Apps, Models and Migrations

Reindert-Jan Ekker

@rjekker   http://nl.linkedin.com/in/rjekker

Summary

Apps

- What is a Django App
- Why do we need them
- Adding an App

Models

- Allow storing data in a db
- Creating a Model
- Field Types

Migrations

- Generating and updating the database
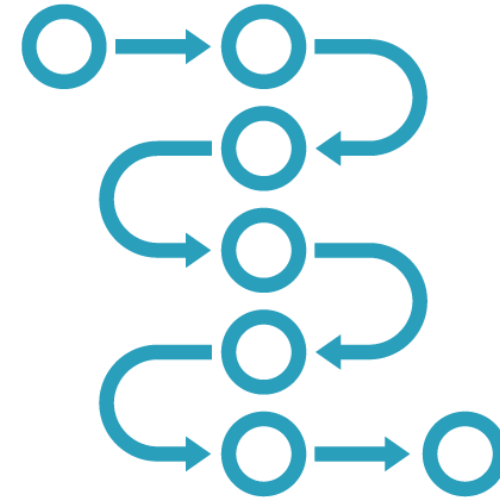
Models and Migrations

Models

Python classes

Mapped to database tables

Each object is a row in the table

Migrations

Python scripts

Keep db structure in sync with code

Auto-generated (but not always)

Not a part of runtime code

Django Apps

Python package

Contains models, views, templates, urls

Most Django projects contain several apps

Apps can be reused between projects

Keep apps small and simple
- Should have a clear, simple purpose

Demo

Running Initial Migrations

Creating a New App

Explore

Demo

Adding Model classes

Demo

Creating and Running Migrations

```
python manage.py startapp appname


# in settings.py

INSTALLED_APPS = [ ..., "appname" ]
```

Adding a New App

The `manage.py` script is located inside the project

Django ORM

Object-relational mapping

Interact with persistent data from Python
(we think in classes and objects)
Generates database and SQL

Backends supported:
MySQL, Postgres, Oracle, SQLite
With 3rd party library:
DB2, MS SQL, and many more

```python
# in models.py

from django.db import models

class Move(models.Model):

    x = models.IntegerField()

    y = models.IntegerField()
```

Model Classes

Subclass of django.db.models.Model

Mapped to a database table (named after the class)

Model Fields

Class attributes are mapped to db columns

Should be instances of a Field class
- E.g. IntegerField, CharField

Field class determines
- Database column type (INTEGER, VARCAR)
- How the field is rendered in a form

Documentation: https://goo.gl/yQUtYW

Model Field Options

```python
# Make Field Nullable (default is non-NULL)

Models.IntegerField(null = True)

# Allow empty values in forms (Not db-related!)

Models.CharField(blank = True)

# Default value

Models.CharField(default = 'F')

# Type-specific options

models.DateTimeField(auto_now = True)
```

Migration Workflow

```
# Step 1: Change Model code

# Step 2: Generate migration script (check it!)

python manage.py makemigrations

# Optional: Show migrations

python manage.py showmigrations

# Optional: Show SQL for specific migration

python manage.py sqlmigrate appname migrationname

# Step 3: Run migrations

python manage.py migrate
```

Summary

Apps

- Packages with models, views, etc.
- `manage.py startapp appname`
- Add app to `INSTALLED_APPS`

Models

- Classes inherit from `models.Model`
- Django Field classes

Migrations

- `manage.py makemigrations`
- `manage.py migrate`