

# Rapport du TP4 Python Django – Les vues

TCHABANA Malia

11-11-2022

Ce TP a pour objectif de nous apprendre à créer des vues en Django. Ainsi la finalité de ce TP est de produire 5 vues très importantes dont une vue pour consulter la liste des élèves avec leurs notes et leur moyenne dans chaque matière, une pour consulter le détail d'un élève, une pour consulter la liste des matières avec leur enseignant, une pour consulter le détail d'une matière et une pour consulter le détail d'un niveau.

## Contents

<b>1</b>	<b>Mise en route</b>	<b>2</b>
<b>2</b>	<b>L'écriture de vues en code Python</b>	<b>2</b>
2.1	l'URL de la racine de l'application . . . . .	2
2.2	Création des fonctions . . . . .	2
2.3	Création des routes . . . . .	2
2.4	Amélioration des fonctions . . . . .	3
2.5	Modification des vues . . . . .	3
2.6	Affichage des données de la table élève et matière . . . . .	4
<b>3</b>	<b>Les templates</b>	<b>5</b>
3.1	Définition d'un template . . . . .	5
3.2	Création du fichier index.html . . . . .	5
3.3	Utilité de la fonction render . . . . .	5
3.4	Modification des vues eleves() et matieres() et affichage des templates correspondants	5
<b>4</b>	<b>La gestion des erreurs</b>	<b>7</b>
4.1	Différence entre les vues . . . . .	7
4.2	La fonction raccourci get_object_or_404 . . . . .	7
4.3	Écriture des vues eleve.py, matiere.py et niveau.py . . . . .	7
4.4	Ligne à supprimer . . . . .	10
<b>5</b>	<b>Navigation et espaces de noms</b>	<b>10</b>
5.1	Navigation entre les pages du site . . . . .	10
5.2	Amélioration du code des templates . . . . .	11
5.3	Différence entre projet et application . . . . .	13
5.4	Création d'un espace de nom pour l'application "notes" . . . . .	13

## 1 Mise en route

Pour continuer notre travail dans le projet Django initial, nous devons tout d'abord activé l'environnement virtuel (**source virtualEnv/bin/activate**) et ensuite se positionner dans le dossier ifnti\_l3 qui est le nom du dossier de notre projet. Après cela il peut suivre le démarrage du serveur avec la ligne de commande **alias manage.py runserver**.

Compte tenu que dans ce TP nous allons travaillé sur les vues, nous allons plus nous concentré sur le fichier notes/views.py et aussi le fichier notes/urls.py car chaque fonction créée dqns le fichier views.py est attaché à une url.

## 2 L'écriture de vues en code Python

### 2.1 l'URL de la racine de l'application

Lors du premier TP, nous avons définie une route qui pointe vers la racine de l'application dans le fichier notes/urls.py. Cette Url en question est la suivante: **http://127.0.0.1:8000/notes/** et la route pointant vers cette Url est la ligne ci-dessous :

```
path('', views.index, name='index')
```

### 2.2 Création des fonctions

Dans le soucis de réussir à afficher les 5 vues décrites en introduction, nous avons tout d'abord commencé à créer des fonctions retournant des réponses HTTP. Ces fonctions agissent uniquement sur les tables élèves, matières et niveaux. La définition de ses fonctions est équivalente à : (voir l'aperçu ci-dessous)



```
def index(request):  
    return HttpResponse("Bonjour Malia !")  
  
def eleves(request):  
    return HttpResponse("Affichage de la liste des élèves")  
  
def eleve(request):  
    return HttpResponse("Affichage des identifiants d'un seul élève")  
  
def matieres(request):  
    return HttpResponse("Affichage de la liste des matières")  
  
def matiere(request):  
    return HttpResponse("Affichage d'une seule matière")  
  
def niveaux(request):  
    return HttpResponse("Affichage de la liste des niveaux")  
  
def niveau(request):  
    return HttpResponse("Affichage d'un seul niveau")
```

Figure 1: Définition des fonctions

### 2.3 Création des routes

Les fonctions ainsi définis dans la section précédente ne pourrons pas avoir effet si elles ne sont pqs attachées à des routes. Du coups nous avons créer une route pour chacune de ses fonctions.(voir l'aperçu ci-dessous)

Ce qui produit le résultats suivant :

```

urls.py > ...
from django.urls import path
from . import views
urlpatterns = [
    path('', views.index, name='index'),
    path('eleves/', views.eleves, name='eleves'),
    path('eleve/<int:id>', views.eleve, name='eleve'),
    path('matieres/', views.matieres, name='matieres'),
    path('matiere/<int:id>', views.matiere, name='matiere'),
    path('niveaux/', views.niveaux, name='niveaux'),
    path('niveau/<int:id>', views.niveau, name='niveau'),
]

```

Figure 2: Définition des routes



Figure 3: Affichage des vues

## 2.4 Amélioration des fonctions

Parmi les fonctions définies précédemment, trois fonctions doivent retourner les détails des objets de certains modèles. Pour cela nous devons préciser l'identifiant de l'objet dont nous souhaitons afficher les détails. La modification de ces fonctions équivalent à : (voir l'aperçu ci-dessous)

```

def eleve(request, id):
    return HttpResponse("Affichage des identifiants d'un seul élève")

def matiere(request, id):
    return HttpResponse("Affichage d'une seule matière")

def niveau(request, id):
    return HttpResponse("Affichage d'un seul niveau")

```

Cette modification produit le résultats suivant :



Figure 4: Affichage d'un seul niveau

## 2.5 Modification des vues

À cette étape, nous avons modifié la définition de nos fonctions de tels sorte qu'elles accèdent aux données de la base de données. Pour ce faire, nous avons donc invoqué les modèles suivant: Eleve, Matiere et Niveau.

Cette modification produit le résultats suivant :

```

def eleves(request):
    list_eleves = Eleve.objects.all()
    les_eleves=" -----La liste des élèves-----" + "<br> "
    for eleve in list_eleves:
        les_eleves = les_eleves + "L'élève ayant l'identifiant " + str(eleve.id) + " s'appelle " + elev
    return HttpResponse(les_eleves)

def eleve(request, id):
    return HttpResponse("Affichage des identifiants d'un seul élève")

def matieres(request):
    list_matieres = Matiere.objects.all()
    les_matieres="----La liste des matières-----" + "<br> "
    for matiere in list_matieres:
        les_matieres = les_matieres + matiere.nom + "<br> "
    return HttpResponse(les_matieres)

def matiere(request, id):
    return HttpResponse("Affichage d'une seule matière")

def niveau(request, id):
    return HttpResponse("Affichage d'un seul niveau")

```

## 2.6 Affichage des données de la table élève et matière

Pour afficher les données des tables élève et matière qui se trouve dans la base de données, nous avons tout d'abord récupéré les données de la base de données et ensuite nous avons fait recours à une boucle itérative **for** pour itérer sur chaque objets de la classe. Ce qui retourne tous les objets de chaque modèle,

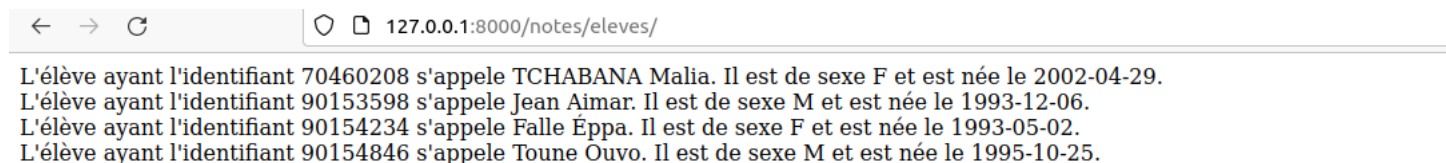


Figure 5: La liste des élèves

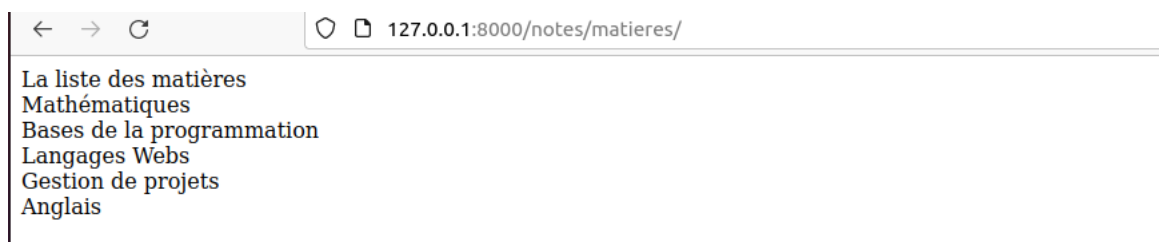


Figure 6: La liste des matières

## 3 Les templates

### 3.1 Définition d'un template

Un Template est un design qui désigne un modèle où seuls certains éléments sont modifiables. Un Template intègre des éléments déjà préconçus et faciles à modifier pour rendre la création web plus facile et rapide. L'utilisation d'un template facilite la conception d'un site internet et sa mise à jour.

### 3.2 Création du fichier index.html

Dans le fichier index.html nous avons placé le code suivant : "`< h1 > Bonjour tout le monde !< /h1 >`". Ce code va afficher dans une page html **Bonjour tout le monde** à l'appel de l'application notes.

### 3.3 Utilité de la fonction render

La fonction render est une fonction raccourci de Django qui permet de retourner des templates ou un objet HttpResponse.

### 3.4 Modification des vues `eleves()` et `matieres()` et affichage des templates correspondants

À cette étape, nous avons modifié les fonction `eleves()` et `matieres()` pour qu'elles retournent des templates en utilisant la fonction render. (voir l'aperçu ci-dessous)

#### views.py

```
def eleves(request):
    liste_eleves=Eleve.objects.all()
    context={"lesEleves":liste_eleves}
    return render(request, "notes/eleves.html", context)

def matieres(request):
    list_matieres = Matiere.objects.all()
    context={"lesMatieres":list_matieres}
    return render(request, "notes/matieres.html", context)
```

#### eleves.html

```
<h1>La liste des élèves</h1><br>
<div>
    <table border=1>
        <thead>
            <tr>
                <td>Identifiant</td>
                <td>Nom</td>
                <td>prenom</td>
                <td>Sexe</td>
                <td>Date de Naissance</td>
                <td>Niveau</td>
            </tr>
        </thead>
        <tbody>
```

```

{% for eleve in lesEleves %}
    <tr>
        <td>{{eleve.id}}</td>
        <td>{{eleve.nom}}</td>
        <td>{{eleve.prenom}}</td>
        <td>{{eleve.sexe}}</td>
        <td>{{eleve.date_naissance}}</td>
        <td>{{eleve.niveau}}</td>
    </tr>
{% endfor %}
</tbody>
</table>
</div>

```

### matieres.html

```

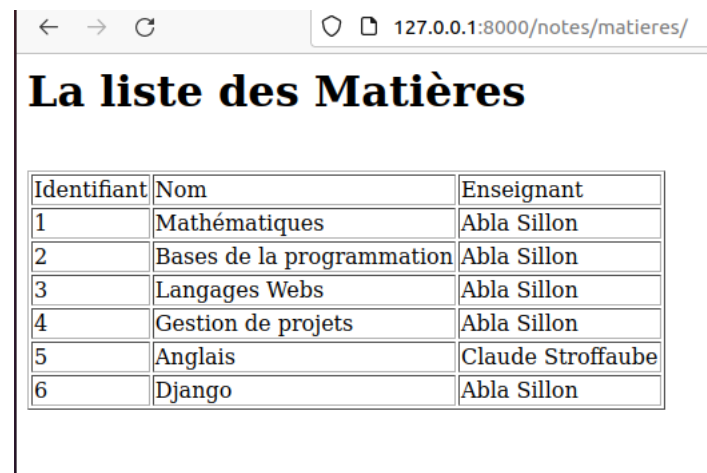
<h1>La liste des Matières</h1><br>
<div>
    <table border=1>
        <thead>
            <tr>
                <td>Nom</td>
                <td>Enseignant</td>
            </tr>
        </thead>
        <tbody>
            {% for matiere in lesMatiere %}
                <tr>
                    <td>{{matiere.nom}}</td>
                    <td>{{matiere.enseignant}}</td>
                </tr>
            {% endfor %}
        </tbody>
    </table>
</div>

```

## La liste des élèves

Identifiant	Nom	prenom	Sexe	Date de Naissance	Niveau
70460208	TCHABANA	Malia	F	29 avril 2002	L3
90153598	Jean	Aimar	M	6 décembre 1993	L3
90154234	Falle	Éppa	F	2 mai 1993	L2
90154846	Toune	Ouvo	M	25 octobre 1995	L1

Figure 7: La liste des élèves



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/notes/matieres/'. The page title is 'La liste des Matières'. Below the title is a table with three columns: 'Identifiant', 'Nom', and 'Enseignant'. The table contains six rows of data.

Identifiant	Nom	Enseignant
1	Mathématiques	Abla Sillon
2	Bases de la programmation	Abla Sillon
3	Langages Webs	Abla Sillon
4	Gestion de projets	Abla Sillon
5	Anglais	Claude Stroffaube
6	Django	Abla Sillon

Figure 8: La liste des matières

## 4 La gestion des erreurs

### 4.1 Différence entre les vues

La différence entre les vues élèves et matières et les autres vues (élève, matière et niveau) est que les vues élève, matière et niveau prennent un identifiant et retournent un seul et unique objet alors que les vues élèves et matières retournent l'ensemble des objets d'une classe donnée.

### 4.2 La fonction raccourci `get_object_or_404`

La fonction `get_object_or_404` appelle le modèle donné et obtient un objet à partir de celui-ci. Si cet objet ou ce modèle n'existe pas, il génère une erreur 404.

Donc pour les vues qu'il nous reste à écrire, cette fonction nous aidera à obtenir l'objet que nous souhaitons obtenir en spécifiant son identifiant. Et si l'objet n'existe pas, elle générera une erreur 404.

### 4.3 Écriture des vues `eleve.py`, `matiere.py` et `niveau.py`

Pour écrire ces vues nous allons nous faire aidé par la fonction `get_object_or_404`. Pour cela, nous allons donc l'importer. **`from django.shortcuts import get_object_or_404`**

Le contenu des fichiers est le suivant : (voir l'aperçu ci-dessous)

#### **views.py**

```
def matiere(request, id):
    eleve = get_object_or_404(Eleve, id=id)
    return render(request, "notes/matiere.html", {"matiere":matiere})

def eleve(request, id):
    eleve = get_object_or_404(Eleve, id=id)
    return render(request, "notes/eleve.html", {"eleve":eleve})

def niveau(request, id):
    niveau = get_object_or_404(Niveau, id=id)
    return render(request, "notes/niveau.html", {"niveau":niveau})
```

## eleve.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body border="1">
  <h1> Information propre à un élève </h1>
  <div>
    <table>
      <thead>
        <tr>
          <td><strong>Identifiant</strong></td>
          <td><strong>Nom</strong></td>
          <td><strong>Prénom</strong></td>
          <td><strong>Sexe</strong></td>
          <td><strong>Date de naissance</strong></td>
          <td><strong>Niveau</strong></td>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td> {{ eleve.id }} </td>
          <td> {{ eleve.nom }} </td>
          <td> {{ eleve.prenom }} </td>
          <td> {{ eleve.sexe }} </td>
          <td> {{ eleve.date_naissance }} </td>
          <td> {{ eleve.niveau }} </td>
        </tr>
      </tbody>
    </table>
  </div>
</body>
</html>

```

Ce code produit le résultat suivant : (voir l'aperçu ci-dessous)



Identifiant	Nom	Prénom	Sexe	Date de naissance	Niveau
70460208	TCHABANA	Malia	F	29 avril 2002	L3

Figure 9: Affichage d'un seul élève

## matiere.html



```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1> Information propre à une matière</h1>
  <table>
    <tr>
      <td><strong>Identifiant</strong></td>
      <td><strong>Nom</strong></td>
      <td><strong>Enseignant</strong></td>
    </tr>
    <tr>
      <td> {{ matiere.id }} </td>
      <td> {{ matiere.nom }} </td>
      <td> {{ matiere.enseignant }} </td>
    </tr>
  </table>
</body>
</html>

```

Ce code produit le résultat suivant : (voir l'aperçu ci-dessous)



Figure 10: Affichage d'une seule matière

### niveau.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1> Information propre à un niveau</h1>
  <table>
    <tr>
      <td><strong>Identifiant</strong></td>

```

```

        <td><strong>Nom</strong></td>
    </tr>
    <tr>
        <td> {{ niveau.id }} </td>
        <td> {{ niveau.nom }} </td>

    </tr>
</body>
</html>

```

Ce code produit le résultat suivant : (voir l’aperçu ci-dessous)



Figure 11: Affichage d’un seul niveau

#### 4.4 Ligne à supprimer

Maintenant que nous n’utilisons plus la fonction `HttpResponse()` pour retourner nos pages, son existence dans nos fichier est donc inutile. Nous allons donc supprimer la ligne d’importation de cette fonction.

## 5 Navigation et espaces de noms

### 5.1 Navigation entre les pages du site

Dans le soucis de pouvoir naviguer entre les pages du site, pouvoir afficher les détails d’un élève ou d’une matière, nous allons faire recours à la balise ”a” en indiquant le chemin de la page à afficher `<a href=’/notes/[chemin]/’>`. Le code des différents fichiers en particulier `matieres.html` et `eleves.html` est le suivant :

#### eleves.html

```

<h1>La liste des élèves</h1><br>
<div>
    <table border=1>
        <thead>
            <tr>
                <td>Identifiant</td>
                <td>Nom</td>
                <td>prenom</td>
                <td>Sexe</td>
                <td>Date de Naissance</td>
                <td>Niveau</td>
                <td>Détails</td>
            </tr>
        </thead>

```

```

<tbody>
  {% for eleve in lesEleves %}
    <tr>
      <td>{{eleve.id}}</td>
      <td>{{eleve.nom}}</td>
      <td>{{eleve.prenom}}</td>
      <td>{{eleve.sexe}}</td>
      <td>{{eleve.date_naissance}}</td>
      <td>{{eleve.niveau}}</td>
      <td><a href="/notes/eleve/{{eleve.id}}">Lien</a></td>
    </tr>
  {% endfor %}
</tbody>
</table>
</div>

```

### matieres.html

```

<h1>La liste des Matières</h1><br>
<div>
  <table border=1>
    <thead>
      <tr>
        <td>Identifiant</td>
        <td>Nom</td>
        <td>Enseignant</td>
        <td>Détails</td>
      </tr>
    </thead>
    <tbody>
      {% for matiere in lesMatieres %}
        <tr>
          <td>{{matiere.id}}</td>
          <td>{{matiere.nom}}</td>
          <td>{{matiere.enseignant}}</td>
          <td><a href="/notes/matiere/{{matiere.id}}">Lien</a></td>
        </tr>
      {% endfor %}
    </tbody>
  </table>
</div>

```

Ce qui nous retourne les pages web suivantes avec existence d'une nouvelle colonne de détails contenant des liens qui affichent le détail de l'objet choisie .

## 5.2 Amélioration du code des templates

Nous avons utiliser la balise "**a**" en indiquant le chemin vers nos pages pour retourner les liens qui pointent vers les pages de détails de nos éléments. Cette manière de faire produit le résultat souhaité.



Identifiant	Nom	prenom	Sexe	Date de Naissance	Niveau	Détails
70460208	TCHABANA	Malia	F	29 avril 2002	L3	<a href="#">Lien</a>
90153598	Jean	Aimar	M	6 décembre 1993	L3	<a href="#">Lien</a>
90154234	Falle	Éppa	F	2 mai 1993	L2	<a href="#">Lien</a>
90154846	Toune	Ouvo	M	25 octobre 1995	L1	<a href="#">Lien</a>

Figure 12: Affichage de la liste des élèves



Identifiant	Nom	Enseignant	Détails
1	Mathématiques	Abla Sillon	<a href="#">Lien</a>
2	Bases de la programmation	Abla Sillon	<a href="#">Lien</a>
3	Langages Webs	Abla Sillon	<a href="#">Lien</a>
4	Gestion de projets	Abla Sillon	<a href="#">Lien</a>
5	Anglais	Claude Strofpaube	<a href="#">Lien</a>
6	Django	Abla Sillon	<a href="#">Lien</a>

Figure 13: Affichage de la liste des matières

Mais nous pouvons améliorer notre code en procédant autrement. Ce qui à été fait dans les fichiers renseignés ci-dessous :

### eleves.html

```
<h1>La liste des élèves</h1><br>
<div>
  <table border=1>
    <thead>
      <tr>
        <td>Identifiant</td>
        <td>Nom</td>
        <td>prenom</td>
        <td>Sexe</td>
        <td>Date de Naissance</td>
        <td>Niveau</td>
        <td>Détails</td>
      </tr>
    </thead>
    <tbody>
      {% for eleve in lesEleves %}
        <tr>
          <td>{{eleve.id}}</td>
          <td>{{eleve.nom}}</td>
```

```

        <td>{{eleve.prenom}}</td>
        <td>{{eleve.sexe}}</td>
        <td>{{eleve.date_naissance}}</td>
        <td>{{eleve.niveau}}</td>
        <td><a href="{% url 'eleve' eleve.id %}">Lien</a></td>
    </tr>
    {% endfor %}
</tbody>
</table>
</div>

```

### matieres.html

```

<h1>La liste des Matières</h1><br>
<div>
    <table border=1>
        <thead>
            <tr>
                <td>Identifiant</td>
                <td>Nom</td>
                <td>Enseignant</td>
                <td>Détails</td>
            </tr>
        </thead>
        <tbody>
            {% for matiere in lesMatiere %}
                <tr>
                    <td>{{matiere.id}}</td>
                    <td>{{matiere.nom}}</td>
                    <td>{{matiere.enseignant}}</td>
                    <td><a href="{% url 'matiere' matiere.id%}">Lien</a></td>
                </tr>
            {% endfor %}
        </tbody>
    </table>
</div>

```

## 5.3 Différence entre projet et application

Une application fait référence à un sous-module du projet. Il est autosuffisant et n'est pas lié aux autres applications du projet, de sorte que, en théorie, nous pouvons le récupérer et le replacer dans un autre projet sans aucune modification. Alors qu'un projet regroupe l'ensemble des applications. Dans notre TP **"notes"** correspond à une application. Et nous avons la possibilité d'ajouter autant d'application que nous désirons.

## 5.4 Création d'un espace de nom pour l'application "notes"

Pour éviter des conflits de nom des vues, nous allons créer un espace de noms pour l'application "notes". En effet **un espace de nom** est tout simplement une collection de noms associés à des références d'objet. C'est ce qui fait la correspondance entre le nom d'une vue et à quelle application elle se réfère.

Ainsi dans le fichier `urls.py`, nous avons déclaré un espace de nom de la manière suivante : **`app_name="notes"`**. Et dans les fichiers `.html` nous avons aussi préciser l'espace de nom de l'application. (voir l'aperçu des fichiers ci-dessous)

### eleves.html

```
<h1>La liste des élèves</h1><br>
<div>
  <table border=1>
    <thead>
      <tr>
        <td>Identifiant</td>
        <td>Nom</td>
        <td>prenom</td>
        <td>Sexe</td>
        <td>Date de Naissance</td>
        <td>Niveau</td>
        <td>Détails</td>
      </tr>
    </thead>
    <tbody>
      {% for eleve in lesEleves %}
        <tr>
          <td>{{eleve.id}}</td>
          <td>{{eleve.nom}}</td>
          <td>{{eleve.prenom}}</td>
          <td>{{eleve.sexe}}</td>
          <td>{{eleve.date_naissance}}</td>
          <td>{{eleve.niveau}}</td>
          <td><a href="{% url 'notes:eleve' eleve.id %}">Lien</a></td>
        </tr>
      {% endfor %}
    </tbody>
  </table>
</div>
```

### matieres.html

```
<h1>La liste des Matières</h1><br>
<div>
  <table border=1>
    <thead>
      <tr>
        <td>Identifiant</td>
        <td>Nom</td>
        <td>Enseignant</td>
        <td>Détails</td>
      </tr>
    </thead>
    <tbody>
      {% for matiere in lesMatieres %}
```

```
<tr>
  <td>{{matiere.id}}</td>
  <td>{{matiere.nom}}</td>
  <td>{{matiere.enseignant}}</td>
  <td><a href="{% url 'notes:matiere' matiere.id%}">Lien</a></td>

</tr>
{% endfor%}
</tbody>
</table>
</div>
```

**C'est la fin du rapport Merci !!**