

# COMPTE RENDU DES ATELIERS 1,2 ET 3

## Atelier 1: Mis en œuvre d'un projet Django

### I. Création du backend: Le projet Django dans PyCharm

Créer un nouveau projet «ecommerce»

L'installation de Django dans notre projet avec la commande :

```
pip install Django
```

Création du projet Django dans notre projet avec la commande :

```
django-admin startproject backend
```

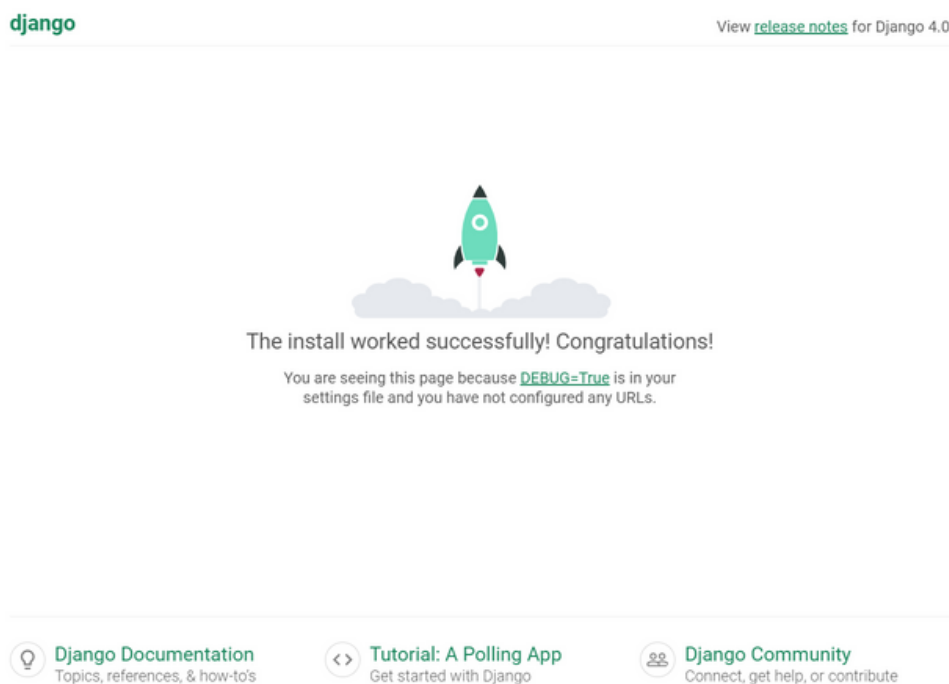
nb. Backend c'est le nom du projet Django on peut le nommer comme on veut

Test du bon fonctionnement en lançant le serveur web de Django depuis votre dossier backend :

```
Cd backend
```

```
Python manage.py runserver
```

la page d'accueil de Django doit apparaître.

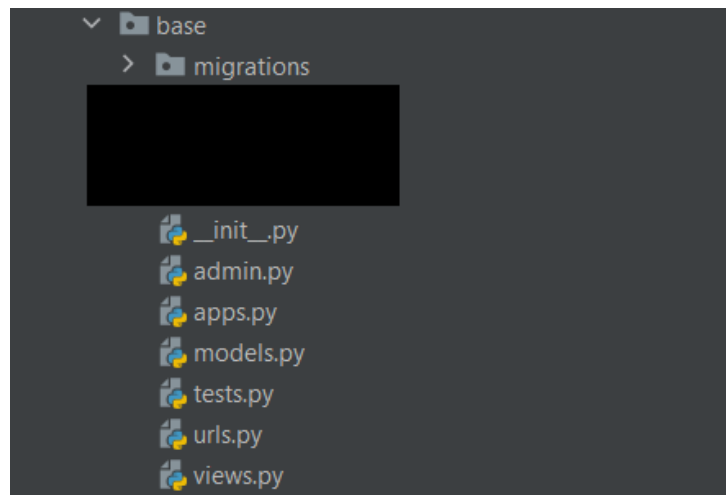


## II. Création d'une application

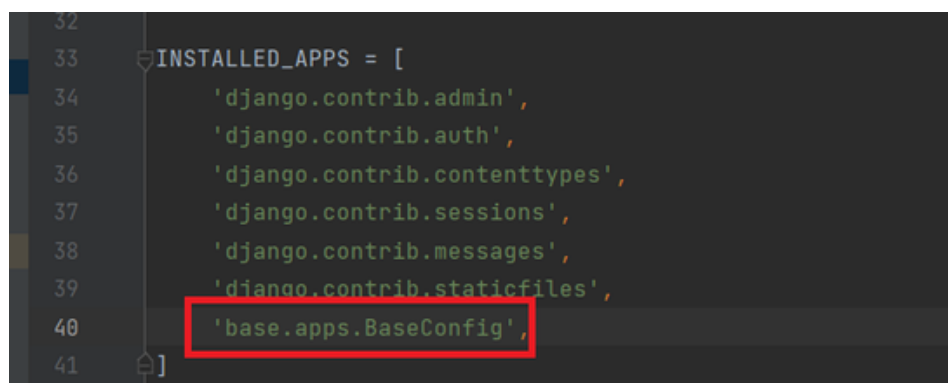
Créez l'application à l'aide de la commande:

```
python manage.py startapp base
```

Cela va créer un répertoire base



Ajouter notre application dans le fichier Settings.py



## Atelier 2: Architecture d'une application Django

### Les Url et les Vues

Ajouter dans le fichier views.py une classe nommée Home

```
class Home(View):
```

Créer dans le répertoire «base» le fichier urls.py de notre application  
définir vers quelle URL pointait cette classe :

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.Home.as_view(), name='Home'),
]
```

Inclure le fichier urls.py de notre application base dans le backend

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('base.urls'))
]
```

Retourner une réponse http

```
class Home(View):
    def get(self, request):
        return HttpResponse("<h1>Ce site représente un catallogue de produit </h1>")
```

Visualiser l'application:

```
cd Backend
python manage.py runserver
```



**Ce site représente un catallogue de produit**

créer un Template

- Créer un répertoire «templates» dans votre application «base» au sein du répertoire «templates»,
- Créer un fichier «hello.html»

```
hello.html x listProducts.html x views.py x base\urls.py x backend\urls.p
1 <html>
2   <body>
3     Ce site represente un catallogue de produit
4   </body>
5 </html>
```

modifier la View «Home»en:

```
# Create your views here.

class Home(View):
    def get(self, request):
        return render(request, 'hello.html', {})
```

Ajouter dans le fichier views.py une nouvelle classe ProductList:

```
class ProductList(View):
    def get(self, request):
        return render(request, 'listProducts.html', {'products': products})
```

router l'url /products

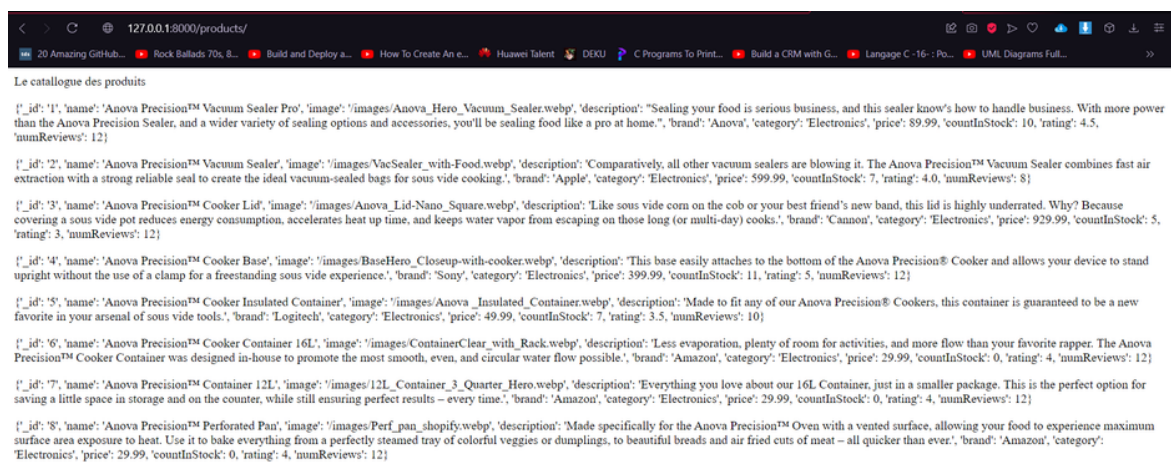
```
from . import views

urlpatterns = [
    path('', views.Home.as_view(), name='Home'),
    path('products/', views.ProductList.as_view(), name='products')
]
```

Créer un nouveau fichier «ListProduit.html» dans le dossier template:

```
hello.html x listProducts.html x base\urls.py x base.html x backend\urls.py x custom_tag.py x products.py x
<html>
  <body>
    Le catalogue des produits
    {% for product in products %}
    <p>{{product}}</p>
    {% endfor %}
  </body>
</html>
```

Nous avons obtenu l'aperçu suivant



## Atelier 3 : Template Django

### Mise en place d'une template a travers bootstrap

Creer un fichier base.html pour definir notre template

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock %}</title>
  </head>
  <body>
    {% block content %}
    {% endblock %}
  </body>
</html>
```

## Intégrer les CDN Bootstrap au fichier base.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>{% block title %}{% endblock %}</title>
6   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
7     integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztQTwFspg3yD65VohhpuuCOMLASjC" crossorigin="anonymous">
8   <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"
9     integrity="sha384-IQsoLX15PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>
10  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
11    integrity="sha384-cVKIPh6WIC2A4u+LWgxfKTR1cfu0JTxR+EQDz/bgldgEyl4H0zUF0QKbrJ9EcQF" crossorigin="anonymous"></script>
12 </head>
13 <body>
14   {% block content %}
15   </body>
16 </html>
```

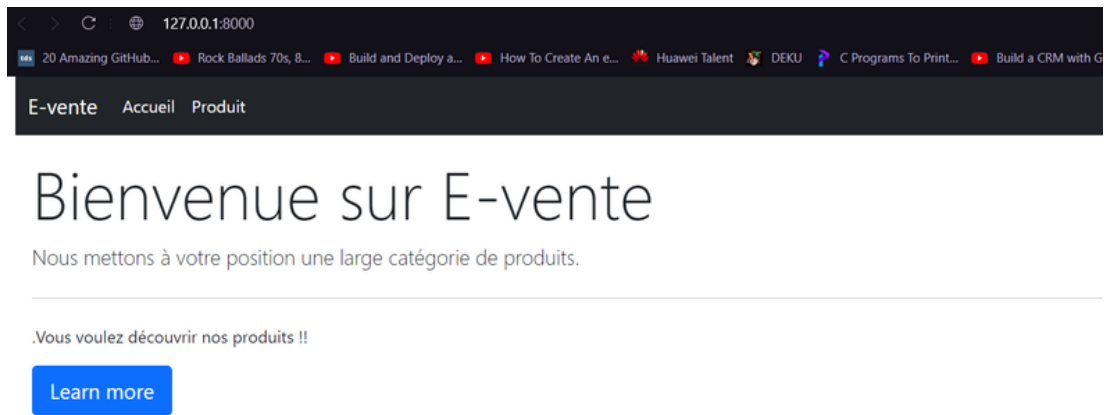
## Mettre en place une barre de navigation et pointer vers les pages

```
1 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
2   integrity="sha384-cVKIPh6WIC2A4u+LWgxfKTR1cfu0JTxR+EQDz/bgldgEyl4H0zUF0QKbrJ9EcQF" crossorigin="anonymous"></script>
3 </head>
4 <body>
5   <div class="container-fluid">
6     <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
7       <div class="container-fluid">
8         <a class="navbar-brand" href="{% url 'Home' %}">E-vente</a>
9         <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarColor02" aria-controls="navbarColor02" aria-expanded="false" aria-label="Toggle navigation">
10           <span class="navbar-toggler-icon"></span>
11         </button>
12       </div>
13       <div class="collapse navbar-collapse" id="navbarColor02">
14         <ul class="navbar-nav me-auto">
15           <li class="nav-item">
16             <a class="nav-link active" href="{% url 'Home' %}">Accueil
17             <span class="visually-hidden">(current)</span>
18           </li>
19           <li class="nav-item">
20             <a class="nav-link active" href="{% url 'Products' %}">Produits
21             <span class="visually-hidden">(current)</span>
22           </li>
23         </ul>
24       </div>
25     </nav>
26   </div>
```

## Page d'accueil de notre application

```
1 {% extends 'base.html' %}
2 {% block title %} Accueil {% endblock %}
3 {% block content %}
4   <div class="container-fluid">
5     <div class="jumbotron m-3">
6       <h1 class="display-3">Bienvenue sur E-vente</h1>
7       <p class="lead">Nous mettons à votre position une large catégorie de produits.</p>
8       <hr class="my-4">
9       <p>Vous voulez découvrir nos produits !! </p>
10      <p class="lead">
11        <a class="btn btn-primary btn-lg" href="{% url 'Products' %}" role="button">Learn more</a>
12      </p>
13    </div>
14  </div>
15 {% endblock %}
```

## Visualiser l'application



## Page de produits de notre application

```
{% extends 'base.html' %}
{% load custom_tag %}
{% block title %} Nos Produits {% endblock %}
{% block content %}
    <div class="content">
        <table class="table table-hover">
        <thead>
            <tr>
                <th scope="col">Identifiant</th>
                <th scope="col">Nom</th>
                <th scope="col">Categorie</th>
                <th scope="col">Prix</th>
            </tr>
        </thead>
        <tbody>
            {% for product in products %}
            <tr class="table-active">
                <th scope="row">Column Content</th>
                <td>Column Content</td>
                <td>Column Contentt</td>
                <td>{Column Content</td>
            </tr>
            {% endfor %}
        </tbody>
        </table>
    </div>
{% endblock %}
```

Créer un filtre pour prendre en paramètre un dictionnaire et une clé et renvoie la valeur

```
base.html x hello.html x listProducts.html x custom_tag.py x
from django import template

register = template.Library()

def get_value(dictionary, key):
    return dictionary.get(key)

register.filter('get_value', get_value)
```

Charger le filtre et utiliser le tag pour accéder aux informations des produits

```
base.html x hello.html x listProducts.html x
1 {% extends 'base.html' %}
2 {% load custom_tag %}
3 {% block title %} Nos Produits {% endblock %}
4 {% block content %}
5     <div class="content">
6         <table class="table table-hover">
7             <thead>
8                 <tr>
9                     <th scope="col">Identifiant</th>
10                    <th scope="col">Nom</th>
11                    <th scope="col">Categorie</th>
12                    <th scope="col">Prix</th>
13                </tr>
14            </thead>
15            <tbody>
16                {% for product in products %}
17                    <tr class="table-active">
18                        <th scope="row">{{ product|get_value: "_id" }}</th>
19                        <td>{{ product|get_value: "name" }}</td>
20                        <td>{{ product|get_value: "category" }}</td>
21                        <td>{{ product|get_value: "price" }}</td>
22                    </tr>
23                {% endfor %}
24            </tbody>
25        </table>
26    </div>
27 {% endblock %}
```

127.0.0.1:8000/Products/

E-vente Accueil Produit

Identifiant	Nom	Categorie	Prix
1	Anova Precision™ Vacuum Sealer Pro	Electronicst	89.99
2	Anova Precision™ Vacuum Sealer	Electronicst	599.99
3	Anova Precision™ Cooker Lid	Electronicst	929.99
4	Anova Precision™ Cooker Base	Electronicst	399.99
5	Anova Precision™ Cooker Insulated Container	Electronicst	49.99
6	Anova Precision™ Cooker Container 16L	Electronicst	29.99
7	Anova Precision™ Container 12L	Electronicst	29.99
8	Anova Precision™ Perforated Pan	Electronicst	29.99