
django

Django Models

- Представляют собой объектную обертку над SQL данными

- Поддерживают большое количество возможностей по поиску/изменению/анализу

- Работают с большим количеством данных

Работа с моделями

- Модели определяются как дочерние классы от Model
- Они могут наследовать друг друга
- Они могут иметь различные "отношения" друг с другом

В Postgres есть дополнительная функциональность

https://docs.djangoproject.com/es/1.9/ref/contrib/postgres/fields
https://docs.djangoproject.com/en/1.9/ref/contrib/postgres/

Написание запросов

- https://docs.djangoproject.com/ja/1.9/topics/db/queries/

Результат операции

- Любой запрос (почти) вернет QuerySet
- QuerySet - ленивый, он не будет выполнен, пока не потребуется
- https://docs.djangoproject.com/en/1.9/topics/db/queries/#querysets-are-lazy
- https://docs.djangoproject.com/en/1.9/ref/models/querysets/#when-querysets-are-evaluated

Практика

Приложение "pizza"

Migration

- Миграции позволяют нам менять структуру базы данных

- Вся база будет привязана к текущей версии кода

- Можно генерировать миграции при помощи фреймворка или писать самостоятельно

- https://docs.djangoproject.com/en/1.9/topics/migrations/

Manager

- Класс, который выполняет наши запросы, называется Manager

- У него есть удобная возможность изменять запросы при выполнении

- https://docs.djangoproject.com/en/1.9/topics/db/managers/

Сигналы

- В Django есть возможность пользоваться сигналами, которые будут срабатывать после определенных действий с базой, например: сохранение или удаление объекта

- Существует разница, когда сигнал сработает, а когда нет

- https://docs.djangoproject.com/en/1.9/topics/signals/