

# Séminaire Django

Yoann Beaugnon

15 décembre 2021

# 1 Introduction

- Installation

# 2 Django

- structure
- models
- views
- urls
- forms
- admin

# Installation

Séminaire  
Django

Yoann  
Beaugnon

Introduction

**Installation**

Django

structure

models

views

urls

forms

admin

## Installation

- `python -m venv env`
- `. env/bin/activate`
- `pip install django`

## Lancement

- `django-admin startproject seminaire`
- `./manage.py runserver`

## Lancement

December 15, 2021 - 18 :45 :34

Django version 4.0, using settings 'hello.settings'

Starting development server at `http://127.0.0.1:8000/`

Quit the server with CONTROL-C.

# structure

Séminaire  
Django

Yoann  
Beaugnon

Introduction  
Installation

Django

**structure**

models

views

urls

forms

admin

## structure

```
seminaire/  
  manage.py  
  seminaire/  
    __init__.py  
  asgi.py  
  settings.py  
  urls.py  
  wsgi.py
```

# Application

Séminaire  
Django

Yoann  
Beaugnon

Introduction  
Installation

Django  
**structure**  
models  
views  
urls  
forms  
admin

## Application

```
django-admin startapp hello
```

## Structure Application

```
hello/  
    admin.py  
    apps.py  
    __init__.py  
    migrations  
        __init__.py  
    models.py  
    tests.py  
    views.py
```

# application

Séminaire  
Django

Yoann  
Beaugnon

Introduction

Installation

Django

**structure**

models

views

urls

forms

admin

## settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'hello',  
]
```



# model

Séminaire  
Django

Yoann  
Beaugnon

Introduction  
Installation

Django  
structure  
**models**  
views  
urls  
forms  
admin

## models.py

```
from django.db import models

class Evenement(models.Model):
    name = models.CharField(
        max_length=64,
    )
    slug = models.SlugField(
        max_length=64,
        unique=True
    )
    date = models.DateField()
    description = models.TextField(
        max_length=1024,
    )
```

## manipulation base de donnée

```
evenement = Evenement(  
    name = 'Toto',  
    slug = 'toto',  
    date = models.DateField('01-01-2021'),  
    description = 'description de toto',  
);  
evenement.save()  
evenement_bis = Evenement.object.get(slug='toto')  
evenement_bis.delete()
```

## migration

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

# model et heritage

Séminaire  
Django

Yoann  
Beaugnon

Introduction  
Installation

Django  
structure

models  
views  
urls  
forms  
admin

## héritage

```
class Person(Profile):  
    telephone = models.CharField(  
        blank=True,  
        max_length=32,  
        verbose_name=_("telephone"),  
    )
```

## views.py | view sous forme de fonction

```
from django.http import HttpResponse

def hello_world(request):
    return HttpResponse("Hello World")
```

## views.py | view sous forme de classe

```
from django.views import View

class HelloWorldView(View):
    def get(self, request):
        return HttpResponse("Hello World")
```

# template

Séminaire  
Django

Yoann  
Beaugnon

Introduction  
Installation

Django  
structure  
models  
views  
urls  
forms  
admin

## views.py | rendu de template

```
from django.views import render

def hello_world_template(request):
    context = {
        titre : 'Hello World',
        message : 'hello,world.',
    }
    return HttpResponse(render(
        request,
        'hello/template.html',
        context,))
```

# template

Séminaire  
Django

Yoann  
Beaugnon

Introduction

Installation

Django

structure

models

views

urls

forms

admin

## template.html

```
<h1>{{titre}}</h1>  
<p>{{message}}</p>
```



# template

Séminaire  
Django

Yoann  
Beaugnon

Introduction

Installation

Django

structure

models

views

urls

forms

admin

## django template

```
{{#commentaire#}}
{{%
    commentaire
    multiligne
%}}

{%if foo == 'bar'%}}

{%endif%}}

{%for element in iterable%}}
{%endfor%}}

{% include "template_name.html" %}}
```

## Utilisez au maximum les views déjà existante

- DetailView
- ListView
- ...

## seminaire/urls.py

```
from django.urls import path, include

urlpatterns = [
    path('', include('hello.urls')),
]
```

## hello/urls.py

```
urlpatterns = [
    path('class', HelloWorld.as_view()),
    path('fn', hello_world),
]
```

# url dispatcher

Séminaire  
Django

Yoann  
Beaugnon

Introduction  
Installation

Django

structure  
models

views

**urls**

forms

admin

## urls.py

```
urlpatterns = [  
    path('evenement/<slug:slug>',  
        EvenementView.as_view()),  
]
```

## views.py

```
from django.shortcuts import get_object_or_404  
  
class EvenementView(View):  
    def get(self, request, slug):  
        evenement = get_object_or_404(  
            Evenement, slug=slug)  
        return HttpResponse(evenement.description)
```

# form

Séminaire  
Django

Yoann  
Beaugnon

Introduction

Installation

Django

structure

models

views

urls

**forms**

admin

## forms.py

```
class AddEvenementForm(forms.Form):
    name = forms.CharField(
        max_length=64,
        label='name',)
    slug = forms.SlugField(
        max_length=64,
        label='slug',)
    date = forms.DateField(
        label='date',)
    description = forms.CharField(
        label='description',
        max_length=1024,)
```

# form

Séminaire  
Django

Yoann  
Beaugnon

Introduction  
Installation

Django  
structure  
models  
views  
urls  
**forms**  
admin

## evenement\_forms.html

```
<form action="/add-evenement/" method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Submit">
</form>
```

# form

Séminaire  
Django

Yoann  
Beaugnon

Introduction  
Installation

Django  
structure  
models  
views  
urls  
forms  
admin

views.py

```
def form_view(request):  
    if request.method == 'POST':  
        form = AddEvenementForm(request.POST)  
        if form.is_valid():  
            return HttpResponseRedirect(  
                form.cleaned_data['slug'])  
    else:  
        form = AddEvenementForm()  
  
    return render(request,  
        'evenement_form.html', {'form': form})
```

# form

Séminaire  
Django

Yoann  
Beaugnon

Introduction  
Installation

Django  
structure  
models  
views  
urls  
**forms**  
admin

## forms.py

```
class ArticleForm(ModelForm):  
    class Meta:  
        model = Article  
        fields = ['name', 'slug', 'date', 'description']
```



# form

Séminaire  
Django

Yoann  
Beaugnon

Introduction  
Installation

Django  
structure  
models  
views  
urls  
**forms**  
admin

views.py

```
def form_view(request):  
    if request.method == 'POST':  
        form = AddEvenementForm(request.POST)  
        if form.is_valid():  
            form.save()  
    else:  
        form = AddEvenementForm()  
  
    return render(request,  
                  'evenement_form.html', {'form': form})
```

# admin

Séminaire  
Django

Yoann  
Beaugnon

Introduction

Installation

Django

structure

models

views

urls

forms

admin

The screenshot displays the Django administration interface. At the top, it says "Django administration" and "WELCOME, ANTHONY. VIEW SITE / CHANGE PASSWORD / LOG OUT". Below this, the "Site administration" section is visible. It contains three main categories: "AUTHENTICATION AND AUTHORIZATION", "HELLO", and "EVENTS". Each category has a table with "ADD" and "CHANGE" links. The "EVENTS" table shows a list of events with their dates and times. To the right, the "Recent actions" section shows a list of actions performed, including "Eventment object (2)", "Eventment object (1)", and "Eventment object (1)".

Site administration

WELCOME, ANTHONY. VIEW SITE / CHANGE PASSWORD / LOG OUT

AUTHENTICATION AND AUTHORIZATION

Groups	ADD	CHANGE

Users

	ADD	CHANGE

HELLO

Eventments	ADD	CHANGE

Recent actions

My actions

- Eventment object (2)  
Eventment
- Eventment object (2)  
Eventment
- Eventment object (1)  
Eventment
- Eventment object (2)  
Eventment
- Eventment object (1)  
Eventment

## admin.py

```
from django.contrib import admin

# Register your models here.
class EvenementAdmin(admin.ModelAdmin):
    pass

admin.site.register(Event, EvenementAdmin)
```