

Техническое задание на разработку веб-сайта

Краткое описание

Разрабатываемый веб-сайт (далее - сайт) предназначен для размещения учебных материалов (далее - уроков) и упражнений к ним (далее - заданий).

Исходники уроков и заданий - текстовые файлы в формате LaTeX + изображения.

Используемые средства, системные требования

Движок сайта должен быть выполнен на фреймворке Ruby on Rails. База данных - PostgreSQL.

Сайт будет размещаться на VPS с Ubuntu 14.04, при необходимости исполнителю будет предоставлен доступ к нему.

Обновление контента

Следующие шаги носят временный характер. Изменения, произведенные в них, применяются только в случае отсутствия ошибок при их выполнении (см. пункт “Закрепление изменений”). Обо всем процессе обновления контента ведется лог.

Обработка исходных файлов

В процессе добавления материалов на сайт происходит “прекомпиляция” материалов. Каждый файл в формате LaTeX конвертируется в файл представления следующим образом: Вначале каждый файл конвертируется в следующую древовидную структуру

```
A:   { type: text,  
      , content: String  
    }  
B:   { type: command,  
      , name: String  
      , args: D  
    }  
C:   { type: environment,  
      , name: String  
      , content: D  
    }  
D:   Array[A|B|C]
```

root = D

Некоторые команды при их обработке обрабатываются дополнительно:

1. В общем случае для обработки каждой команды запускается одна и та же функция, из которой поток обработки переходит затем в специализированные функции в зависимости от имени команды.
2. В специальную таблицу базы данных заносится (обновляется) информация о данном файле:
 - a. Название главы - из команды `\subsection{...}`
 - b. Название главы латиницей - из команды `\aliasname{...}`
 - c. Случайная последовательность букв и цифр (далее - "хеш-ссылка на страницу") (UNIQUE)
 - d. Дата/время обновления (добавления) записи о данном файле (время компиляции)
3. Для каждой команды `\label{labelname}` (а так же для некоторых других блоков) вызывается функция, заносщая в специальную таблицу базы данных следующую информацию:
 - a. хеш-ссылка на соответствующую страницу
 - b. `labelname` (UNIQUE)
 - c. Object number (например, в случае `\begin{equation}` ссылка на него должна включать в себя номер объекта)

Дальнейшая конвертация

Древовидная структура рекурсивно конвертируется в текстовый файл. Для этого должна использоваться единая функция "обработать блок", которая в зависимости от типа блока вызывает другие функции.

Отдельного внимания заслуживают функции обработки уравнений. Отдельные типы уравнений ($\$. \$$, $\$ \$$, $\[...\]$) конвертируются просто в текст (для дальнейшего отображения "вживую" с помощью `mathjax`). Другие же типы (например, `\begin{equation}`), конвертируются в изображения (см. пункт "компиляция формул").

Компиляция формул

Для компиляции формул используется компилятор TeX (см.

<http://tex.stackexchange.com/questions/11866/compile-a-latex-document-into-a-png-image-tats-as-short-as-possible>)

Таблицы

На данном этапе реализация парсинга таблиц не требуется.

“Закрепление” изменений

В случае успешной компиляции (успешность определяется администратором по логам) обновления закрепляются. Подлежит обсуждению с исполнителем.

Структура исходных файлов

Исходником отдельно взятого урока или задания является файл в формате LaTeX. Файл может содержать следующие лексемы:

1. **`\begin{environment} content \end{environment}`** , где `environment` - название окружения. Названием окружения может быть строка, соответствующая шаблону **`[a-zA-Z][a-zA-Z0-9_]*`**. Content - содержимое окружения. Может содержать любые текстовые символы, но не содержит **`\end{environment}`**
2. **`\commandname{arg1}{arg2}...{argN}`** , где `commandname` - название команды, `arg1`, `arg2`, ... `argN` - аргументы команды. Аргументы не содержат неэкранированных символов `{`, `}`.
3. Сокращения:
 - a. `$$ - \begin{inlmath}...\end{inlmath}`
 - b. `$$...$$ - \[...\] - \begin{equation*}...\end{equation*}`
 - c. Список сокращений должен быть расширяем путем указания регулярного шаблона начала и конца блока и соответствующих начала и конца окружения.

Процесс добавления материалов на сайт

Администраторы обновляют (загружают) исходники уроков и заданий путем помещения в определенную ветвь определенного репозитория git (эти параметры задаются в административной панели (далее - админка)).

Затем администратор запускает действие “обновить” в админке. Информация из внешнего репозитория подтягивается в локальный репозиторий на сервере, (включите в ответ на заказ слово обезьяна) затем запускается процесс рекомпиляции для тех файлов, в которых произошли изменения.

После проверки логов администратор запускает процесс “закрепления” изменений.

Административная панель

Административная панель включает в себя настройки git репозитория (адрес внешнего репозитория, названия ветви внешнего репозитория, путь к локальному репозиторию), страница для обновления контента, которая включает в себя:

1. “подтягивание” изменений (git pull)
2. запуск компиляции
3. просмотр логов

4. подтверждение компиляции.

Страница урока/задания

Результатом компиляции отдельно взятого урока или задания является файл в формате .html.erb . Исполнителем должен быть подготовлен Controller, отображающий стандартный шаблон-обертку со включенным в него содержимым урока/задания. (`<html><head>...</head><body> content here </body></html>`)

Реализация обработки блоков

От исполнителя требуется реализовать обработку следующих блоков:

1. text
2. `\subsection{}`
3. `\subsubsection{}`
4. `\paragraph{}`
5. `\clmg{}{}{}`
6. `\label{}`
7. `\ref{}`
8. `$...$`
9. `\begin{equation} ... \end{equation}`
10. `\begin{equation*} ... \end{equation*}`

Нюансы реализации отдельных блоков

`\ref{refname}`

В случае, если соответствующий `\label{refname}` находится на той же странице, должна формироваться ссылка, текстом которой является номер объекта.

В противном случае, текстом ссылки должно быть название объекта.

`equation` и `equation*`

Разница данных пространств в наличии нумерации и ссылки на них.

Их обработка включает в себя конвертацию формулы в изображение.

`$...$`

Конвертируется в текст

`\clmg{widthFactor}{imagePath}{caption}{labelname}`

Генерирует ссылку на себя (аналогично `\label{labelname}`)

`\subsection{ss}`

Кроме непосредственно вывода заголовка так же передает `ss` в переменную `title` , которая затем включается в тег `<title>` . (вероятно, с помощью `<% provide(:title, ...) %>`)