Starting a Django Project

Reindert-Jan Ekker
nl.linkedin.com/in/rjekker/
@rjekker

pluralsight
hardcore dev and IT training

In This Module

- **Create a new project**
- **Adding a “Hello, World” page**
  - Create a new App
  - Adding a simple view and template
  - Adding styling, JavaScript, static files
- **Run the application**
- **MTV design pattern**

Creating a New Project

- `django-admin.py startproject projectname`
- On Windows: make sure your path is set to the django installation

Running the Project

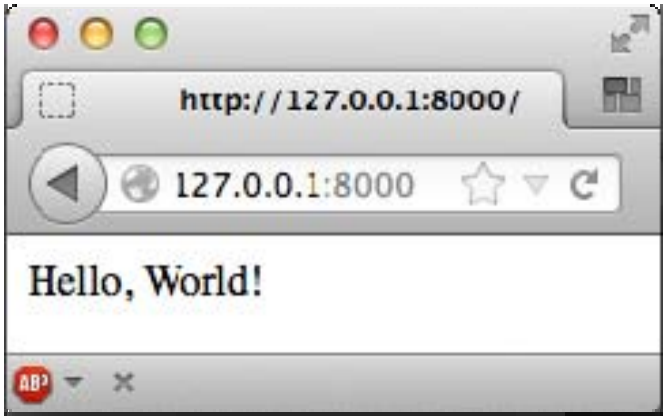
- **To run the new project**
  - `cd boardgames`
  - `python manage.py runserver`
- **The development server**
  - Will reload python code automatically
  - Don't use this as a production server!

Django Apps

- **An app is**
  - A Python package
  - A more-or-less complete web application
  - With it's own models, views, templates, url mappings
- **A typical Django project consists of one or more apps**
- **Keep your apps small and simple**
  - Each app should have one simple, well-defined purpose
- **Apps can be reusable between projects**

Adding a new App

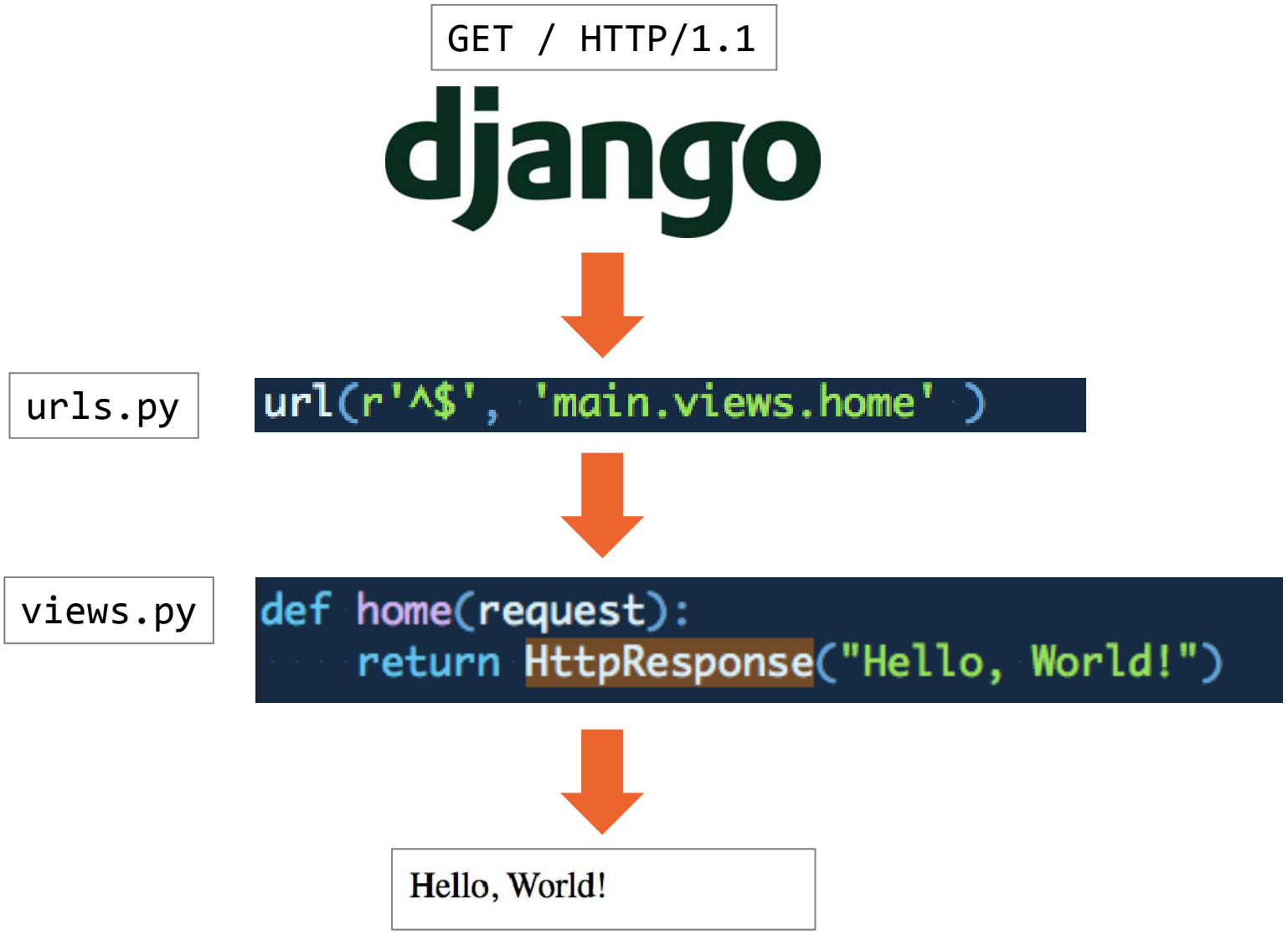
- `python manage.py startapp appname`
- Add ‘appname’ to INSTALLED_APPS in settings.py




```
GET / HTTP/1.1
```

runserver

URLs and Views



How URLs are mapped

- **When an HTTP request comes in for some URL**
- **Django looks in urls.py**
  - Finds a `urlpatterns` variable
  - This holds a list of url mappings
- **Tries to find a pattern that matches the URL**
- **More about python regex: http://goo.gl/5uJsfy**

Django Views

- **Django Views are what other MVC frameworks call “Controllers”**
- **A view is a callable**
  - That takes a request object
  - And returns a response object

Returning a HTTP Response

- **Returning a HTML page**
  - Status code 200
  - Response contains HTML
  - Simply return a new HttpResponse instance

```
def home(request):
    return HttpResponse("Hello, World!")
```

- **But we don't want our Views to contain presentation logic**
  - Move generation of HTML to a template instead

Templates

```
def home(request):
    return render(request, "main/home.html")
```

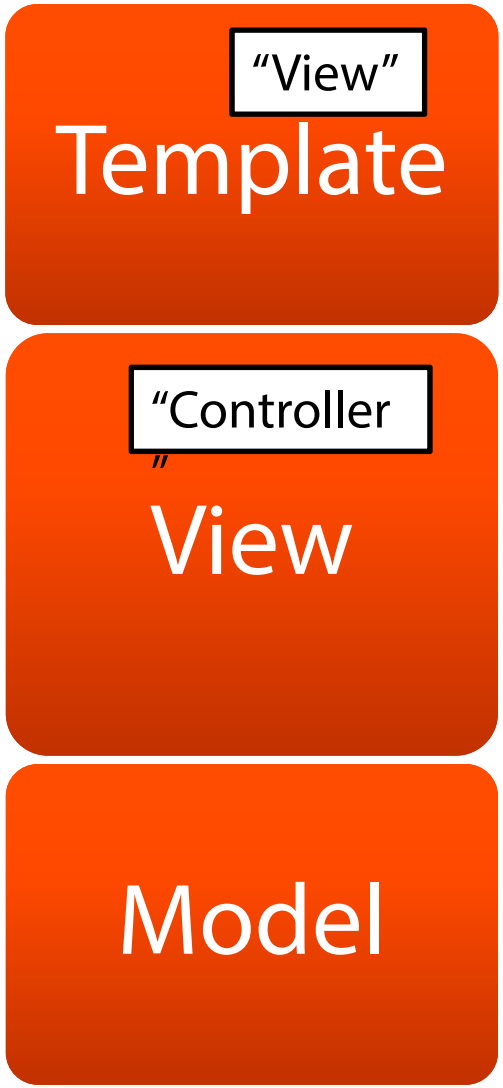
- **Can render HTML or any kind of text-based format**
- **To render a template from a view:**
  - Use `django.shortcuts.render`
- **Templates go in the templates/ dir of your app**
- **Best practice: use another dir under templates**
  - Name it after your app
  - project/app/templates/app/template.html
  - boardgames/main/templates/main/home.html

Static Files

- For non-dynamic content like CSS, JavaScript, images
- May be hosted separately
- In settings.py
  - `STATICFILES_DIRS = (os.path.join(BASE_DIR, "static"),)`
- In templates:
  - `{% load staticfiles %}` at start of template
  - `{% static 'path/to/file' %}` to refer to static content

```
<link rel="stylesheet"
      href="{% static 'bootstrap/css/bootstrap.min.css' %}">
```

Model Template View



- Generates HTML
- Presentation logic only

- Takes HTTP request and returns response
- May use model to retrieve/store data
- May call a template to present data

- Represents your data
- Each model class represents a database table

Summary

- Creating a project
- Running it
- Creating a new App
- Model-template-view
- Mapping URLS
- Views
- Templates
- Static files