

Très Facile

Tutoriels Python du débutants aux experts



 ACCUEIL

COURS PYTHON

EXERCICES & TP PYTHON AVEC SOLUTION

DÉVELOPPEMENT WEB

JAVA

TICE-MULTIMÉDIA

CONTACT

BOUTIQUE



Upload des fichiers en django: mini-système de gestion des étudiants

📅 21 JUIN 2020 👤 ADMIN 📁 LE FRAMEWORK DJANGO 💬 1 COMMENT



Mini - Système de gestion des étudiants

django



1 - Mise en place d'un formulaire d'upload au sein d'un modèle django

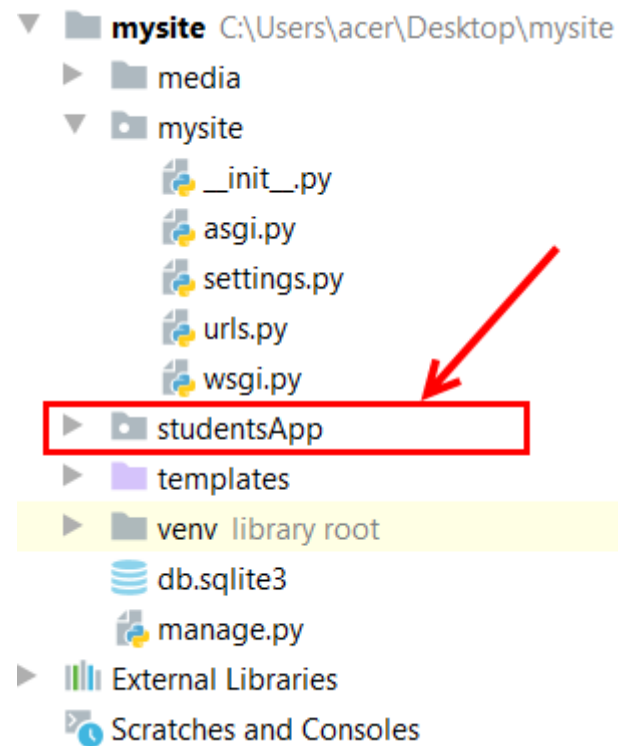


Il vous arrive souvent que vous souhaitez ajouter un champ fichier (file field) à un



d'une façon isolée ou au sein d'un modèle. Pour faire simple nous allons traiter ce dernier type.

Commençons par créer un **projet mysite** au sein duquel nous créons une application nommée **studentsApp**:



```
1 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
2 MEDIA_URL = '/media/'
```

Configurons ensuite le fichier **url.py** de l'application **studentsApp**:

urls.py

```
1 from django.contrib import admin
2 from django.urls import path
3 from uploadFile import views
4
5 urlpatterns = [
6     path('admin/', admin.site.urls),
7     path('students/', views.index),
8 ]
```

Considérons l'exemple de **modèle Students** que nous avons déjà traité dans le tutoriel [des modèles django](#), auquel on ajoute un champ **file field** qui offre aux étudiants de joindre leurs photos pendant l'inscription:





Start Your Free Trial 1

AccountsPortal Makes It Easy To
Your Invoicing, Tax, And Account

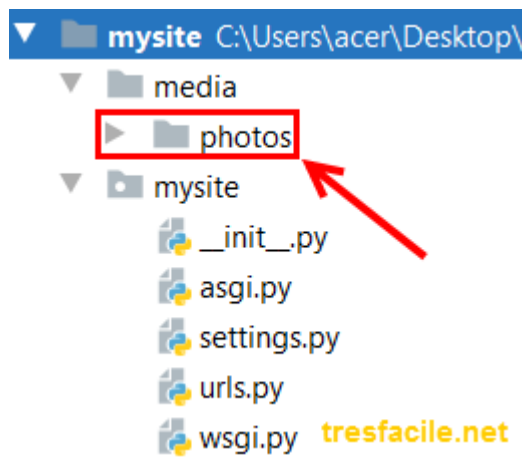
AccountsPortal

models.py

```
1  from django.db import models
2  class Students(models.Model):
3      name          = models.CharField(max_length=25)
4      email          = models.EmailField(max_length=40)
5      phone          = models.IntegerField(max_length=40)
6      section        = models.CharField(max_length=25)
7      photo          = models.FileField(upload_to = 'photos')
```



il créer manuellement ce répertoire ? Et bien ne vous inquiétez pas! **Django le créera automatiquement** lors d'exécution du script.



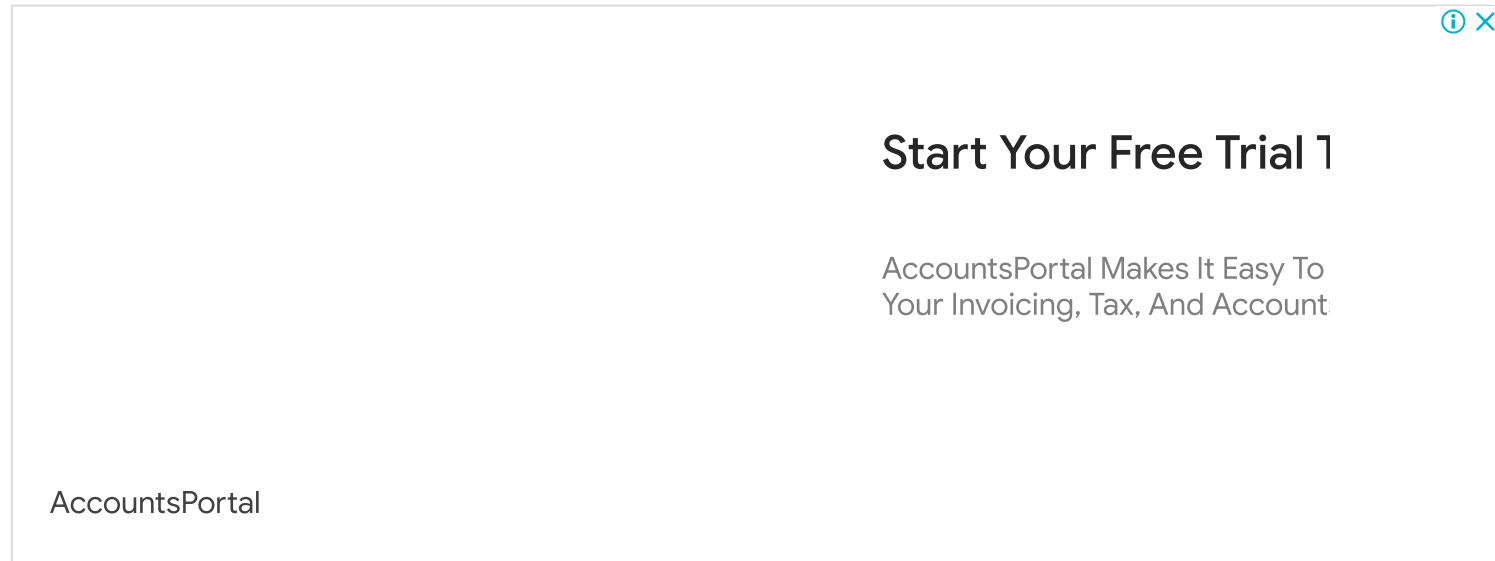
On effectue ensuite les migrations nécessaires:

```
1 | python manage.py makemigrations studentsApp
```

et ensuite :



Nous créons ensuite un fichier **forms.py** qui permettra de générer le formulaire:



forms.py

```
1 from django.forms import ModelForm
2 from studentsApp.models import Students
3
4 class StudentsForm(ModelForm):
```


Le **views**: sera ensuite créée de la même façon que dans le tutoriel précédent concernant [les modèles de django](#) à la différence d'ajout d'un **paramètre request.FILES**

views.py

```
1  from django.http import HttpResponseRedirect
2  from django.shortcuts import render
3  from .forms import StudentsForm
4
5  def index(request):
6      if request.method == 'POST':
7          form = StudentsForm(request.POST, request.FILES)
8          if form.is_valid():
9              form.save()
10             return HttpResponseRedirect('/students/')
11     else:
12         form = StudentsForm()
13     return render(request, 'index.html', {'form': form})
```

Finalemment, il nous reste qu'à créer le fichier de template **index.html**:



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8
9     <form enctype="multipart/form-data" method="post">
10     {% csrf_token %}
11     {{ form.as_p }}
12     <button type="submit" class="save btn btn-default">Save</button>
13 </form>
14
15 </body>
16 </html>
```

Sans oublier le paramètre `enctype="multipart/form-data"`



Name:

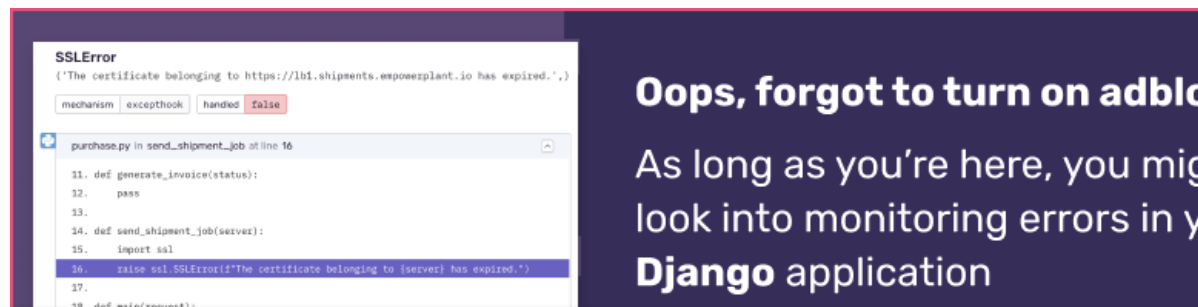
Email:

Phone:

Section:

Photo: No file selected.

2 - Application: création d'une mini-application de gestion des étudiants



Nous allons maintenant nous servir de ce qui précède pour créer un mini système de gestion des étudiants.

En se basant sur le premier paragraphe, tout est là rien que des modifications minime.

2.1 - modification au niveau du fichier urls.py:

Afin que les fichiers du répertoire médias soient accessible, il faut ajouter **MEDIA_URL** et **MEDIA_ROOT** au block des urls:

```
1 | + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

sans oublier les imports nécessaires:

```
1 | from django.conf.urls.static import static
```



```
1  from django.contrib import admin
2  from django.urls import path
3  from studentsApp import views
4  from django.conf.urls.static import static
5  from . import settings
6
7  urlpatterns = [
8      path('admin/', admin.site.urls),
9      path('students/', views.index),
10 ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

2.2 - Modification au niveau du fichier views.py

Au niveau du fichier views.py, rien que l'ajout de la variable:

```
1  | 'data' : Students.objects.all()
```



```
1 from django.http import HttpResponseRedirect
2 from django.shortcuts import render
3 from .forms import StudentsForm
4 from .models import Students
5
6 def index(request):
7     if request.method == 'POST':
8         form = StudentsForm(request.POST, request.FILES)
9         if form.is_valid():
10             form.save()
11             return HttpResponseRedirect('/students/')
12     else:
13         form = StudentsForm()
14     return render(request, 'index.html', {'form': form, 'data': data})
```

3.2 - Modification au niveau du fichier de template/index.html

Au niveau du fichier **template/index.html**, on ajoute uniquement l'affichage des champs nom, email, phone, section, photo...



```
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8
9      <h2>Gestion des étudiants</h2>
10
11      <form enctype="multipart/form-data" method="post">
12      {% csrf_token %}
13      {{ form.as_p }}
14      <button type="submit" class="save btn btn-default">Save</butt
15  </form>
16  <br />
17  <table border ='1' width="600" style="border-collapse:collaps
18      <tr><td bgcolor="#dddddd"><b>Name</b></td>
19      <td bgcolor="#dddddd"><b>Email</b></td>
```

Et maintenant en exécutant le script nous obtenons notre beau mini-système de gestion des étudiants:



Gestion des étudiants




Name:

Email:

Phone:

Section:

Photo: No file selected.

Name	Email	phone	section	photos
David	daniel@gmail.com	2125487	Info	
Natalie	natalie@gmail.com	333269587	SVT	
Najib	najib@gmail.com	212589654	Marketing	



[Younes Derfoufi](#)

[CRMEF OUJDA](#)

📊 Post Views: 4 037

[◀ Previous Post](#)

[Next Post ▶](#)

1 thought on “Upload des fichiers en django: mini-système de gestion des étudiants”

✓ Ping : [Cours Python](#) | [Cours Python Très Facile](#)



RECHERCHER





Cours Python

Les bases en Python

1. Introduction au langage Python
2. Ide Python
3. Premier programme Python
4. Les variables en Python
5. Les commentaires en Python
6. Les opérateurs Python
7. Les fonctions en Python
8. Structure If else... Python
9. For While Python
10. Les Chaines De Caractères En Python
11. Formatage des chaines Python
12. Les listes en Python
13. Les tuples en Python
14. Les dictionnaires en Python
15. Les ensembles en Python

- 19. [Héritage en Python](#)
- 20. [Les exceptions en Python](#)
- 21. [Le gestionnaire des packages pip](#)
- 22. [Serveur Web En Python](#)
- 23. [Télécharger le cours complet](#)

Les modules en Python

Nous rejoindre sur Facebook



 **Les Programmeurs**
15 714 followers

Suivre la Page Partager

 **Les Programmeurs**
il y a environ 2 semaines

<https://www.tresfacile.net/mini-logiciel-qui-conjuge-les-.../>

Saisir un verbe

Conjugaison du verbe **apprendre au passé-composé**

J'ai appris
tu as appris
il a appris
nous avons appris
vous avez appris
ils ont appris

Apprendre au passé composé

Conjuguer

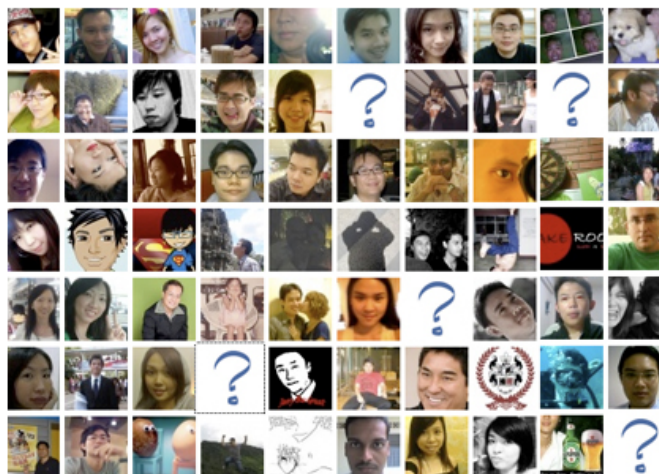
TRESFACILE.NET

Mini logiciel qui conjugue les ver...

Cours python pour débutants complet av...



Groupe Facebook



Utilisateurs En Ligne !

Se connecter

Identifiant ou adresse e-mail

Mot de passe

☐ Se souvenir de moi

SE CONNECTER

[Inscription](#)[Mot de passe oublié ?](#)

