Setting up a Data Model

Reindert-Jan Ekker

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

Overview

Data Model

- Create Django model classes

- Create and run migrations

- Edit data with admin interface
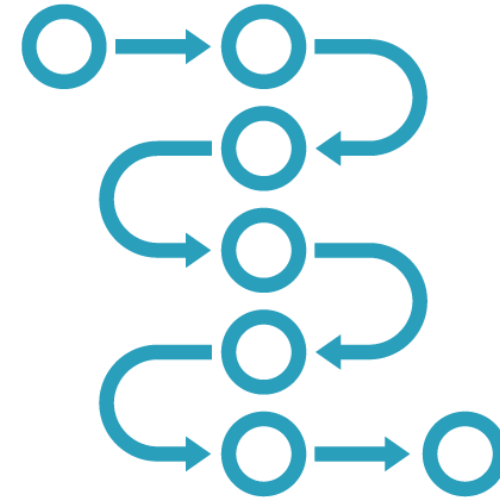
Models and Migrations

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)

Demo

Running initial migrations

Demo

Creating a data model

- Creating an app
- Adding a model class
- Create and run migration

Demo

Admin interface

- Create and edit model data

- Register model with admin site

- Configure superuser

Django Models

Saving Python objects in a database

Model classes are mapped to tables

- Fields are mapped to columns

SQL is generated

- Create/change tables (migrations)
- Insert/update/delete rows (admin)

Supported databases:

- PostgreSQL, MariaDB, MySQL, Oracle, SQLite
- With package: DB2, MS SQL, and more

Model Classes

```python
from django.db import models


class Person(models.Model):
    name = models.CharField(max_length=100)

    age = models.IntegerField()
```

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
```

Registering Models with the Admin Site

admin.py

```python
from django.contrib import admin

from .models import Meeting

admin.site.register(Meeting)


# Don't forget to create a superuser (in the terminal)

python manage.py createsuperuser
```

Demo

Bringing it all together
- Workflow
- Add another model
- Add a foreign key

Summary

Data Model
- Create Django model classes
- Create and run migrations
- Edit data with admin interface