*ВЕСТНИК ЮГОРСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА*
2011 г. Выпуск 3 (22). С. 27–30

УДК 004.422.83

# СОЗДАНИЕ WEB-ПРИЛОЖЕНИЯ ДЛЯ ГЕНЕРАЦИИ ОТЧЕТОВ НА БАЗЕ КАРКАСНОЙ СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ «DJANGO»

А. Ю. Мелихов, Р. Ф. Юхимук

## Введение

Каркасные системы управления контентом (Content Management Framework – CMF) в настоящее время могут рассматриваться как очередной виток эволюции инструментов создания информационных систем с web-интерфейсом, поскольку они предоставляют практически неограниченный набор возможностей, позволяющих облегчить процесс разработки информационной системы (ИС), повысить надежность и безопасность, а также расширить функционал ИС с минимальными ресурсными затратами. Современные каркасные системы управления контентом позволяют не только генерировать HTML-страницы, но и предоставляют интерфейс доступа к базе данных. В этой связи использование CMF становится все более востребованным, так как в корпоративных приложениях наиболее распространенными функциями являются функции, отвечающие за взаимодействие с БД (т. н. функции «CRUD» – Create-Read-Update-Delete) и генерацию различных отчетов. В CMF часть указанных выше функций зачастую уже реализована и разработчику остается лишь адаптировать их для решения стоящих перед ним задач.

Информация, получаемая из БД с помощью высокоуровневых функции «CRUD», часто требует дальнейшей обработки в других приложениях. Для этого желательно предусмотреть выгрузку отчетов установленной формы в формате CSV (Comma Separated Values – значения, разделённые запятыми), который в настоящее время широко используется при обмене данными между ИС. Вместе с тем на практике в большинстве случаев бывает затруднительно априорно предусмотреть все необходимые отчеты, структура которых к тому же может изменяться в процессе эксплуатирования ИС. В этой связи возникает потребность в механизме, который позволил бы администратору системы, не меняя программный код, гибко изменять структуру существующих отчетов, а также создавать новые отчеты, структура которых была заранее неизвестна. Настоящая работа посвящена реализации указанного механизма генерации отчетов в ИС, построенной на базе CMF Django [1, 2].

## Структура проекта

Современные каркасные системы управления контентом построены на базе архитектуры MVC – модель-представление-контроллер (Model-View-Controller), в которой модель данных приложения, пользовательский интерфейс и управляющая логика разделены на три отдельных слоя таким образом, что модификация одного из них оказывает минимальное воздействие на остальные [1, 2]. Компонентами архитектуры MVC являются: модель (Model – предоставляет данные, а также реагирует на запросы, изменяя своё состояние), представление (View – отвечает за отображение информации (пользовательский интерфейс)), контроллер (Controller – интерпретирует данные, введённые пользователем, и информирует модель и представление о необходимости соответствующей реакции).

Немаловажным критерием при выборе CMF является лицензия, под которой распространяется платформа. CMF Django распространяется под лицензией BSD (Berkley Software Distribution – Программная лицензия университета Беркли), которая в настоящее время является одной из самых популярных лицензий для свободного программного обеспечения, поскольку налагают меньше ограничений на пользователя.

На рис. 1 приведена структура типового проекта в CMF Django. Коротко охарактеризуем каждую ее составляющую.

Пользователь с помощью браузера посылает HTTP-запросы веб-серверу через канал связи. В файле контроллера (urls.py) описывается порядок выбора одного из нескольких представлений в зависимости от URL-адреса запроса, поступившего от браузера пользователя. В CMF Django представления играют роль интерфейса между моделью данных и шаблоном, который используется для публикации данных на HTML-форме в заданном виде. Шаблон представляет собой HTML-разметку, в которую были добавлены переменные и шаблонные теги. Отображение объектов модели в таблицы БД происходит автоматически посредством использования механизма объектно-реляционного преобразования (Object-Relational Mapping – ORM), реализованного в Django.

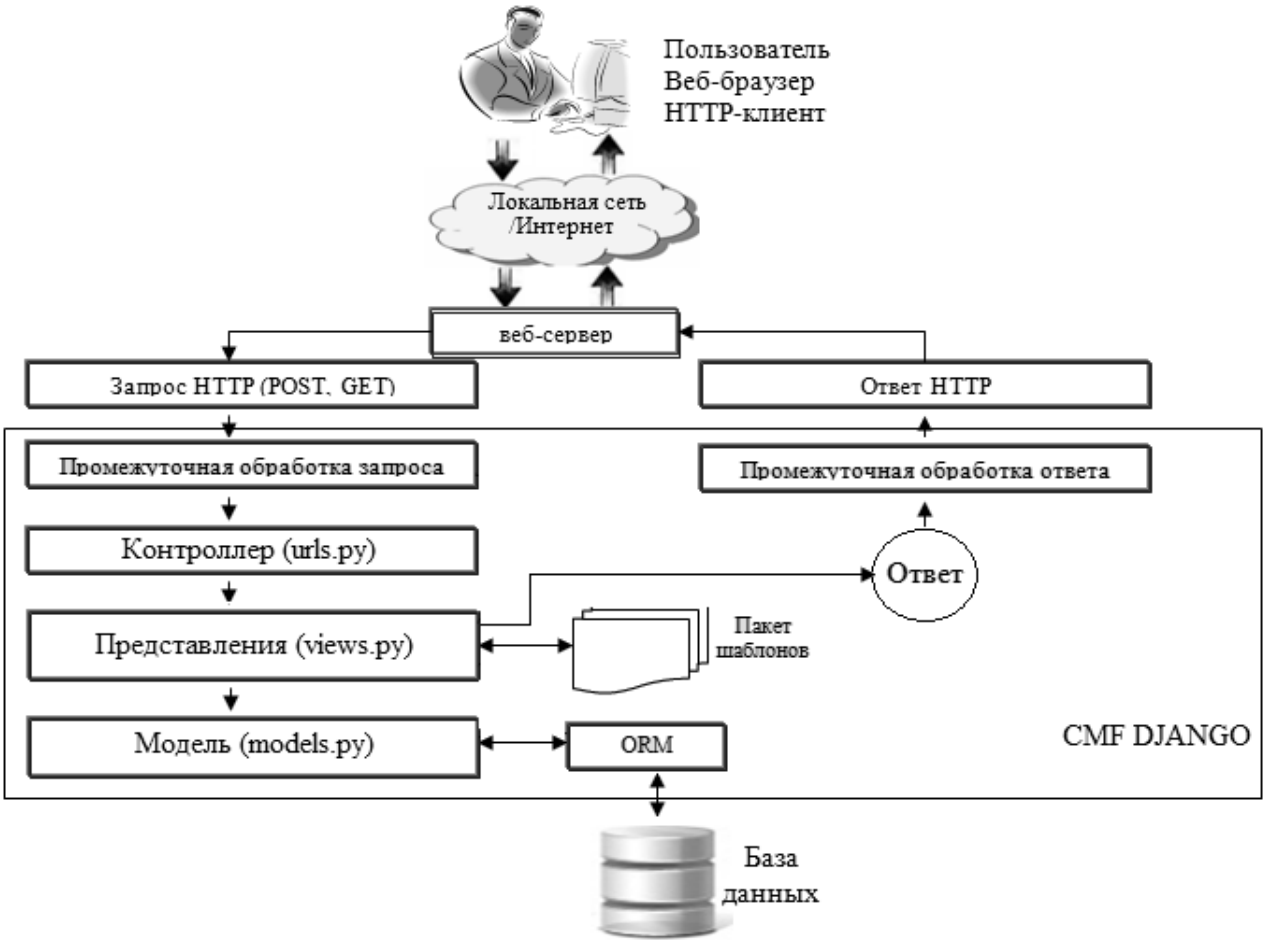


Рис. 1. Структура проекта в CMF Django

Принцип работы web-приложения на платформе CMF Django состоит в следующем: поступающие HTTP-запросы передаются веб-сервером платформе CMF Django, которая принимает их на промежуточном уровне обработки запросов (на промежуточном уровне происходит первичная обработка заголовка HTTP-запроса). После этого, исходя из шаблонов URL-адресов, контроллер (urls.py) передает запросы соответствующему представлению, которое выполняет основную часть работы, задействуя при этом модель и/или шаблоны, необходимые для создания ответа. Затем ответ проходит один или более промежуточных уровней, на которых выполняется окончательное формирование HTTP-ответа обратно веб-серверу, который, в свою очередь, отправляет ответ пользователю [2].

**Реализация проекта**

Одной из основных задач проекта является выгрузка отчета в формате CSV для дальнейшего использования его в таких приложениях, как Microsoft Office или Open Office. Поскольку формат CSV широко распространен для обмена данными между приложениями, разработчики CMF Django реализовали метод выгрузки отчета в формате CSV с применени-

ем шаблона. В этом случае программист создает для каждого отчета отдельный класс в файле models.py.

Например, на рис. 2 (а) приведен вид отчета, которому соответствует класс, синтаксис которого показан на рис. 2 (б).

Атрибуты класса соответствуют полям формируемого отчета. На рис. 2 (б) показан шаблон, предназначенный для генерации указанного отчета в виде HTML-формы. В двойных фигурных скобках указывается название переменной, содержащей данные для отчета, символ «;» необходим для указания местоположения ячейки на странице. Шаблонный тег {% for %} позволяет выполнить обход всех элементов последовательности.

| | A | B | C |
|---|---|---|---|
| 1 | | Отчет № 1 | |
| 2 | | | |
| 3 | Дата: | 21.05.2011 | |
| 4 | | | |
| 5 | Предприятие | Местоположение | Дата создания |
| 6 | ЗАО "Елки" | г. Ханты-Мансийск | 01.06.2011 |
| 7 | | | |

(а)

```
{% for row in data %};Отчет №{{ row.0}},

Дата: ;{{ row.1}},
Предприятие;Местоположение;Дата создания
{{ row.2}},{{ row.3}}, {{ row.4}}
{% endfor %}
```

(в)

Рис. 2. Типовой отчет, формируемый в CMF Django с помощью шаблона:
(а) вид отчета; (б) класс модели; (в) фрагмент шаблона

Зачастую на практике предусмотреть априорно и спроектировать все необходимые отчеты достаточно затруднительно. В этом случае перед программистом стоит задача, которая имеет строгое решение в реляционной алгебре, которое зачастую приводит к увеличению времени обработки SQL запроса (Structured Query Language – структурированный язык запросов).

Для решения указанных выше проблем предлагается реализовать собственный метод выгрузки отчета в формате CSV.

На рис. 3 представлен фрагмент модели предметной области, на котором приведены классы необходимые для работы системы, они соответствуют сущностям предметной области. В качестве комментария для того или иного класса приведен фрагмент листинга кода на языке Python, демонстрирующий определение атрибутов (имя, тип (CharField, ForeignKey, IntegerField и проч.), параметры (verbose_name, max_length и проч.)). Благодаря реализации в Django механизма объектно-реляционного преобразования любое изменение количества или типа атрибутов классов предметной области будет отображаться на структуру БД автоматически.

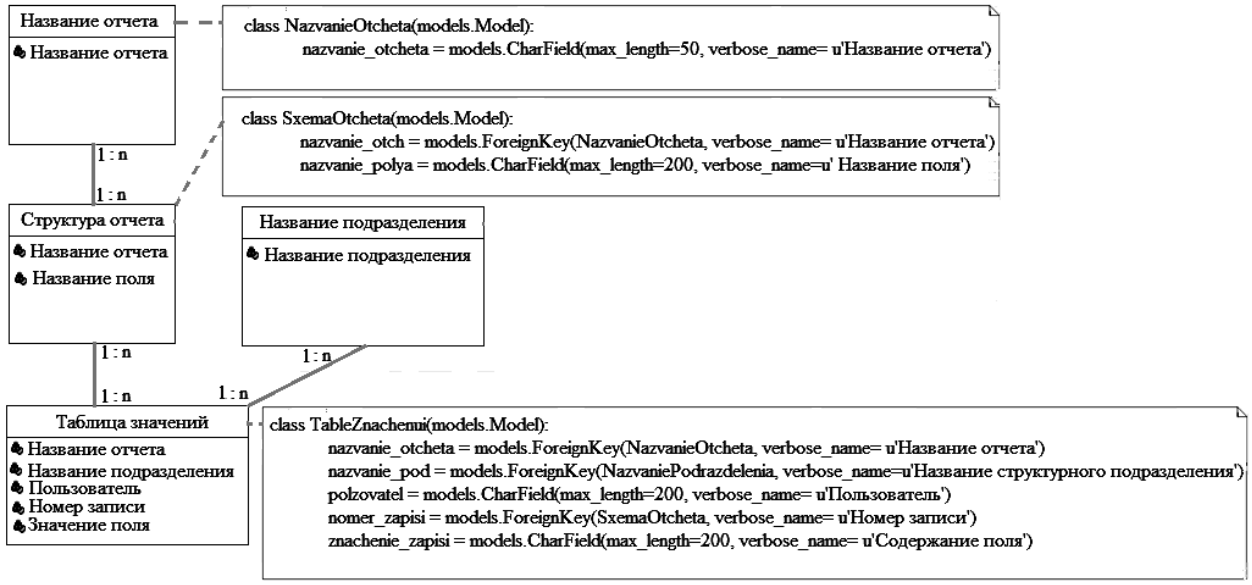


Рис. 3. Фрагмент модели предметной области

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Предприятие | Местоположение | Дата создания | |
| 2 | ЗАО "Ёлки" | г. Ханты-Мансийск | 01.06.2011 | Отдел по работе с клиентами |
| 3 | ОАО "Вестник" | г. Нижневартовск | 01.05.2011 | Отдел юриспруденции |
| 4 | | | | |

Рис. 4. Фрагмент отчета в формате CSV

## Заключение

В настоящей работе обсуждается проблема построения отчетов, характерная для большинства каркасных систем управления контентом. Традиционный подход формирования отчетов предполагает использование классов, описываемых в слое модели, на стадии проектирования информационной системы. При этом в последующем для изменения структуры того или иного отчета потребуется модифицировать соответствующий ему класс. Предлагаемый подход позволяет исключить необходимость модификации программного кода приложения за счет использования механизма, описанного в настоящей статье, и предполагающего хранение описания структуры отчета в служебной таблице реляционной БД. В этом случае изменение описания отчета в БД с помощью, например, административного интерфейса CMF позволит изменить его структуру без модификации программного кода. Описанное в работе решение было реализовано на языке Python в каркасной системе управления контентом Django.

## ЛИТЕРАТУРА

1. Holovaty, A. The Definitive Guide to Django: Web Development Done Right / A. Holovaty, J. Kaplan-Moss. – NY : Apress, 2009. – 500 p.
2. Форсье, Дж. Django. Разработка веб-приложений на Python / Дж. Форсье, П. Биссекс, У. Чан. – СПб. : Символ-Плюс, 2010. – 456 с.