

Пример 1. Hello World. Четыре простых шага

Вольный перевод <http://www.getsymphony.com/learn/tutorials/view/say-hello-to-the-world/>

Пояснения:

1. Символ нижнего подчёркивания используется для связи слов в названиях и в их переводах;
2. Идентичный – полностью совпадающий по всем параметрам (клон, дубль, копия);
3. Аналогичный – сходный по нескольким параметрам, но не всем, то есть не идентичный;
4. В зависимости от версии Symphony, наборы секций, полей и их месторасположение могут меняться, но общий смысл остаётся. В нашем случае используется версия 2.4.

Symphony CMS (не путать с Symfony PHP FrameWork, который для тех из Вас, кто решил сам написать админку для сайта) предназначена для оперирования данными (получение, хранение, обработка) и передачи их в удобоваримом виде посетителю сайта. То, что видит этот самый посетитель называется front end (фронт энд, web-страница, интерфейс пользователя); администратор/модератор/контент-менеджер, перейдя по специальной ссылке и авторизовавшись, видит перед собой back end (бэк энд, админку, административную зону, интерфейс администратора). Задача разработчика состоит в программировании передачи данных из админки (а иногда и не только) на разрабатываемую web-страницу. Язык программирования Symphony - XSLT: универсален, прост, позволяет на-GOOGLE-ить огромное количество примеров, help-ов, он-лайн проверок ошибок, форумов и имеет чётко задокументированный стандарт:

http://www.w3schools.com/xsl/xsl_w3celementref.asp

Предполагается, что у Вас установлена работоспособная Symphony (пустая: не включающая workspace (рабочую_область) по умолчанию), Вы залогинились (авторизовались, ввели логин и пароль в поля Username и Password) и зашли в админку. Если же нет, проведите [install](#) (установку).

Шаг 1. Контент

Начинаем с Blueprints » Sections (проекты » разделы) для определения базовых (основных, корневых) типов контента: вначале у пустой Symphony здесь наблюдается пустой список (рис.1).

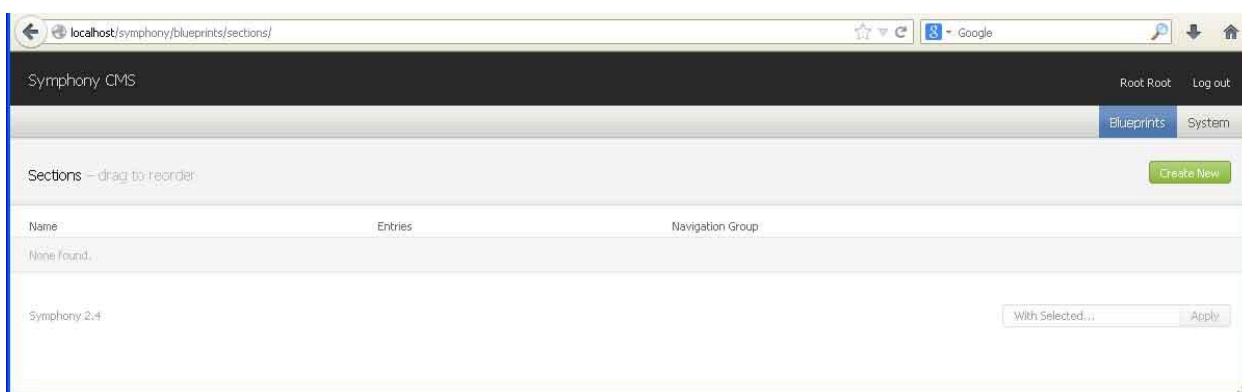


рис.1

Нажав зелёную кнопку **Create_New (создать_новый)** вверху справа, откроется редактор разделов (рис.2).

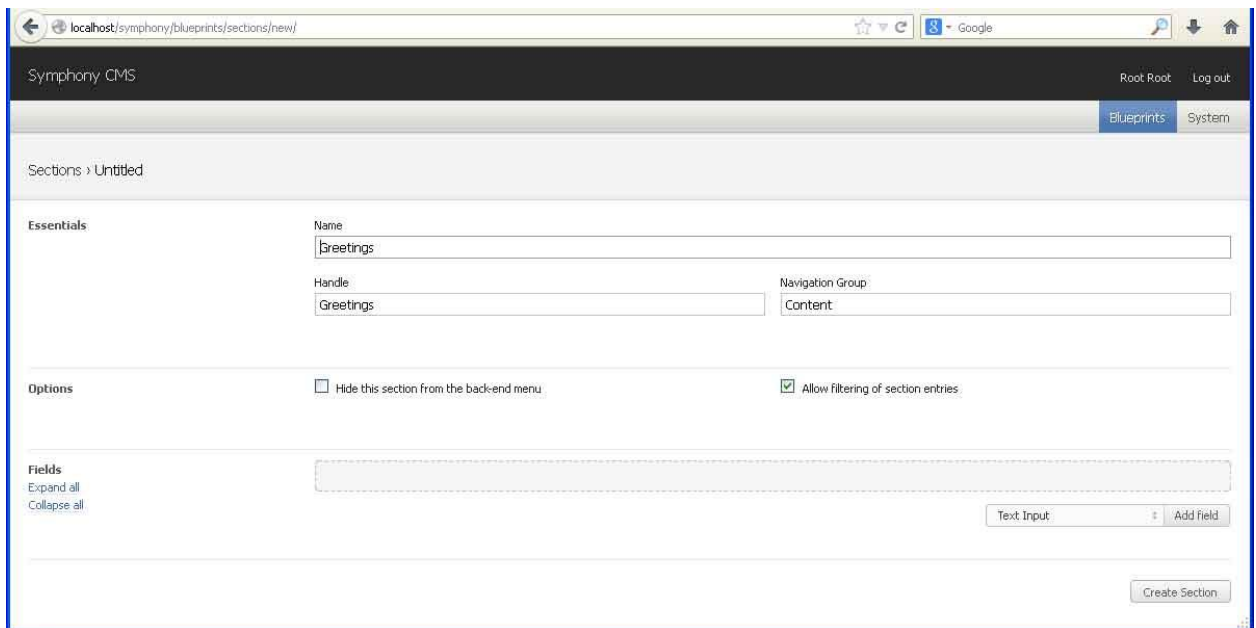


рис.2

Верхняя секция Essentials (Основы) включает поле Name (имя: название базового типа: пусть будет Greetings), Handle (ссылка: т.н. валидное (правильно написанное) слово для ссылки: пусть будет Greetings) и Navigation_Group (навигационная_группа: пункт в верхнем меню админки, в который будут группироваться Поля: слово Content вполне подойдёт).

Ниже находится секция Fields (поля), у нового раздела список Полей пуст. Нажав выпадающее меню слева от кнопки Add_field (добавить_поле), мы увидим список системных типов полей:

- **Author (автор)** – обозначает пользователя, у которого задаются значения:
 - Label (наименование, метка) – У ВСЕХ базовых типов присутствует идентичный пункт; в описании у каждого базового типа ниже повторяться не будет
 - Handle (ссылка) – см.примечание к Author » Label
 - Placement (расположение) в редакторе Раздела – см.примечание к Author » Label:
 - Sidebar (боковая_панель)
 - Main_content (основная_область)
 - Author_types (типы_авторов):
 - Author (простой автор)
 - Manager (крутой автор, контент-менеджер, модератор)
 - Developer (разработчик, админ, root, программист)
 - Allow_selection_of_multiple_authors (разрешить_выделять_нескольких_авторов) – аналогичный пункт имеется у всех базовых типов - списков
 - Select_current_user_by_default (текущий_пользователь_будет_по_умолчанию) – аналогичный пункт присутствует почти у всех базовых типов
 - Make_this_a_required_field (сделать_поле_обязательным_для_заполнения) – см.примечание к Author » Label
 - Display_in_entries_table (значения_выводить_в_табличном_виде) – см.примечание к Author » Label
- **Checkbox (флажок)** – обозначает булеву переменную (которая может принимать значения 0|1, true|false, вкл|выкл) со значением:
 - Checked_by_default (включено_по_умолчанию)
- **Date (дата)** со значением:

- Default_date (дата_по_умолчанию): стоит now (текущая)
- **File_Upload (загружаемый_файл)** со значениями:
 - Destination_Directory (папка_расположения_файла)
 - Validation_Rule (правило_валидации) – маска проверки названия файла: щёлкнув по слову image или document под строкой ввода, устанавливается значение маски проверки для графического файла или документа
- **Select_Box (выпадающий_список)** со значениями:
 - Static_Values (статичные_значения) – перечисление значений
 - Dynamic_Values (динамические_значения) – берутся из структуры Symphony
 - Allow_selection_of_multiple_options (разрешить_выделять_несколько_значений)
 - Sort_all_options_alphabetically (сортировать_по_алфавиту)
 - Display_relationship_in_entries_table (показать_фильтр_в_списке_Разделов) – отображать название этого выпадающего списка в строке соответствующего Раздела в списке Разделов: крайне удобно для фильтрации: количество Разделов с таким же значением сразу отображается
- **Select_Box_Link (выпадающий_список_ссылки)** со значениями:
 - Values (значения)
 - Maximum_entries (наибольшее_количество)
 - Allow_selection_of_multiple_options – *идентичен Select_Box » Allow_selection_of_multiple_options см.выше*
 - Hide_when_prepopulated (свернуть_после_добавления)
 - Display_relationship_in_entries_table – *идентичен Select_Box » Display_relationship_in_entries_table см.выше, но (!) в фильтре находятся не выбранные значения из выпадающего списка Поля, а ссылки на соответствующие Поля соответствующего базового типа*
- **Tag_List (список_тэгов)** со значениями:
 - Suggestion_List (список_предложений)
 - Validation_Rule (правило_валидации) – маска проверки: щёлкнув по слову number, email или URI под строкой ввода, устанавливается значение маски проверки для числовых значений, E-mail адресов или URL
- **Textarea (текстовое_поле)** со значениями:
 - Number_of_default_rows (количество_строк_по_умолчанию)
 - Text_Formatter (метод_форматирования_текста) – выбор набора спецсимволов для кодирования html-тэгов
 - None (отсутствует)
 - Markdown
 - Markdown_Extra
 - Markdown_Extra_(With_SmartyPants)
 - Markdown_(With_HTML_Purifier)
- **Text_Input (вводимый_текст)** со значением:
 - Validation_Rule (правило_валидации) – *идентичен Tag_List » Validation_Rule см.выше*

Нажав на кнопку Add_field (добавить_поле), Поле с выбранным базовым типом добавится в список и станут доступны для редактирования соответствующие значения. При щелчке на строке с названием Поля, область со значениями сворачивается, а при щелчке на строке с названием свёрнутого Поля, соответственно, разворачивается. При щелчке на надписи Remove_field (удалить_поле) в этой же строке справа, Поле может быть удалено.

Для нашего примера добавим Поле Greeting_Text (текст_поздравления) типа Text_Input, установив флажок обязательно_для_заполнения (результат представлен на рис.3) и нажмём кнопку Create_Section (создать_раздел):

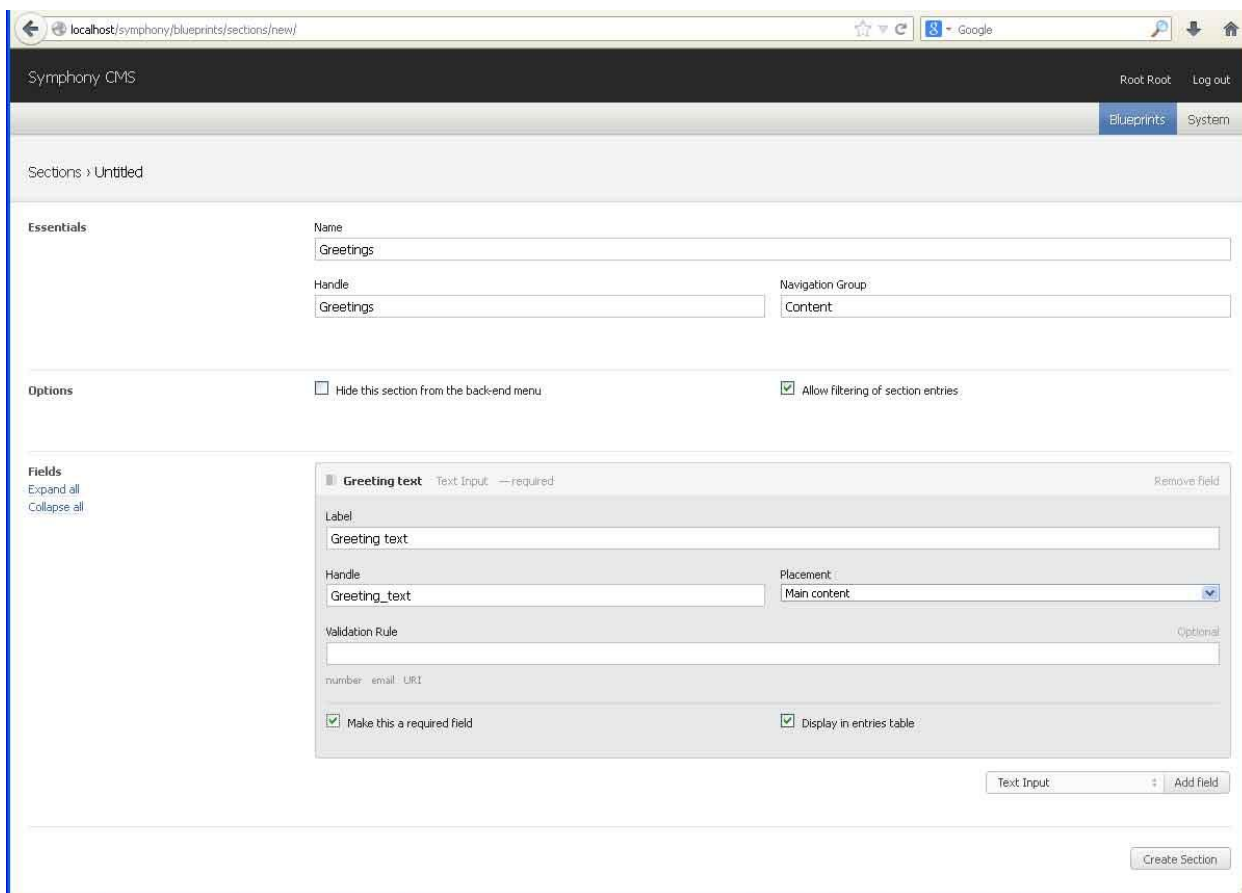


рис.3

После некоторых операций в самом верху страницы возможно появление строки, которая вначале имеет яркий зелёный цвет, а через некоторое время меняет его на чёрный (рис.4 и рис.5). Строка содержит отчёт о проведённой операции, в нашем примере это: Section_created_just_now (только_что_создан_раздел) или Section_created_13_minutes_ago (раздел_создан_13_минут_назад). А также предложение выбора следующего действия, в нашем примере это: Create_another? (создать_следующий?) и View_all_Sections (перейти_к_списку_разделов). Перейдём к списку разделов.

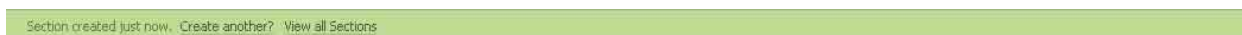


рис.4



рис.5

В верхнем меню админки появится пункт Content, соответствующий имени навигационной_группы, а в нём будет присутствовать подпункт Greetings, соответствующий названию нашего базового типа. Щелкнем же по нему, открыв при этом совершенно пустой список Экземпляров этого базового типа (рис.6).



рис.6

Нажав зелёную кнопку **Create_New (создать_новый)** вверху справа, откроется редактор Экземпляра. Введя **Hello_World!** в строку ввода **Greeting_text** (рис.7) и нажмём **Create_Entry** (создать_экземпляр). А также добавим новый Экземпляр со строкой **G'day_World! (привет, мир! по-австралийски)**.



рис.7

Шаг 2. Создание фронт энда

Вначале проект Symphony не имеет фронт энда, то есть при заходе на сайт Вы увидите лишь сообщение об ошибке **Page_Not_Found** (страница_не_найдена), исправим это! В верхнем меню перейдём по **Blueprints » Pages (проекты » страницы)** и увидим пустой список Страниц (рис.8).

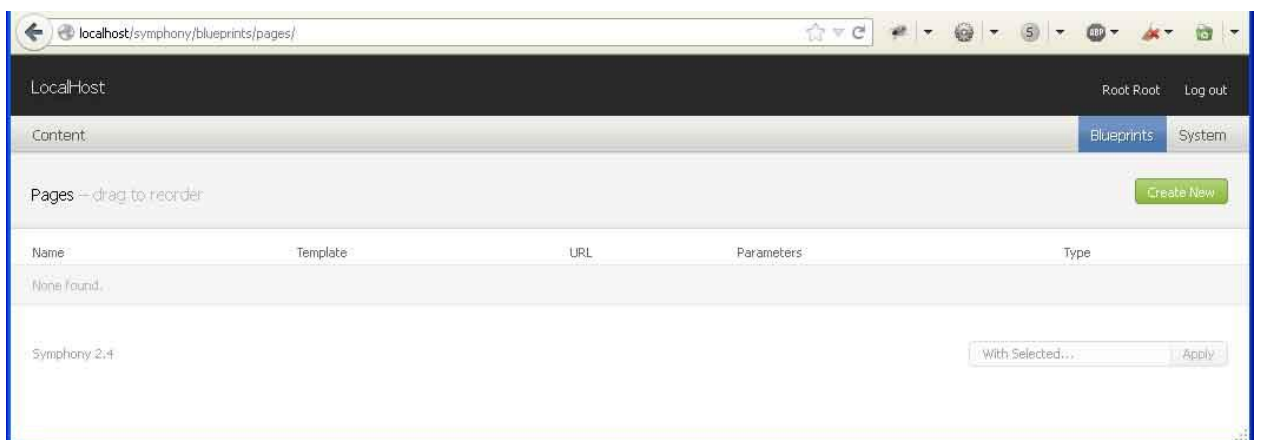


рис.8

Нажав зелёную кнопку **Create_New (создать_новый)** вверху справа, откроется редактор Страницы (рис.9).

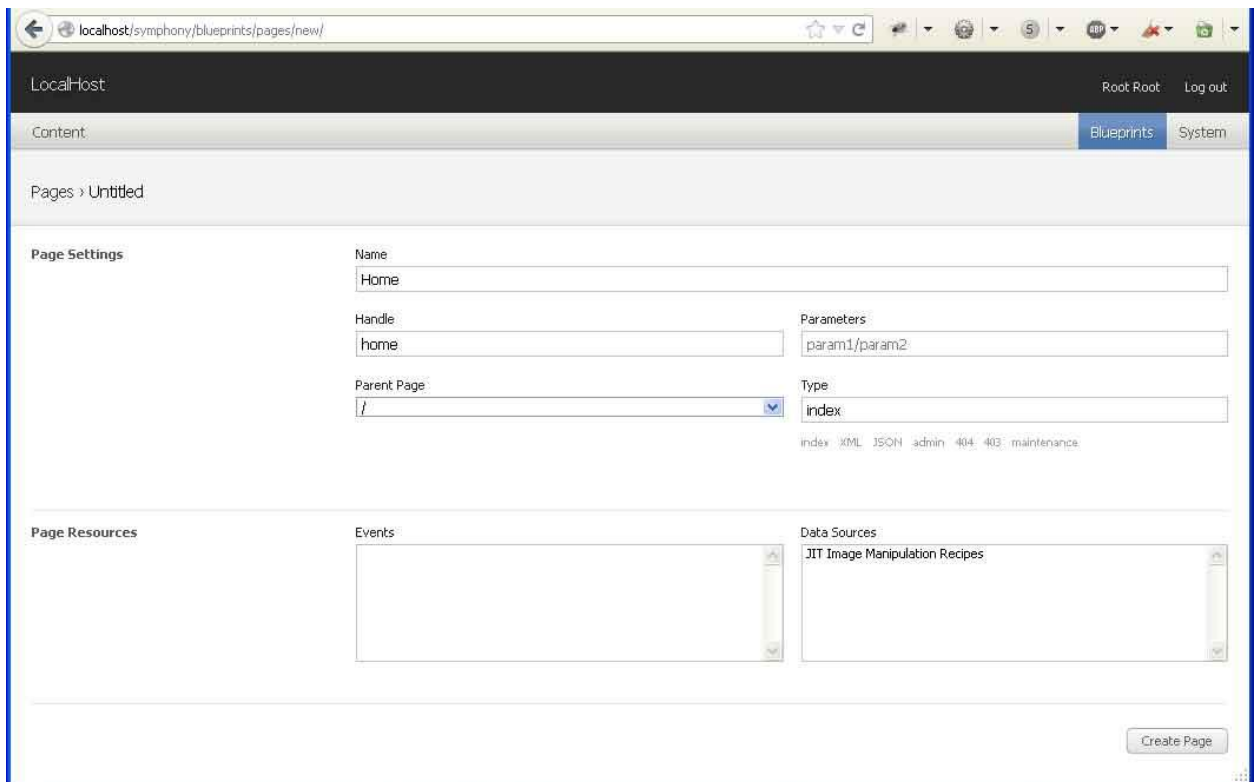


рис.9

Введём слово Home (домашняя, главная) в поле Title (заголовок), слово home в поле URL_Handle (URL-адрес), поле URL_parameters (URL-параметры, URL-суффиксы, которые вводятся в адресной строки после символа ? в конце адреса) оставим пустое. Поле Parent_Page (родительская страница, страница более высокого уровня) оставим как есть: символ слеша обозначает, что страниц выше уровнем на сайте нет. Далее поле Type (тип), под которым присутствуют предустановленные значения (и на которые можно щёлкнуть для подстановки в само Поле):

- index (титульная) – главная страница сайта;
- XML – возвращает страницу с типом text/xml, вместо text/html;
- JSON – возвращает страницу в формате JSON;
- admin – страница, доступная только авторизованным пользователям;
- 404 – страница ошибки «страница не найдена»;
- 403 – страница ошибки «доступ запрещён»;
- maintenance (тех.обслуживание) – страница «страница находится в разработке» (включается в конфигурационном файле).

Для нашего примера выберем index и нажмём кнопку Create_Page (создать_страницу), результатом чего будет появление новой страницы Home в списке страниц. Теперь при заходе на сайт, мы увидим слово Home (по крайней мере так будет в браузере Firefox), что означает, что страница Home стала доступна посетителям, но слова приветствий, ранее нами добавленные, не привязаны к этой странице и шаблон её совершенно пуст (за исключением слова Home в тэгах <h1></h1>), то есть отображаемая страница даже не является валидным html-документом.

Шаг 3. Получение данных

Data_Sources (источники_данных) производят выборки данных и передают их страницам. Для примера нам нужен такой источник, который передаст слова приветствия на фронт энд. Соответственно, переходим в **Blueprints** » **Data_Sources** (проекты » источники_данных) и жмём

кнопку зелёную кнопку **Create_New (создать_новый)** вверху справа. После произведённых манипуляций откроется редактор источников данных (рис.10).

localhost/symphony/blueprints/datasources/new/

LocalHost Root Root Log out

Content Blueprints System

Data Sources > Untitled Source Greetings

Essentials Name

Conditions Leaving these fields empty will always execute the data source.
Required Parameter Optional Forbidden Parameter Optional
 Redirect to 404 page when no results are found

Filters Use { \$param }-syntax to filter by page parameters.
 Add filter

Sorting Use { \$param }-syntax to order by page parameters.
Sort By
Sort Order
asc desc random

Grouping Group By

Pagination Use { \$param }-syntax to limit by page parameters.
Entries per Page Page Number
 Enable pagination

Content
Included Elements
system: pagination
system: date
Greeting_Text
Parameters
\$ds-untitled.system-id
\$ds-untitled.system-creation-date
\$ds-untitled.system-modification-date
\$ds-untitled.system-author
\$ds-untitled.Greeting_Text
 Include a count of entries in associated sections
 HTML-encode text

Attach to Pages The data will only be available on the selected pages.

Create Data Source

рис.10

В выпадающем списке Source (источник) вверху страницы справа мы наблюдаем предустановленное значение Greetings, которое соответствует добавленному нами ранее разделу. Для нашего примера введём название источника данных: слово Greetings в поле Name. И оставим значения Filters (фильтры), Sorting (сортировка), Grouping (группировка), Pagination (разбивка_на_страницы) как есть. В секции Content (содержимое) в окне Included_Elements

(включаемые_элементы) выделим строку `Greeting_Text`, соответствующую полю с этим же именем у добавленного нами ранее раздела `Greetings`; также «включим» флажок `Include_a_count_of_entries_in_associated_sections` (включить количество полученных результатов в выдачу). В окне `Attach_to_Pages` (привязка_к_страницам) выберем страницу `Home`: единственную ранее нами добавленную (что одновременно ведёт к выделению строки `Greetings` в редакторе страницы `Home` в `Blueprints` » `Pages` » `Home`). И нажмём кнопку `Create_Data_Source` (создать_источник_данных).

Для изменения внешнего вида фронт энда используем встроенный (!) в `Symphony` дебаггер: перейдя на фронт энд, добавим к строке адреса суффикс `?debug` и, перейдя по новому адресу, увидим интерфейс (рис.11), из которого почерпнем данные, генерируемые `Symphony`. Также воспользуемся Вашим любимым текстовым редактором, в котором Вы и верстаете свои сайты: шаблоны страниц в `xsl`-файлах лежат в папке `workspace\pages`.

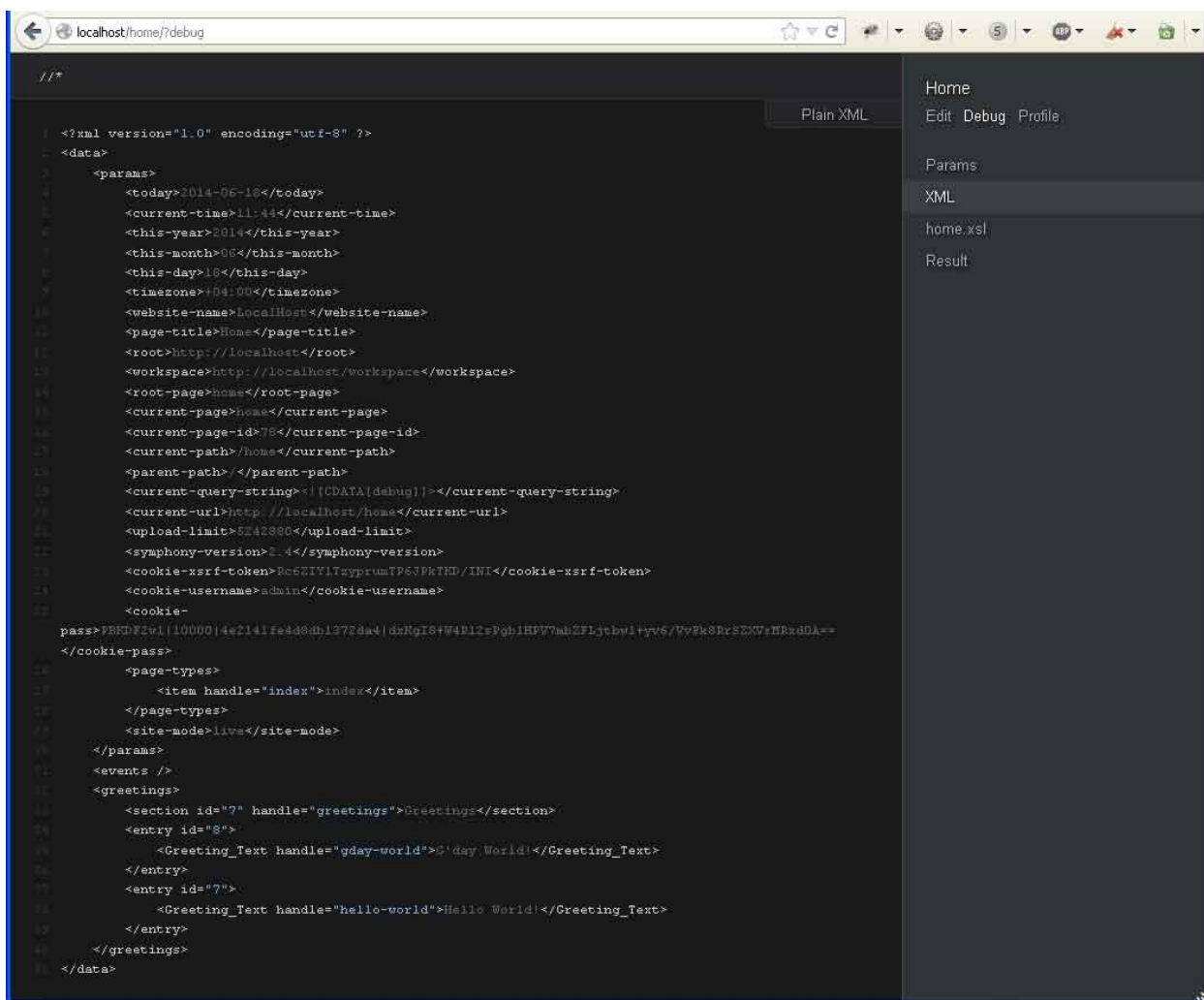


рис.11

Шаг 4. Редактирование шаблона

Шаблоны – это текстовые файлы с `XSLT`-инструкциями для преобразования `XML`-данных в текстовые `XHTML`-файлы, которые и передаются в клиентский браузер. Вначале файл `home.xsl` шаблона страницы `Home` фактически пуст:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```



```

<xsl:output method="xml" doctype-public="-//W3C//DTD XHTML 1.0 Strict//EN" doctype-
system=http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd omit-xml-declaration="yes" encoding="UTF-8"
indent="yes" />
<xsl:template match="/">
    <h1><xsl:value-of select="$page-title"/></h1>
</xsl:template>
</xsl:stylesheet>

```

Изменим корневой шаблон, выводящий лишь основной заголовок:

```

<xsl:template match="/">
    <h1><xsl:value-of select="$page-title"/></h1>
</xsl:template>

```

Сформировав валидную html-структуру, используя xsl-параметр \$page-title и выведя другой заголовок (доктайпы, метатеги, стили, скрипты и т.д. пока опустим, чтобы не загромождать наш пример):

```

<xsl:template match="/">
    <html>
        <head>
            <title><xsl:value-of select="$page-title"/></title>
        </head>
        <body>
            <h1>Symphony Greets the World</h1>
        </body>
    </html>
</xsl:template>

```

Теперь, используя в дебаггере в самой верхней строке (где стоит //*) вводим строку /data/greetings/entry и видим, что ярким зелёным цветом помечаются узлы, внутри которых находятся введённые нами приветствия на разных языках, которые мы и хотели бы вывести на web-странице. Изменяем файл home.xsl, дописав обработку этих нескольких узлов и вывода из них данных на фронт энд в виде нумерованного списка (используя инструкции XPath):

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" doctype-public="-//W3C//DTD XHTML 1.0 Strict//EN" doctype-
system=http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd omit-xml-declaration="yes" encoding="UTF-8"
indent="yes" />
<xsl:template match="/">
    <html>
        <head>
            <title><xsl:value-of select="$page-title"/></title>
        </head>
        <body>
            <h1>Symphony Greets the World</h1>
            <h2><xsl:value-of select="/data/greetings/section"/></h2>
            <ul>
                <xsl:apply-templates select="/data/greetings/entry"/>
            </ul>
        </body>
    </html>
</xsl:template>
<xsl:template match="greetings/entry">

```

```
<li><xsl:value-of select="Greeting_Text"/></li>
</xsl:template>
</xsl:stylesheet>
```

Результат на фронт энде появится после сохранения файла home.xml и перезагрузки страницы:

Symphony Greetс the World

Greetings

- G'day World!
- Hello World!

Соответственно, добавив ещё одну страницу через админку (Blueprints » Pages) со словами приветствия Hallo_Welt! (это по-немецки) в поле Greeting_Text, получим такую страницу:

Symphony Greetс the World

Greetings

- Hallo Welt!
- G'day World!
- Hello World!

Ну, вот и всё! В следующем примере будет получение вводимых пользователем данных и вывод их на фронт энде.