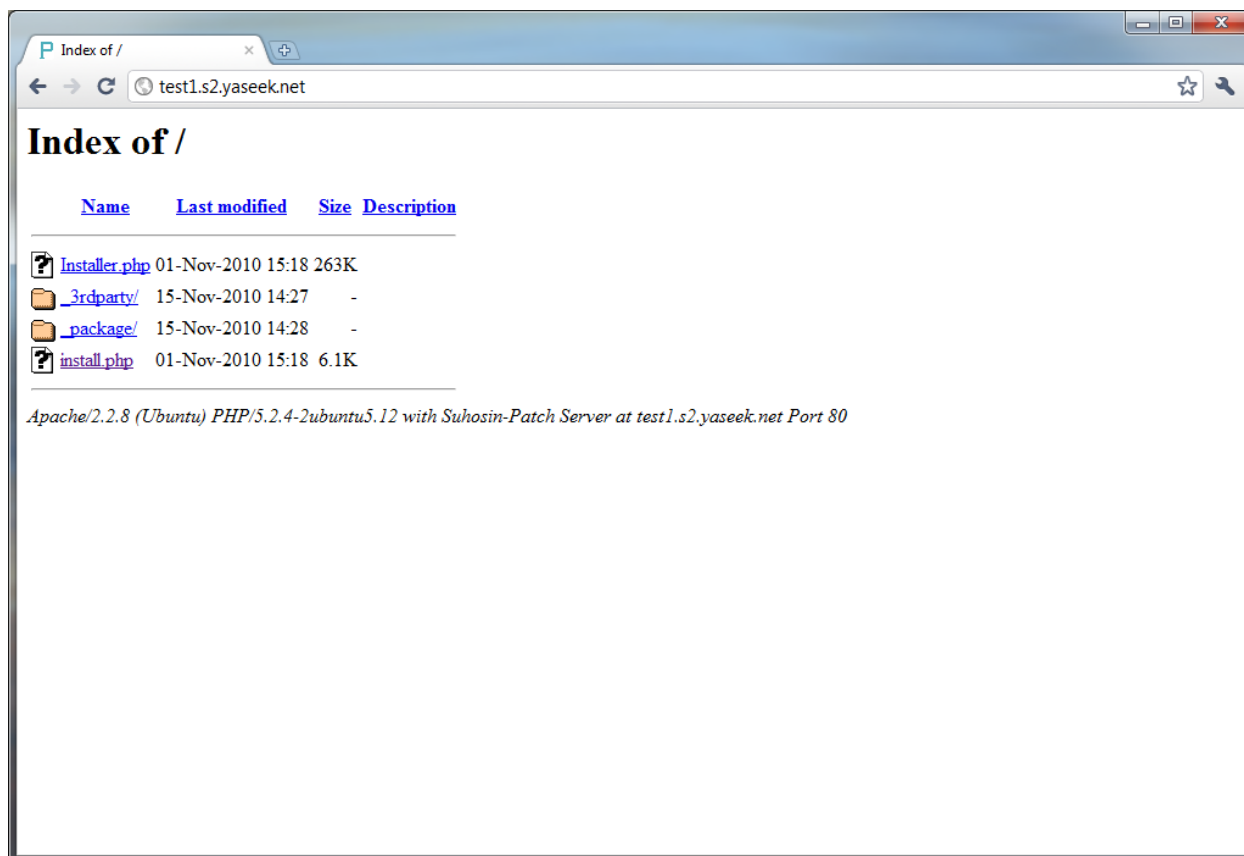
Име. № подл.	Подпись и дата
Взам. инв. №	Име. №
Подпись и дата	Подпись и дата

3 УСТАНОВКА SIMBIX AP

3.1 Установка

На сервере приложений должны быть установлена ОС Red Hat Enterprise Linux 5.5 или Ubuntu Server Edition 9.10 и предустановлены пакеты Apache 2 и PHP5 и СУБД MySQL 5.5 или PostgreSQL 8.3 или Oracle Database 10g. СУБД может быть также установлена на отдельном сервере (сервер базы данных).

Для установки программного ядра портала необходимо загрузить установочные файлы «install.php» (инсталлятор), «Installer.php» (библиотека инсталлятора) и «package.zip» (архив системы) и две директории «_3rdparty» (необходимые для работы сторонние библиотеки) и «_package» (пустая директория, необходимая для работы инсталлятора, в которую будет распаковано содержимое файла «package.zip») на сервер в директорию, доступную для веб-сервера.

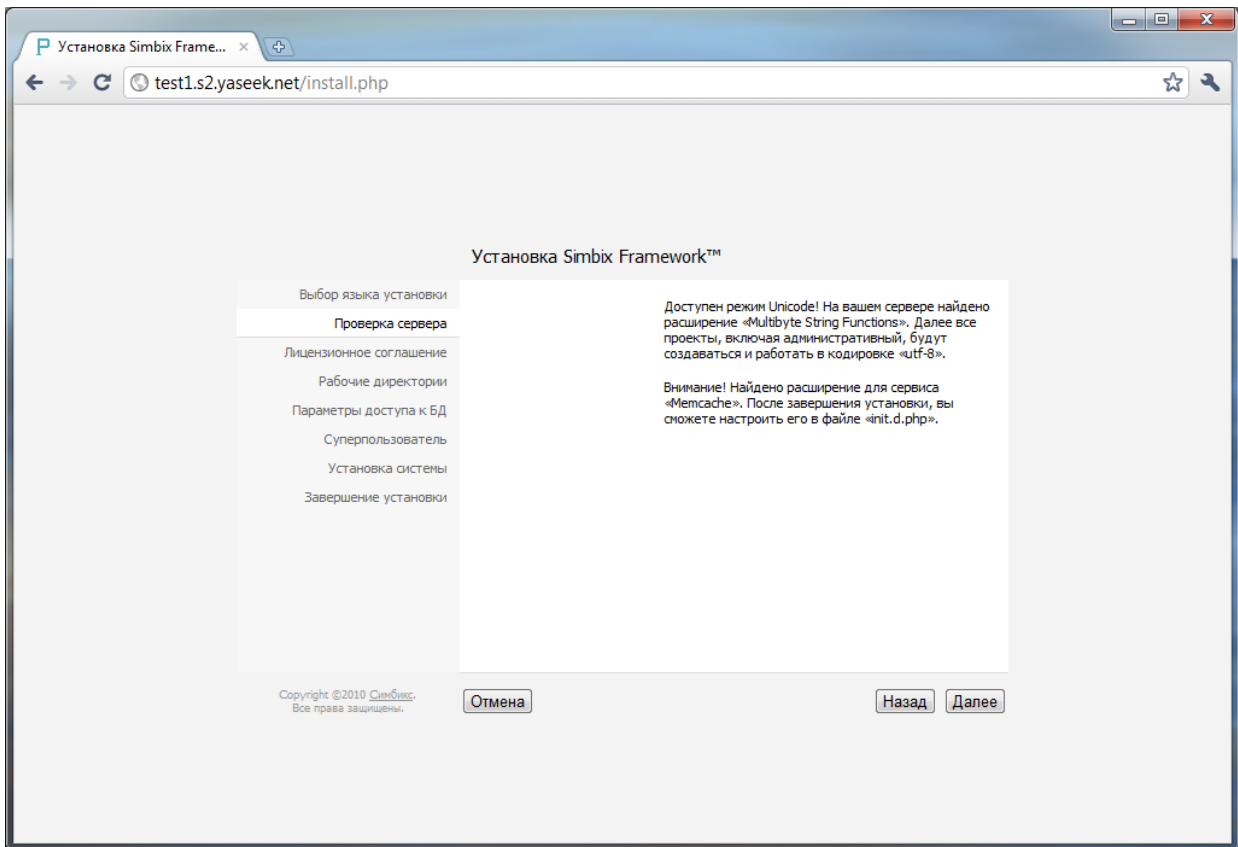
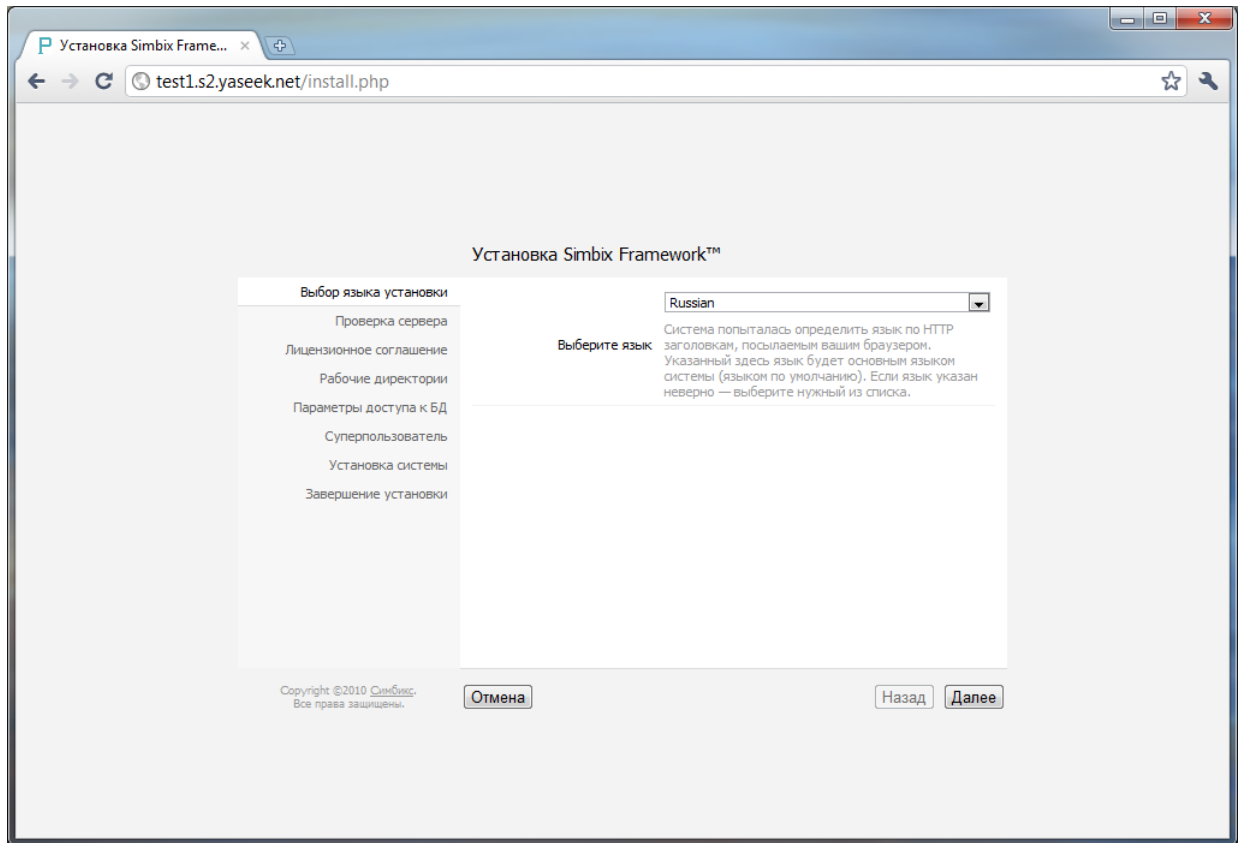


Процесс установки запускается путем обращения к файлу «install.php».

Далее необходимо выбрать язык установки (Русский или Английский)

Име. № подл.	Подпись и дата	Взам. инв. №	Инв. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						7

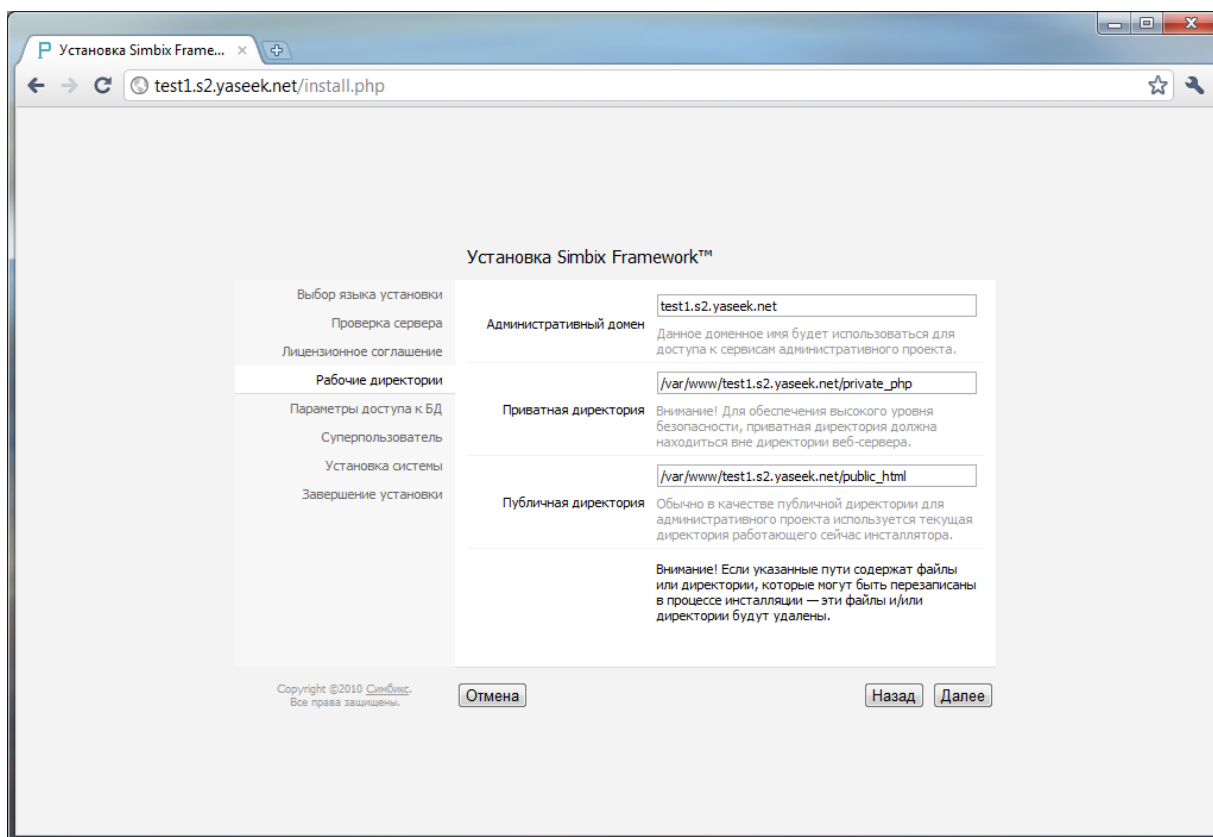


На шаге «Рабочие директории» необходимо указать домен административного проекта (если инсталлятор неверно определил его), а так же «приватную» и «публичную» директории. В «приватной» директории будут расположены исполняемые модули системы. Так же, для обеспечения высокого уровня безопасности, рекомендуется размещать приватную

Име. № подл.	Подпись и дата	Взам. инв. №	Инд. №	Подпись и дата

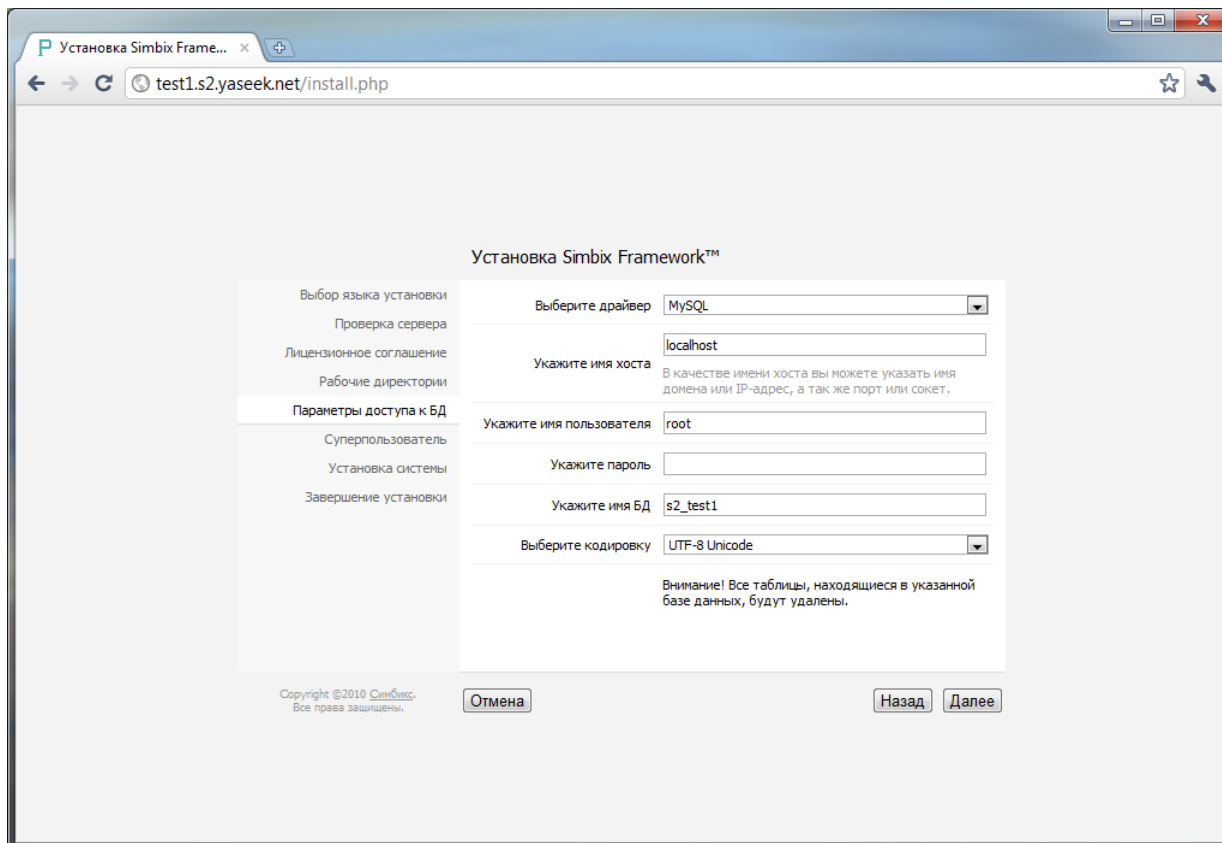
Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						8

директорию вне директории веб-сервера. В качестве публичной директории следует указать директорию, где после установки системы будет находиться административный проект (обычно это директория, в которой работает инсталлятор).



На шаге «Параметры доступа к БД» необходимо указать тип драйвера СУБД, хост или сокет СУБД, а так же имя пользователя, пароль и имя базы данных. Пользователь и база данных должны быть созданы заранее и указанный пользователь должен иметь привилегии создания таблиц, последовательностей и триггеров (функций) в БД. В случае, если БД уже содержит какие-либо данные – эти данные будут удалены.

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата



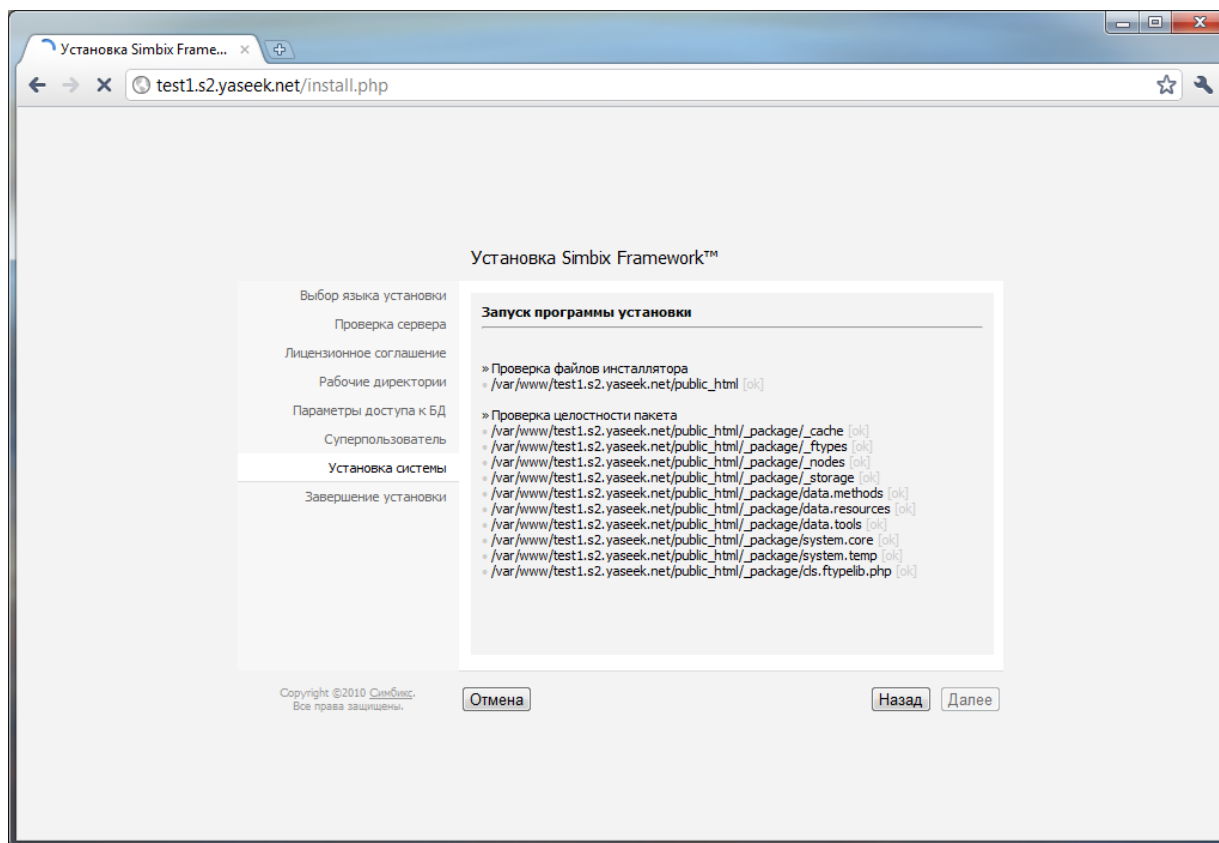
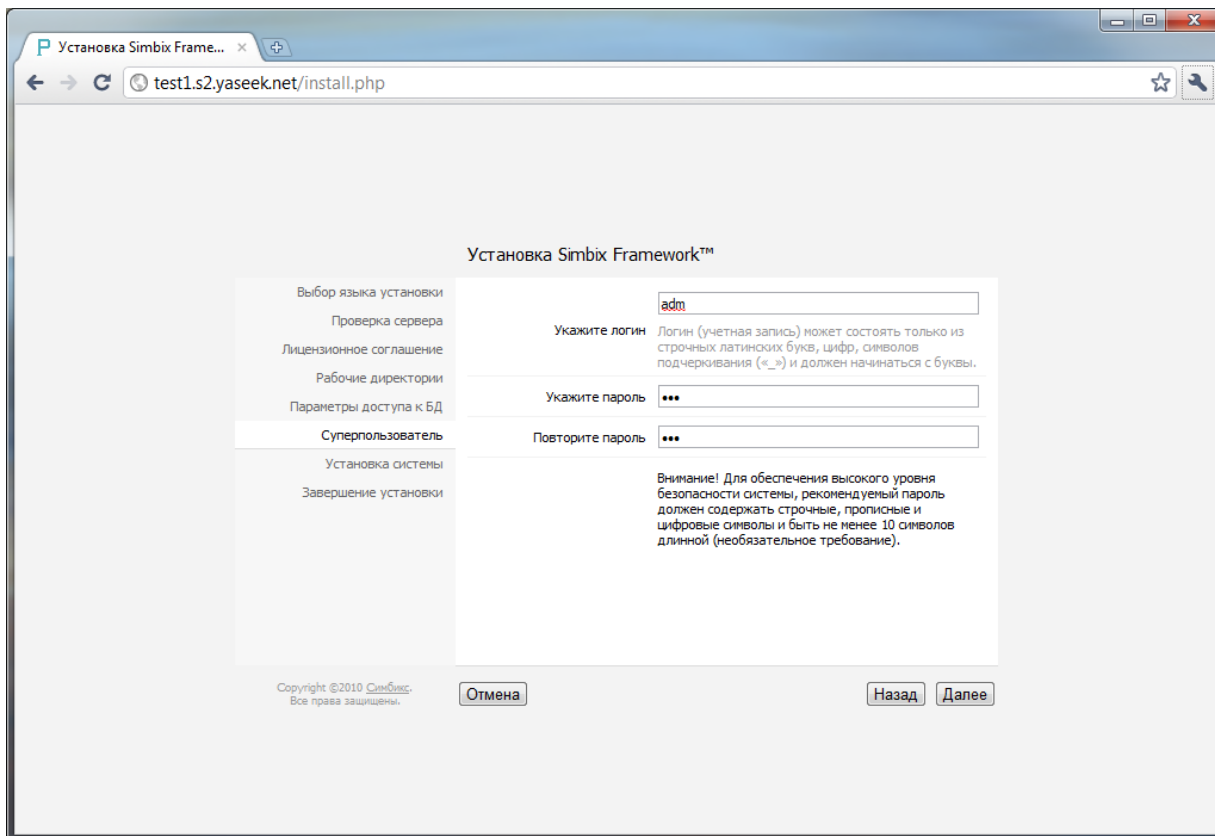
Далее необходимо создать учетную запись суперпользователя. Данный пользователь системы и обладает абсолютно полным набором прав. Исключительными привилегиями суперпользователя являются:

- создание нового проекта;
- создание (привязка) первой администраторской группы в проекте;

Все остальные привилегии суперпользователя доступны на всех или некоторых уровнях остальных полномочий.

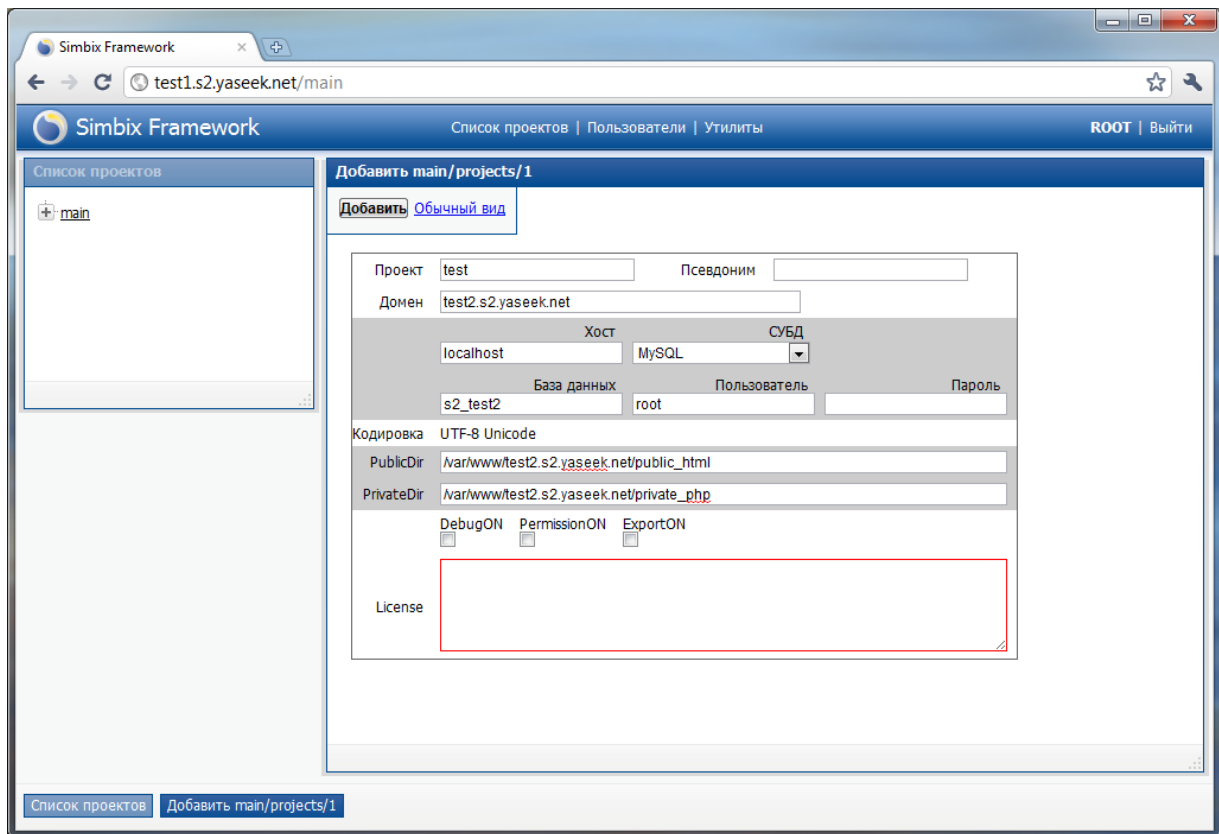
Име. № подл.	
Подпись и дата	
Взам. инв. №	
Инв. №	
Подпись и дата	

					SIMBIX.00003-01 33	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10



После завершения установки программного ядра портала система предлагает вам войти в административный проект. Это можно сделать при помощи учетной записи суперпользователя, созданной на предыдущем шаге установки.

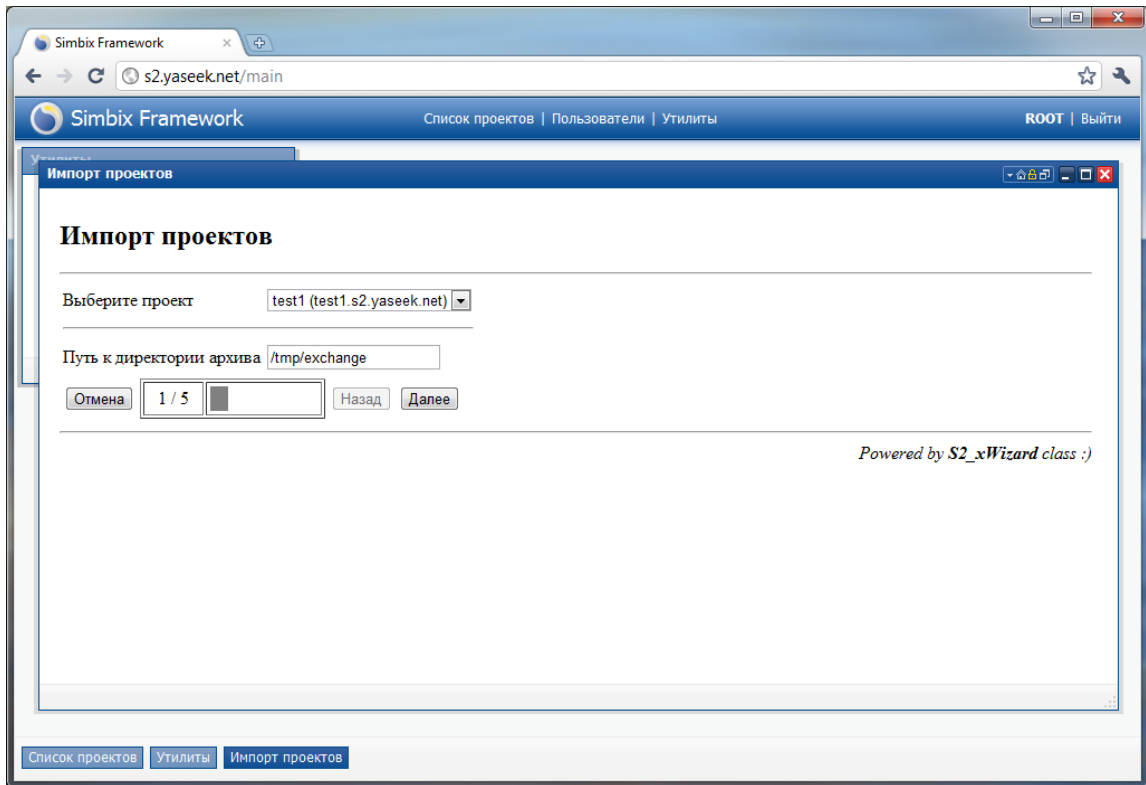
Име. № подл.	Подпись и дата	Взам. инв. №	Име. №	Подпись и дата



3.3 Разворачивание архива проекта

Для того, чтобы развернуть архив проекта необходимо кликнуть на ссылку «Утилиты» в верхней части интерфейса и в открывшемся дереве утилит выбрать пункт «Импорт проектов». Далее, в открывшейся форме необходимо выбрать проект, в который будет произведен импорт и указать директорию, в которой располагаются файлы архивов проектов программного ядра портала. Далее, следуя инструкциям «помощника» необходимо выбрать файл архива и нажать кнопку «Развернуть».

Име. № подл.	Подпись и дата	Име. №	Подпись и дата



3.4 Настройка

Некоторые параметры программного ядра портала могут быть настроены более тонко, для полного соответствия требованиям заказчика. Все они находятся в файле {путь к приватной директории}/_core/init.d.php. Среди них:

- S2_DEFAULT_LANGUAGE - Код языка системы по умолчанию
- S2_SESSION_LIFE - Время жизни пользовательской сессии
- S2_ERROR_REPORTING - Назначает уровень ошибок PHP
- S2_TIMEZONE - Идентификатор системной временной зоны
- S2_GZCOMPRESS - Сжимать ли содержимое страниц
- S2_BUSY_LIMIT - Максимальный уровень нагрузки на сервер
- S2_EXECUTION_TIME - Максимальное время выполнения процессов
- S2_MEMCACHE_USE - Использовать модуль Memcached
- S2_EXCHANGE_PATH - Путь "по умолчанию" для создаваемых архивов

3.5 Проверка работоспособности

Обратиться браузером по адресу домена, который был указан при создании проекта и дождаться загрузки главной страницы ФПО.

Ине. № подл.	Подпись и дата
Взам. ине. №	Ине. №
Подпись и дата	Подпись и дата

4 МОДЕЛЬ ДАННЫХ

4.1 Узел

Основным элементом модели данных в Simbix.Интеграционный портал является **узел**. Не давая отвлеченных определений этого понятия, приведем несколько примеров объектов, которые могут быть представлены в Simbix AP в виде узла: новость, статья, блог, запись в блоге, пользователь, объект недвижимости, товар, заказ, строка заказа, фотография, пункт в меню сайта. Эти узлы имеют определенное поведение, которое определяется как общими для всех узлов свойствами, так и свойствами, придуманными разработчиком сайта.

Общие для всех узлов в Simbix AP являют такие свойства, как идентификатор узла, его название, дата создания и проч. Для всех узлов в качестве их стандартного поведения в Simbix реализованы следующие операции: вкладывание узлов друг в друга в виде дерева, добавление, редактирование, удаление, перемещение и копирование узлов. Кроме того, узлы могут содержать произвольные ссылки друг на друга. Например, узел "Новость" может ссылаться на узел "Автор", узел "Строка заказа" явно должен содержать ссылку на товар, узел "Заказ" ссылается на узел "Покупатель" и т.п. Важно понимать, что Simbix AP изначально не содержит готовых объектов – их проектируют, разрабатывают и создают разработчики каждого конкретного проекта, указывая их свойства и ссылки между ними.

4.2 Модуль

Второй важный элемент модели данных Simbix – это модуль. При разработке проекта на платформе Simbix не следует вкладывать в понятие «модуль» понимание, характеризующее традиционные CMS. В Simbix.Интеграционный портал модуль не является заранее заданным программным блоком, представленным авторами системы. Модуль в Simbix очень похож по своему назначению на логический диск в операционной системе.

Известно, что опытные пользователи ОС часто разбивают свой физический жесткий диск на два и более раздела: на первом разделе (C:) они ставят операционную систему, а на всех остальных (D:, E:...) хранят свои данные. Делается это для того, чтобы разграничить файлы ОС и программ от данных пользователя, увеличивая, таким образом, отказоустойчивость и быстроту физического доступа к данным на диске.

Модуль в Simbix служит для очень похожей цели: он позволяет выделить в отдельную логическую группу определенные узлы. Таким образом, можно свалить все узлы в один модуль (перемешать данные пользователя и файлы ОС), но значительно удобнее их хранить разделив по каком-либо принципу. Например, удобно хранить в одном модуле все узлы, относящиеся к новостям: лента новостей, новость, журналист, rss-поток. Представим, что помимо новостей у вас на сайте есть еще и блог. Можно все узлы, относящиеся к блогу, тоже хранить в модуле "Новости", но удобнее сделать для них свой отдельный модуль, в котором

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. №	Подпись и дата
--------------	----------------	--------------	--------	----------------

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						15

будут аккуратно сложены дневники, записи, «друзья», комментарии. При этом, ничто не мешает разработчику создавать ссылки между узлами, находящимися в разным модулях, поэтому записи в блоге могут содержать ссылки на узлы типа "Новость".

4.3 Метод

Третий, очень важный элемент модели данных Simbix. Интеграционный портал – это метод. Спроектировав и создав узлы, разложив их по модулям, разработчик неизбежно сталкивается с вопросом: а как из этого всего сделать страницы на сайте? **Метод** в Simbix как раз и отвечает за вывод страницы. Для этого он производит выборку необходимых узлов и выводит их в определенном дизайне в виде HTML-документа. Кроме того, с помощью методов можно задавать свое поведение для узлов. В этом смысле методы можно сравнить с программами в ОС: например программа Windows Explorer показывает дерево с файлами и директориями, а программа Microsoft Word позволяет редактировать текстовые файлы специфичным для них методом.

Име. № подл.	Подпись и дата		Взам. инв. №	Инв. №	Подпись и дата		
Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33		Лист
							16

5 СОЗДАНИЕ ПЕРВОЙ СТРАНИЦЫ

Представим процесс создания первой страницы в виде следующей задачи: создать модуль с новостями и вывести одну новость на главной странице сайта. С целью учебного упрощения договоримся, что на главной странице выводится только заголовок, дата публикации и текст новости.

HTML-источник должен выглядеть следующим образом:

Листинг 1 :: Необходимый нам документ

```
1. <html>
2.   <head>
3.     <title>Ваша первая страница</title>
4.   </head>
5.   <body>
6.     <h1>Заголовок новости</h1>
7.     <div>Дата публикации новости</div>
8.     <div>
9.       Текст новости...
10.    </div>
11.  </body>
12. </html>
```

5.1 Создание модуля

Для реализации поставленной задачи, создадим модуль, в котором у нас будут храниться узлы новостей: один узел – одна новость. Следует подчеркнуть, что узлы всегда существуют в рамках модуля, не может быть узлов вне модуля.

Для того чтобы создать новый модуль:

- 1) в дереве проектов раскройте пункт дерева, обозначающий ваш проект.
- 2) щелкните правой кнопкой мыши и выберите в появившемся контекстном меню пункт "Создать модуль".
- 3) В появившейся справа форме заполните следующие поля:

Модуль – системное название модуля. Название должно соответствовать стандарту на системные названия Simbix. Интеграционный портал. Мы назовем наш модуль news.

Псевдоним – это свободное название модуля, более удобное для понимания человеком. В дальнейшем показывается в редакторском интерфейсе Simbix. В нем можно использовать любые символы. Псевдоним является необязательным свойством модуля. Если он не задан, в интерфейсе будет показываться системное название модуля. Мы дадим нашему модулю псевдоним "Новости".

Описание – произвольный текст, описывающий модуль. Это поле необязательно и служит исключительно для справки.

- 4) Нажмите на кнопку "Добавить".
- 5) Новый модуль появился в дереве проектов.

Име. № подл.	Подпись и дата
Взам. име. №	Име. №
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						17

Обратите внимание, что каждый проект имеет свой набор модулей.

5.2 Создание типа

Тип – это описание свойств, отличающих одни виды узлов от других. Например, созданные в рамках реализации поставленной задачи, узлы-новости относятся к типу "Новость", и этот тип содержит указание на то, что у любого узла, описываемого им, есть название, дата публикации и текст. Таким образом, когда мы говорим "узел типа "Новость"" мы имеем в виду такой узел, который описан типом "Новость".

В сравнении с традиционным подходом в объектно-ориентированном программировании (ООП), можно заметить, что связка тип – узел очень похожа на классическую связку класс – объект. Действительно, тип и узел соотносятся так же, как в классических объектно-ориентированных языках соотносятся класс и объект.

Тип также можно сравнить с образцом, а узел с конкретным предметом, изготавливаемым по этому образцу. Таким образом, становится очевидным, без описания типа узлов, создание узлов невозможно.

Для того, чтобы в модуль "Новости" можно было добавлять узлы-новости, необходимо определить типы узлов в данном модуле. В рамках реализации задачи, представим, что необходимо, чтобы у каждой новости был заголовок, дата публикации и текст.

Для создания типа:

1) Щелкнуть правой кнопкой на созданном в дереве проектов пункте "Новости";
2) В появившемся контекстном меню выбрать пункт "Создать тип". Здесь необходимо сделать небольшое теоретическое отступление:

3) В появившейся справа форме мы заполним следующие поля:

Название – системное название типа. Оно должно соответствовать стандарту на системные названия Simbix. Мы назовем наш тип "news_post".

Псевдоним – как и в случае с псевдонимом модуля, это удобное для пользователя название типа. Мы назначим нашему типу псевдоним "Новостное сообщение".

Остальные поля в секции "Описание" пропустим. В секции "Поля" мы видим три поля: "NodId", "NName" и "NFlag". Эти свойства обязательно присутствуют в любом типе, и в них соответственно хранятся уникальный идентификатор, название и флаг видимости узла. В отношении этих полей мы можем изменять только их псевдонимы. Для новости нам удобно использовать поле "Название" для хранения заголовка, возможно задать ему псевдоним "Заголовок".

Теперь добавим в тип остальные поля для узлов типа «новость».

Име. № подл.	Подпись и дата
Взам. име. №	Име. №
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						18

Для создания поля "Дата публикации" – нажать кнопку "Добавить поле" – в списке дополнительных свойств появится новая строка, в которой нам необходимо задать следующие характеристики поля:

Название – системное название поля, соответствующее стандарту на системные имена Simbix. Мы назовем наше поле "Date".

Псевдоним – удобное для пользователя название поля. В нашем случае мы дадим ему псевдоним "Дата публикации". Это поле необязательно для заполнения.

Тип – тип данных, которые будут храниться в этом поле.

Тип поля – это указание на свойства данных, для хранения которых предназначено поле. Эти свойства, в частности, определяют, как физически будут храниться данные поля в базе данных, как будет происходить сортировка и поиск по этому полю, как будет выглядеть это поле в интерфейсе редактора, в какой форме данные можно загружать в это поле (файл, текст, url и т.п.) и многое другое. В Simbix.Интеграционный портал имеется заранее заданный набор типов полей.

Для поля "Дата публикации" подходит два из имеющихся в списке типа поля: "date" и "datetime". Оба этих типа позволяют хранить данные о дате, разница между ними состоит лишь в том, что "date" хранит только день, месяц и год, в то время как "datetime" еще и час, минуту и секунду. В реализации нашей задачи следует выбрать тип "datetime", как более полный.

Остальные характеристики поля ("Параметры", "Значение по умолчанию" и "Обязательное поле") при реализации первого примера можно пропустить.

Аналогичным полем "Дата публикации" способом следует добавить поле "Текст":

Создать системное название "Text";

Назначить псевдоним "Текст" и тип "html";

Нажать кнопку "Сохранить".

В дереве проектов в модуле "Новости" появится только что созданный тип "Новостное сообщение". Теперь возможно добавление узлов типа "Новость".

Создадим первый узел:

1) в дереве проектов щелкнуть левой кнопкой мыши на модуле "Новости" – откроется пока еще не содержащий ни одного узла модуль "Новости".

2) чтобы добавить узел, щелкнуть правой кнопкой на корневом узле модуля и выбрать пункт "Добавить" – при этом справа откроется форма для добавления нового узла типа "Новость".

3) проверить, что в форме присутствуют обязательные для всех узлов полей ("Идентификатор", "Родитель", "Заголовок", "Видимость") и те поля, которые мы описали в типе: "Дата публикации" и "Текст".

Име. № подл.	Подпись и дата
Взам. инв. №	Име. №
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						19

4) установить у узла поле "Видимость" в состояние "узел виден" (в противном случае узел не будет показываться на сайте).

5) задать заголовок новости.

6) заполнить поля "Дата публикации" и "Текст".

Следует обратить внимание, что для поля "Дата публикации" система предлагает выбрать дату с помощью специального элемента управления "Календарь", а для поля текст в интерфейсе показывается удобный WYSIWYG-редактор. Такое поведение полей определяется их типом ("datetime" и "html" соответственно).

При нажатии кнопки "Добавить" в контекстном меню, видно, что в дереве модуля в корневом узле появился новый узел типа "Новость". В правой панели только что созданный узел доступен для редактирования. Следует обратить внимание, что в поле "Идентификатор" появился идентификатор. Это автоматически присваиваемый номер узла – по нему можно легко найти конкретный узел в модуле. Номер узла уникален в рамках одного модуля. Сохраните или запомните этот номер – он понадобится при реализации следующих шагов в решении задачи.

5.3 Вывод узла на странице. Создание метода

Создав модуль, тип и узел данного типа, можно приступить к основной части задачи – выводу созданного узла на главной странице сайта.

Для вывода объектов на страницах сайта служат методы – таким образом, чтобы создать страницу на сайте (часто, не одну страницу, а целую группу однотипных страниц), необходимо создать метод, который ее формирует. В Simbix.Интеграционный портал определено, что метод, который выводит главную страницу сайта, имеет название index и прикрепляется непосредственно к пункту проекта в дереве проектов.

Для создания метода:

1) в дереве проектов щелкнуть правой кнопкой на пункте, обозначающем ваш проект, и в появившемся контекстном меню выбрать действие "Методы".

2) в нижней панели откроется пустой список методов – нажать кнопку "Добавить". В открывшемся окне необходимо указать имя метода, которое должно соответствовать стандарту на системные названия платформы Simbix (в нашем случае это index).

3) указать в полях "Шаблон" и "Процессор" файлы на локальном диске. С помощью любого текстового редактора (например, Notepad) можно пустой файл с произвольным названием (например, index.htm) и выбрать его в поле "Шаблон". Для решения поставленной задачи процессор не понадобится, поэтому поле "Процессор" следует оставить пустым. Подробно о том, как соотносятся шаблоны и процессоре описывается в следующих параграфах.

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата
--------------	----------------	--------------	--------	----------------

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						20

4) Нажать кнопку "Добавить". В списке методов появится метод с названием "index".

Для того, чтобы написать код шаблона:

1) Щелкнуть на кнопку "Редактировать" напротив шаблона "index".

2) В открывшемся окне выбрать действие "Редактировать". Откроется окно для редактирования кода шаблона.

3) Для вывода новости на странице, наберем следующий код:

Листинг 2 :: Шаблон вывода новости

```
1. <html>
2.   <head>
3.     <title>Ваша первая страница</title>
4.   </head>
5.   <body>
6.     {s2GetNode mod="news" NodId=2 var="my_news_post"}
7.
8.     <h1>{$my_news_post.NName}</h1>
9.     <div>{$my_news_post.Date}</div>
10.    <div>
11.      {$my_news_post.Text}
12.    </div>
13.  </body>
14. </html>
```

В строке № 6 в параметре "NodId" вместо "2" необходимо указать номер объекта типа "Новость", который мы только что создали.

Разберем код представленного шаблона подробнее. Он представляет собой смесь обычного HTML и инструкций шаблонного языка Smarty. Подробнее о синтаксисе Smarty следует прочитать в документации Smarty, здесь же мы поясним лишь следующее:

- *Строка №6* {s2GetNode mod="news" NodId=2 var="my_news_post"}
Мы выбираем в смарти-переменную \$my_news_post узел из модуля "news" с идентификатором 2.
- *Строки №№8,9,11* {\$my_news_post.<...>}
Мы выводим содержимое полей "NName" ("Заголовок"), "Date" ("Дата публикации") и "Text" ("Текст"), обращаясь к ним по системным именам.

4) Сохранить шаблон метода и посмотреть результат на сайте.

Для того чтобы увидеть результат выполнения метода "index", нужно набрать в адресной строке браузера адрес разрабатываемого сайта (URL, указанный в поле "Домен проекта" при создании проекта.). На первой странице отобразятся заголовок, дату и текст созданной в рамках реализации задачи новости. Первая страница на Simbix.Интеграционный портал – создана.

6 МНОЖЕСТВЕННАЯ ВЫБОРКА

Как показывает практика разработки веб-проектов на платформе Simbix, при выборке данных для показа на странице разработчику чаще всего приходится иметь дело не с одним

Име. № подл.	Подпись и дата	Взам. инв. №	Име. №	Подпись и дата
--------------	----------------	--------------	--------	----------------

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						21

конкретным узлом, а с множеством. Чтобы усложнить и одновременно приблизить к реальности задачу, приведенную в предыдущей главе, выведем на главной странице сайта не одну конкретную новость, а пять последних новостей. При этом каждую новость мы представим только датой и заголовком. HTML-код страницы будет выглядеть примерно так:

Листинг 3

```

1. <html>
2.   <head>
3.     <title>Ваша первая страница</title>
4.   </head>
5.   <body>
6.     <h1>Заголовок новости 1</h1>
7.     <div>Дата новости 1</div>
8.
9.     <h1>Заголовок новости 2</h1>
10.    <div>Дата новости 1</div>
11.
12.    <h1>Заголовок новости 3</h1>
13.    <div>Дата новости 3</div>
14.    ...
15.  </body>
16. </html>

```

Для того чтобы приступить к выполнению поставленной задачи, сначала требуется создать необходимое количество узлов типа "Новость". Также следует обеспечить возможность убедиться, что разработанный код выбирает именно пять последних по дате новостей, поэтому для чистоты эксперимента добавим еще 5-6 новостей с разными значениями поля "Дата публикации". Добавив узлы, перейдем к коду шаблона:

Листинг 4

```

1. <html>
2.   <head>
3.     <title>Ваша первая страница</title>
4.   </head>
5.   <body>
6.     {s2GetNodes mod="news" Ntype="news_post" limit=5 sort="Date
7.       desc" var="my_news_posts"}
8.
9.     {foreach from=$my_news_posts item="my_news_post"}
10.      <h1>{$my_news_post.NName}</h1>
11.      <div>{$my_news_post.Date|date_format:"%d.%m.%Y"}</div>
12.    {/foreach}
13.  </body>
14. </html>

```

В строке № 6, в отличие от Листинга 2, мы выбираем теперь уже не конкретный узел, а все узлы типа "news_post", имеющиеся в модуле "news".

Функции s2GetNode и s2GetNodes, встретившиеся в Листингах 2 и 4 – функции из Simbix API. Simbix API (Application Programming Interface) – это список функций (команд), с помощью которых можно манипулировать узлами Simbix в коде методов. С помощью них можно производить с узлами все те же действия, которые с ними может производить оператор через интерфейс платформы. А именно: выбирать (искать), добавлять, изменять, удалять,

Име. № подл.	Подпись и дата
Взам. инв. №	Инв. №
Подпись и дата	Подпись и дата

копировать, перемещать узлы. Набор функций в Simbix API очень компактен, он включает всего 18 основных функций и 12 вспомогательных. Функции API описаны в документе «Simbix API». Основные 18 функций делятся на 3 группы:

1. Функции, возвращающие один узел.
2. Функции, возвращающие множество узлов.
3. Функции, изменяющие узлы.

Функция s2GetNode относится к первой группе, s2GetNodes ко второй. Каждая функция имеет набор параметров, с помощью которых разработчик уточняет выполняемое действие (выборку или изменение). Давайте подробнее рассмотрим вызов функции s2GetNodes в Листинге 4. В данном случае в функции указаны следующие параметры:

- **mod** – модуль, узлы которого необходимо искать. В нашем случае это модуль "Новости" ("news").
- **NType** – указание типа узлов, которые следует выбирать. В нашем случае нам нужны узлы типа "news_post". Если не указать данный параметр, функция будет производить поиск среди узлов всех типов, описанных в данном модуле.
- **limit** – количество узлов, которым следует ограничить результаты выборки. Важно понимать, что это максимальное число узлов в выборке. Таким образом, функция возвратит меньшее количество узлов, если условию выборки удовлетворило число узлов, меньшее limit. В нашем случае функция вернет не более 5 узлов.
- **sort** – сортировка результатов выборки. Данный параметр указывает, по какому полю или полям и в каком порядке (по возрастанию или по убыванию) следуют отсортировать узлы. Здесь можно указать несколько полей для сортировки. Например, если указать в параметре "sort" значение "Date desc, NName asc", то все узлы будут отсортированы по убыванию даты публикации, а те узлы, у которых даты публикации совпадают, разместятся в выборке в порядке возрастания заголовка.
- **var** – название переменной, в которую необходимо поместить результаты выборки. Этот параметр доступен только при вызове функции Simbix API внутри шаблона Smarty, поскольку является специфическим для данного шаблонизирующего языка. Забегая вперед, скажем, что при вызове функции Simbix API в PHP данный параметр не используется, поскольку в PHP команды Simbix являются обычными функциями, возвращающими результаты в переменную присвоения: \$a = s2GetNodes(...).

Это неполный список параметров функции s2GetNodes. Полный список всех параметров для всех функций Simbix API можно посмотреть в справочнике Simbix API.

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						23

Из перечисленных здесь параметров только "mod" является обязательным. Все остальные параметры могут быть опущены. Нетрудно догадаться, что если вызвать функцию s2GetNodes только с параметром mod="news", то она выберет все узлы из модуля "news" и вернет их в неотсортированном виде.

s2GetNode возвращает один узел. К его полям можно обращаться в шаблоне Smarty через точку по системному названию поля: \$a.NName. Функция же s2GetNodes возвращает, как мы уже говорили, набор узлов. Поэтому результат ее работы – это массив, каждый элемент которого это узел. Таким образом, чтобы вывести заголовок, например, третьего узла из массива \$my_news_posts, необходимо явно указать индекс узла в массиве: \$my_news_posts[2].NName. Нумерация элементов в массиве, естественно, начинается с 0. Однако разработчику редко приходится напрямую обращаться к определенному элементу из массива выбранных узлов. Гораздо чаще необходимо вывести подряд все выбранные узлы.

Для того чтобы вывести все выбранные узлы, в реализации задачи, используем конструкцию языка Smarty "foreach". Эта конструкция организует цикл по массиву, указанному в ее параметре "from", и помещает очередной элемент в переменную с названием, указанным в параметре "item". Таким образом, внутри этой конструкции можно обращаться к очередному узлу без индекса, через именованную переменную. Снова забежим вперед: в РНР набор узлов также возвращается как обычный массив со всеми вытекающими из этого приятными последствиями.

В строке №9 Листинга 4 представлена конструкция {\$my_news_post.Date|date_format:"%d.%m.%Y"}. "date_format" – это так называемый модификатор Smarty. Назначение модификатора – изменять внешний вид выводимого значения. В данном случае мы выводим дату новости не в формате "гггг-мм-дд чч:мм:сс", принятом по умолчанию, а в формате, традиционном для России – "дд.мм.гггг". Модификаторы – это очень мощный и удобный инструмент, подробнее о них можно прочитать в руководстве Smarty.

Таким образом, разобрав код Листинга 4., можно посмотреть, какие данные он выводит. В результате работы кода на главной странице вашего сайта появятся пять новостей из модуля "Новости", отсортированных по дате публикации.

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						24

7 ВТОРАЯ СТРАНИЦА

В результате реализации первой задачи, создана первая страница, на которой выводится пять самых свежих заголовков новостей. Сформулируем новую задачу: на главной странице заголовки новостей необходимо сделать ссылками, при нажатии на которые открывается страница с полным текстом новости. Задача предполагает создание второго метода, который будет отображать выбранную новость в виде заголовка, даты публикации и полного текста.

На первом этапе работа проходила с методом "index", прикрепленного непосредственно к проекту. Теперь необходимо создать второй метод. Перед тем, как это сделать, сделаем небольшое теоретическое отступление.

7.1 Выбор размещения метода

Метод в Simbix.Интеграционный портал может быть прикреплен к трем различным сущностям: непосредственно к проекту (именно так мы поступили с "index"), к любому из модулей, к любому из типов в модуле. В нашем случае помимо самого проекта, мы можем прикрепить новый метод к модулю "news" или к типу "news_post".

От чего зависит выбор варианта размещения, и на что он влияет? Зависит он от желания разработчика логически отделить одну группу методов от другой. Логично, если метод главной страницы сайта прикрепляется непосредственно к проекту, а метод, выводящий, например, новость, к объекту "Новость".

Общий критерий выбора варианта размещения метода можно сформулировать так: метод лучше прикреплять к той сущности, вывод подсущностей которой является основной задачей данного метода.

Другими словами, если основной задачей метода является вывод последних новостей, свежих статей, последних сообщений из блога, информации о проекте, то есть узлов из совершенно разных модулей, то этот метод лучше прикрепить к проекту, поскольку именно проект является той сущностью, для которой все выводимые узлы являются подсущностями. Если же необходимо вывести текст одной новости, то единственной сущностью, для которой конкретный узел-новость является подсущностью, это тип "Новость". От того, к какой сущности прикреплен метод, зависит URL, по которому его можно вызвать в браузере.

Если метод прикреплен непосредственно к проекту, то его URL имеет вид `http://имя домена/имя метода/`

Если метод прикреплен к модулю, то его URL имеет вид `http://имя домена/имя модуля/имя метода`

Если метод прикреплен к типу, то его URL имеет вид `http://имя домена/имя модуля/имя типа/имя метода/`

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата
--------------	----------------	--------------	--------	----------------

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						25

Таким образом, в рамках реализации поставленной задачи, целесообразно прикрепить новый метод для вывода страницы с текстом новости к типу "Новость". Назовем такой метод "show", поскольку он показывает узлы того типа, к которому он прикреплен. Название метода совершенно ни на что не влияет: разработчик волен выбирать удобные для него и для пользователя названия. Единственное требования к ним – соответствие стандарту на системные названия платформы Simbix.

7.2 Добавление метода к типу "Новость". Вывод данных.

Для добавления метода к типу «Новость»:

- 1) Щелкнуть правой кнопкой мыши на пункте "Новостное сообщение" в дереве проектов.
- 2) В появившемся контекстном меню выбрать действие "Методы".
- 3) Выполнить дальнейшие действия аналогично тем, которые выполнялись при добавлении метода "index".

Код метода:

Листинг 5

```

1. <html>
2.   <head>
3.     <title>Страница новости</title>
4.   </head>
5.   <body>
6.     {s2GetNode mod="news" NodId={$_THE.PARAMS[0]} var="my_news_post"}
7.
8.     <h1>{$my_news_post.NName}</h1>
9.     <div>{$my_news_post.Date}</div>
10.    <div>
11.      {$my_news_post.Text}
12.    </div>
13.  </body>
14. </html>

```

Этот код практически полностью повторяет Листинг 2, за исключением строки №6, где вместо жесткого указания идентификатора узла, мы подставили значение из переменной \$_THE.PARAMS[0]. Снова немного теории.

Разработанный метод отображает одну конкретную новость. Для того, чтобы указать методу, какую новость он должен вывести, воспользуемся передачей параметров в URL метода. Те, кто знаком с веб-разработкой, наверняка вспомнят о передаче параметров в URL после знака "?". Безусловно, можно воспользоваться этим удобным способом передачи параметров. Однако платформа Simbix предлагает и другой, альтернативный, способ. Дело в том, что адрес страницы типа <http://example.com/show.php?a=1&b=3&c=4> выглядит не очень привлекательно не только для человека, но и для поисковых систем. В платформе Simbix параметры можно передавать в конце URL, разделяя их точкой; после последнего параметра обязательно ставить ".htm". За счет этого URL выглядит гораздо лучше

Име. № подл.	Подпись и дата
Взам. име. №	Име. №
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист 26
------	------	----------	---------	------	---------------------------	------------

(<http://example.com/news/1.3.4.htm>). Параметры, переданные таким образом, доступны в шаблоне метода в переменной `$_THE.PARAMS` (в PHP это находящийся в глобальной видимости массив `$_THE['PARAMS']`). Эта переменная является массивом, в котором все переданные параметры нумеруются от 0.

В нашем методе мы предполагаем, что идентификатор узла-новости, который нужно вывести, передается первым параметром в URL: например, http://домен/news/news_post/show/2.htm. Важно понимать, что разработчик сам придумывает, какие параметры передавать и в каком порядке – Simbix лишь предоставляет удобную возможность их передачи непосредственно в URL.

Итак, метод показа новости готов, осталось изменить метод главной страницы так, чтобы заголовки статей стали ссылками на этот метод.

Листинг 6

```

1.  {foreach from=$my_news_posts item="my_news_post"}
2.      <h1>{$my_news_post.NName}</h1>
3.      <div>{$my_news_post.Date|date_format:"%d.%m.%Y"}</div>
4.      <a href="/news/news_post/show/{$my_news_post.NodId}.htm">
5.          читать всю новость
6.      </a>
7.  {/foreach}

```

Мы добавили ссылку на метод `show`, прикрепленный к типу "Новостное сообщение", с передачей идентификатора новости через первый параметр. Теперь можно проверить, как все это работает.

Име. № подл.	Подпись и дата	Взам. инв. №	Инв. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						27

8 ПРОЦЕССОР

Как было сказано выше, метод в Simbix.Интеграционный портал состоит из двух элементов: *шаблона* и *процессора*. До сих пор в реализации примеров были использованы исключительно шаблоны, теперь рассмотрим работу с процессором.

Если шаблон представляет собой смесь кода HTML с языком Smarty, то процессор - это обычный код на языке PHP. Важно помнить, что этот код выполняется *до* обработки и вывода шаблона, поэтому в нем можно готовить данные для дальнейшего вывода через шаблон. То, что процессор не является обязательной частью метода, мы уже успели убедиться - до сих пор мы создавали методы, состоящие только из шаблона. Однако и шаблон не является обязательной частью метода, он вполне может состоять исключительно из процессора. Таким образом, метод в Simbix может иметь следующую структуру:

- процессор;
- процессор + шаблон;
- шаблон.

Процессор требуется в тех случаях, когда для вывода данных на странице необходимо произвести операции, которые невозможно, сложно или неудобно сделать средствами языка Smarty. PHP – полноценный язык программирования с огромным количеством полезных функций. Его использование в процессорах делает практически безграничными возможности расширения функциональности ваших методов.

8.1 Создание RSS-трансляции. Задача

С целью понимания работы с процессорами решим следующую задачу: необходимо сделать RSS-трансляцию новостей из модуля "Новости".

1) Создать новый метод с именем "rss" и прикрепим его непосредственно к модулю "Новости".

Напомним, что в Главе 2 мы прикрепляли метод "index" непосредственно к проекту, в Главе 3 – к типу "Новостное сообщение"; теперь же мы реализуем третий и последний вариант размещения метода - мы прикрепим его к модулю.

2) Для прикрепления метода к модулю в дереве проектов на пункте "Новости" щелкнуть правой кнопкой мыши и в появившемся контекстном меню выбрать пункт "Методы".

3) В открывшемся пустом списке методов нажать на кнопку "Добавить".

4) В появившемся окне ввести в поле "Название" значение "rss".

Созданный метод будет состоять из процессора и шаблона, поэтому на этот раз нужно будет указать пустые файлы с диска и в поле "Процессор", и в поле "Шаблон".

Име. № подл.	Подпись и дата	Взам. име. №	Име. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						28

5) Сохранив метод, можно приступить к написанию процессора.

Поскольку мы создаем метод, который будет показывать данные в формате XML (RSS - это подмножество языка XML), а не HTML, нам необходимо отправить в браузер заголовок, который сообщил бы, что отсылаемые с сервера данные содержат XML. Для этого мы можем воспользоваться функцией PHP *header()* и написать в процессоре следующее:

Листинг 7

```
1. <?php
2.     header('Content-type: text/xml');
3. ?>
```

Теперь код шаблона:

Листинг 8

```
1. <?xml version="1.0" encoding="UTF-8"?>
2.
3. <rss version="2.0">
4.     <channel>
5.         <title>Новости сайта www.example.com</title>
6.         <link>http://www.example.com/</link>
7.         <description>Все самое интересное с www.example.com</description>
8.         <language>ru</language>
9.         <generator>SIMBIX</generator>
10.
11.         {s2GetNodes mod="news" Ntype="news_post" limit=5 sort="Date ¶
12.             desc" var="my_news_posts"}
13.         <item>
14.             <title>{$mp.NName}</title>
15.             <link>
16.                 http://www.ex.com/news/news_post/show/{$mp.NodId}.htm
17.             </link>
18.             <description>{$mp.Text}</description>
19.             <pubDate>{$mp.Date}</pubDate>
20.         </item>
21.     {/foreach}
22. </channel>
23. </rss>
```

В коде шаблона все остается таким же, как и при решении первой задаче, кроме, возможно, тэгов, использующихся в RSS. Подробнее об этом можно в открытых источниках, описывающих этот формат данных.

Если запустить наш метод в браузере (его URL будет таким *http://ваш домен/news/rss/*), ты мы увидим RSS-поток из пяти свежих новостей.

Ине. №	Ине. №	Взам. ине. №	Ине. №
Подпись и дата	Подпись и дата	Подпись и дата	Подпись и дата
Ине. № подл.			

9 ОРГАНИЗАЦИЯ ДЕРЕВА УЗЛОВ

Дерево – традиционная и удобная модель представления данных, которая встречается во многих в интерфейсах программ и сайтов. Файловая система, системный реестр Windows, меню в программах, каталоги товаров в интернет-магазинах – все это деревья.

Основным способом организации узлов в платформе Simbix также являются деревья. Узлы могут вкладываться друг в друга как файлы в директории. При этом непосредственно разработчик определяет, узлы каких типов допускают вложения, а каких нет.

Для того чтобы познакомиться с деревьями узлов в Simbix, решим следующую задачу. Сделаем так, чтобы на сайте редактор мог создавать произвольное количество новостных лент. На главной странице при этом будет выводиться по одному блоку от каждой ленты, в каждом блоке будет транслироваться пять последних новостей из соответствующей ленты.

9.1 Создание дерева. Лента новостей

Для создания ленты новостей, прежде всего, добавим в модуль "Новости" новый тип с названием "news_feed" с псевдонимом "Лента новостей". Для этого типа не нужны дополнительные поля, вполне достаточно поля "Название".

Теперь необходимо указать, что узлы типа "Новостное сообщение" должно вкладываться в узлы типа "Лента новостей":

1) щелкнуть мышью на только что созданный тип "Новостное сообщение". В открывшейся справа форме редактирования типа мы увидим секцию "Типы дочерних объектов".

В нем перечислены все имеющиеся в текущем модуле типы: в нашем случае это будут "Новостное сообщение" и "Новостная лента".

2) отметить флажок напротив "Новостного сообщения" и сохранить тип. Таким образом, системе указывается, что тип "Новостная лента" допускает вложение узлов типа "Новостное сообщение".

3) теперь необходимо создать несколько узлов типа "Новостная лента" и в один из них перенести уже имеющиеся у нас новости.

4) в дереве проектов щелкнуть левой кнопкой на пункте "Новости". В появившемся дереве модулей щелкнем правой кнопкой на корне дерева. На экране появится панель с выбором типа добавляемого узла.

Обратите внимание, что ранее, когда в модуле "Новости" у нас был всего один тип, эта панель не появлялась. Панель с выбором типа появляется только в том случае, если для данного узла возможно несколько дочерних узлов. Для корневого узла любого модуля возможными дочерними типами являются все описанные в модуле типы. В этом отличие корневого узла от всех остальных, для которых список типов в панели

Име. № подл.	Подпись и дата	Взам. инв. №	Инв. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						30

определяется тем, какие флажки были выставлены в форме описания типа в секции "Типы дочерних объектов".

5) Добавить в корень модуля новый узел типа "Новостная лента" с названием "Главная лента и перенести в него все созданные до этого узлы типа "Новостное сообщение":

6) Для этого выделить все узлы новостных сообщений (нажать клавишу "s", выделив нужный узел, или нажать пункт "Выделить узел" в его контекстном меню), а затем на узле "Главная лента" выбрать пункт "Перенести выделенные узлы" в его контекстном меню.

7) Теперь можно создать еще одну ленту, например, с названием "Новости партнеров".

8) Щелкнув правой кнопкой на только что созданном узле "Новости партнеров" и выбрав пункт "Добавить", можно убедиться, что система сразу открывает справа форму для добавления новостного сообщения – таким образом, установленная галочка, выставленная в секции "Типы дочерних объектов" в описании типа "Новостная лента", работает корректно.

Заметим, что в связи с тем, что мы указали в качестве возможных дочерних узлов только тип "Новостное сообщение", система не показала панель с выбором добавляемого типа.

Добавив несколько узлов типа "Новостное сообщение", можно приступить к модификации шаблона главной страницы. Ее код теперь будет выглядеть так:

Листинг 9

```
1. <html>
2.   <head>
3.     <title>Ваша первая страница</title>
4.   </head>
5.   <body>
6.     {s2GetNodes mod="news" NType="news_feed" var="news_feeds"}
7.
8.     {foreach from=$news_feeds item="news_feed"}
9.       <div>
10.        <h1>Новости из ленты "{$news_feed.NName}"</h1>
11.
12.        {s2GetChildren mod="news" NType="news_post" ¶
13.          ParId=$news_feed.NodId limit=5 sort="Date desc" ¶
14.          var="my_news_posts"}
15.
16.        {foreach from=$my_news_posts item="my_news_post"}
17.          <h2>{$my_news_post.NName}</h2>
18.          <div>
19.            {$my_news_post.Date|date_format:"%d.%m.%Y"}
20.          </div>
21.        {/foreach}
22.      </div>
23.    {/foreach}
24.  </body>
25. </html>
```

Конструкция в строке №6 уже описывалась в предыдущих параграфах и должна быть понятна: с помощью функции s2GetNodes мы возвращаем все имеющиеся в модуле "Новости"

Име. № подл.	Подпись и дата
Взам. инв. №	Име. №
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист 31
------	------	----------	---------	------	---------------------------	------------

10 РАБОТА СО ССЫЛКАМИ

Возможности по представлению сложных объектов реальности с помощью модели данных Simbix.Интеграционный портал были бы весьма ограничены, если бы они сводились исключительно к конструированию произвольных полей и построению деревьев узлов. Один из основных элементов модели данных в платформе Simbix.Интеграционный портал – это ссылки между узлами.

Ссылки реализованы как особый тип поля под названием `Ink`, который позволяет указать в качестве значения поля узел заранее заданного типа. Приведем пример.

У нас имеется два типа узлов: "Книга" и "Автор". Мы хотим сделать так, чтобы в узле "Книга" указывался бы ее автор (или авторы), и это указание представляло бы собой просто ссылку на объект типа "Автор". Для этого достаточно создать в типе "Книга" поле с произвольным названием, скажем, "Автор книги", и, указав для него тип "Ink", настроить параметры поля так, чтобы в нем можно было выбрать один или несколько узлов типа "Автор". Попробуем воспользоваться ссылками на практике.

Задача, в рамках которой мы рассмотрим реализацию ссылочных полей, будет следующей: необходимо создать в системе тип "Автор", который будет хранить информацию обо всех редакторах, пишущих новости на сайте. Кроме того, в узлах типа "Новостное сообщение" необходимо добавить поле для указания автора или авторов новости. На главной странице необходимо выводить ФИО автора рядом с заголовком.

Начнем с создания типа "Автор".

В поле "NName" мы будем хранить ФИО автора, остальные поля для нас сейчас не важны – вы можете добавить их по собственному усмотрению.

Например, вы можете хранить в создаваемом типе телефон, электронный адрес, номер ICQ и т.п. Теперь необходимо добавить новое поле "Авторы" в тип "Новостное сообщение".

- 1) В форме редактирования типа нажать на кнопку "+" под списком полей. Назвать поле "NewsAuthors" и задать псевдоним "Авторы новости".
- 2) В выпадающем списке типов поля выбрать тип "Ink".
- 3) Сохранить тип с помощью кнопки "Сохранить".
- 4) Далее необходимо указать дополнительные параметры поля. Для этого нажмем на кнопку "↑" справа от списка типов поля. В открывшемся окне мы увидим следующие поля и секции:
 - *Единственная ссылка* - если выставить этот флажок, то в поле можно будет указать только один узел. Поскольку мы хотим, чтобы у новости можно было указать несколько автором, мы оставим этот флажок неотмеченным.

Име. № подл.	Подпись и дата	Взам. ине. №	Име. №	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						33

- *Показывать в виде списка* - если отметить этот флажок, то выбор объекта в форме редактирования узла будет выглядеть как выпадающий список. При небольшом количестве узлов этот способ выбора очень удобен, поскольку требует всего два клика. Однако он становится неэффективным при большом количестве узлов. В этом случае удобнее использовать форму подбора узлов. Мы оставим этот флажок неотмеченным.
- *Типы прикрепляемых узлов* - это список все типов, имеющихся в проекте. Отмечая галочками различные типы, вы сообщаете системе, узлы какого типа можно будет указывать в данном поле. Мы отметим только созданный тип "Автор".

Сохранив тип, перейдем к созданию тестовых узлов.

1) В модуле "Новости" добавить несколько узлов типа "Автор". Пока их можно разместить непосредственно в корне модуля, рядом с объектами типа "Новостная лента".

2) Указать авторов для нескольких произвольных новостей. Для этого в форме редактирования новости в поле "Авторы" нажать на кнопку "+". В открывшемся окне подбора узлов можно выбрать узел из списка всех узлов – для этого нужно нажать на кнопку "Найти" или найти нужные узлы по полю.

Если вы хотите найти, например, узел, у которого в поле "ФИО" указано "Толстой", то нажмите на вкладку "Условия", а затем на кнопку "Добавить". В появившемся списке полей, выберите "ФИО". В области построения условия вы увидите следующее выражение "ФИО = ". Щелкните на значке и введите в появившемся поле "Толстой". После нажатия на кнопку "Найти" во вкладке "Результат" появится список узлов, в которых поле ФИО содержит слово "Толстой".

Обратите внимание, что в поле "Авторы" вы можете указать несколько узлов типа "Автор".

Закончив с созданием тестовых объектов, перейдем к модификации шаблона первой страницы.

Листинг 10

```

1. <html>
2.   <head>
3.     <title>Ваша первая страница</title>
4.   </head>
5.   <body>
6.     {s2GetNodes mod="news" NType="news_feed" var="news_feeds"}
7.
8.     {foreach from=$news_feeds item="news_feed"}
9.       <div>
10.        <h1>Новости из ленты "{$news_feed.NName}"</h1>
11.
12.        {s2GetChildren mod="news" NType="news_post" ¶
            ParId=$news_feed.NodId limit=5 sort="Date desc" ¶
            var="my_news_posts"}

```

Ине. № подл.	Подпись и дата
Взам. ине. №	Ине. №
Подпись и дата	Подпись и дата

```

13.
14.         {foreach from=$my_news_posts item="my_news_post"}
15.         <h2>{$my_news_post.NName}</h2>
16.         {foreach name="authors" from=$my_news_post.Authors ¶
           item="author"}
17.             {$author.NName}{if !$smarty.foreach.authors.last}, {/if}
18.         {/foreach}
19.         <div>
20.             {$my_news_post.Date|date_format:"%d.%m.%Y"}
21.         </div>
22.         {/foreach}
23.     </div>
24. {/foreach}
25. </body>
26. </html>

```

Строки №№ 14-16 выводят список авторов новости. Несложно догадаться, что поле "Author" типа "lnk" представляет собой обычный массив, в котором хранятся все те узлы, которые контент-менеджер указал в этом поле в интерфейсе. Таким образом, чтобы вывести всех авторов, мы просто организовали цикл по этом массиву (\$my_news_post.Authors). На каждой итерации цикла для каждого автора мы выводим его ФИО, следом за которыми ставим запятую, если текущая итерация цикла не последняя (более подробно о конструкции проверки последней итерации см. описание *foreach* в документации Smarty). У разработчика может возникнуть справедливый вопрос: а как будет выглядеть поле "Authors" в узле, если в нем допускается указание всего одного узла (т.е. у поля в параметрах отмечена галочка "Единственная ссылка")? Поле будет выглядеть точно также - это будет массив. Просто у него будет всего один элемент с индексом 0. Если бы у новости мы допускали бы указание только одного автора, то код строк №№ 14-16 Листинга 10 выглядел бы вот так:

Листинг 11

```

1. ...
2. {foreach from=$my_news_posts item="my_news_post"}
3. <h2>{$my_news_post.NName}</h2>
4. {$my_news_post.Author[0].NName}
5. <div>{$my_news_post.Date|date_format:"%d.%m.%Y"}</div>
6. {/foreach}
7. ...

```

В строке №4 мы вывели ФИО первого автора, хранящегося в поле "Author" текущей новости. Таким образом, разработчику важно помнить, что независимо от того, сколько узлов можно указывать в поле типа "lnk", в шаблоне и процессоре узел содержит в соответствующем поле массив минимум из одного элемента. Часто встречается ошибка, когда доступ к полям связанного узла пытаются получить вот так – {\$my_news_post.Author.NName}, когда как правильное обращение выглядит следующим образом – {\$my_news_post.Author[0].NName}.

Открыв первую страницу сайта в браузере, вы должны увидеть, что для тех новостей, у которых указаны авторы, они выводятся под заголовком через запятую.

Име. № подл.	Подпись и дата	Име. №	Подпись и дата
Взам. име. №			

Изм.	Лист	№ докум.	Подпись	Дата	SIMBIX.00003-01 33	Лист
						35

