

Django

Веб-фреймворк

Веб-фреймворк

Каркас веб-приложений (Web application framework, WAF) – это каркас, предназначенный для создания динамических вебсайтов, сетевых приложений, сервисов или ресурсов. Он упрощает разработку и избавляет от необходимости написания рутинного кода. Многие каркасы упрощают доступ к базам данных, разработку интерфейса, и также уменьшают дублирование кода.

Главная цель Фреймворка - предоставить программисту удобную среду и структуру проекта с большим и хорошо расщиряемым функционалом.

MVC

Model-view-controller (MVC, «модель-представление-контроллер») – схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области.

Django

Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка — DRY (англ. Don't repeat yourself)

Django

Также, в отличие от других фреймворков, обработчики URL в Django конфигурируются явно при помощи регулярных выражений, а не выводятся автоматически из структуры моделей контроллеров.

Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

Django / Основные принципы

- Django стремится быть “питоническим”;
- DRY(Don't Repeat Yourself);
- Слабая зависимость и гибкость;
- Быстрая разработка.

Модели

Модели отображают информацию о данных, с которыми вы работаете. Они содержат поля и поведение ваших данных. Обычно одна модель представляет одну таблицу в базе данных.

Основы:

- Каждая модель это класс унаследованный от `django.db.models.Model`;
- Атрибут модели представляет поле в базе данных;
- Django предоставляет автоматически созданное API для доступа к данным; смотрите [Выполнение запросов](#).

Настройка представления стандартных ошибок

Когда Django не может найти регулярное выражение, соответствующее требуемому URL, или возникает исключение, Django вызовет соответствующий обработчик стандартных ошибок.

- `handler404;`
- `handler500;`
- `handler403;`
- `handler400.`

Шаблонизатор

Jinja — это шаблонизатор для языка программирования Python. Он подобен шаблонизатору Django, но предоставляет Python-ические выражения, обеспечивая исполнение шаблонов в песочнице. Это текстовый шаблонизатор, поэтому он может быть использован для создания любого вида разметки, а также исходного кода. Лицензирован под BSD лицензией.

Шаблонизатор Jinja позволяет настраивать теги, фильтры, тесты и глобальные переменные. Также, в отличие от шаблонизатора Django, Jinja позволяет конструктору шаблонов вызывать функции с аргументами на объектах.

Спасибо

Спасибо за внимание!