Installations preProjet

**Plugs in VS Code :**

Python

Django

React

javaScript

**Django :**

pip install django djangorestframework

**React**

1. Installer npm

2.

**Fiche Projet Django : Création Projet**

**Etape 0 :** se mettre dans un répertoire ou voudra stocker le projet

**Etape 1 : Creation projet**

Taper python -m django startproject music\_controler, On a un dossier music\_controler avec à l’interieur un autre music controller qui correspond au projet.

**Etape 2 : aller dans le rep music controller avec cd**

**Etape 3 : création de l’app.**

python -m django startapp api

On obtient ca

Une image contenant texte, Police, capture d’écran, conception

Description générée automatiquement

Api est l’app est music\_controler est le projet, à la pace de api on peut mettre tout ce que l’on veut

**Etape 4 : Configurer le fichier settings**

Dans le fichier setting du projet sur installed\_apps rajouter :

‘api.apps.ApiConfig’

‘rest\_framework’

**Etape 5 : Test avec construction d’une view dans l’app:**

1. Dans l’app créer un fichier url.py

2.Sur le fichier url du projet :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

On ajoute un path -> ‘’ des qu’on lance le site on arrive sur cette page, la page correspond au fichier urls qu’on vient de créer dans api

Si je mets ‘home’ alors pour accéder a ma page je devrais taper …/home

3.Coder le fichier views

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Main est une fonction

4.Coder le fichier urls du d’app qui va appeler la fonction main. Ce fichier URL est appelé dans urls du projet

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

**Etape 5 : Lancer le projet (a faire des que changement sur les modèles ou database)**

Sur le terminal taper :

python manage.py makemigrations api

puis

python manage.py migrate

puis lancer le serveur

python manage.py runserver

On peut aller sur le lien et voir le site

Connecter Base Donnée

<https://www.youtube.com/watch?v=cv8Qle5EwIg>

**Fiche Projet Django : Intégration react et babble**

**Etape 1 : Créer une nouvelle app pour le front**

Lancer la commande django-admin

On a donc l’arborescence suivante :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**Etape 2 : cd frontend**

**Etape 3 : Dans frontend créer les dossiers suivants :**

-templates

-static puis dans static :

-frontend

-css

-images

-src puis dans src :

-components

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

**Etape 4 : on est dans le cd et on lance un npm project**

Taper sur le terminal : npm init -y

**Etape 5 : installer webpack de npm**

Taper sur le terminal: npm i webpack webpack-cli --save-dev

**Etape 6 : installer babble pour faciliter le site sur tous les navigateurs**

Taper sur le terminal: npm i @babel/core babel-loader @babel/preset-env @babel/preset-react --save-dev

**Etape 7 : installer react**

Taper sur le terminal: npm i react react-dom --save-dev

**Etape 8 : Installer materialUI**

Taper sur le terminal: npm -f install @material-ui/core

**Etape 9 : Autre installation babel**

Taper sur le terminal: npm -f install @babel/plugin-proposal-class-properties

**Etape 10 : react router**

Taper sur le terminal: npm -f install react-router-dom

**Etape 11 : autre materialUI**

Taper sur terminal : npm -f install @material-ui/icons

**Etape 12 : dans l’app frontend créer le fichier babel.config.json et y mettre :**

{

"presets": [

[

"@babel/preset-env",

{

"targets": {

"node": "10"

}

}

],

"@babel/preset-react"

],

"plugins": ["@babel/plugin-proposal-class-properties"]

}

**Etape 13 : dans l’app frontend new fichier webpack.config.js et y mettre**

const path = require("path");

const webpack = require("webpack");

module.exports = {

entry: "./src/index.js",

output: {

path: path.resolve(\_\_dirname, "./static/frontend"),

filename: "[name].js",

},

module: {

rules: [

{

test: /\.js$/,

exclude: /node\_modules/,

use: {

loader: "babel-loader",

},

},

],

},

optimization: {

minimize: true,

},

plugins: [

new webpack.DefinePlugin({

"process.env": {

// This has effect on the react lib size

NODE\_ENV: JSON.stringify("production"),

},

}),

],

};

**Etape 14 : modifier me package.json :**

Dans scripts rajouter 2 scripts :

  "scripts": {

  "dev":"webpack --mode development --watch",

  "build":"webpack --mode production"

  },

**Etape 15 : Ajouer les fichiers js dans src componenets et dans templates rajouter un dossier front end avec à l’interieur le fichier html index.hml**

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

**Etape 16 : remplir le fichier HTML**

<!DOCTYPE html>

<html>

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <title>Music Controller</title>

    {% load static %}

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <link

      rel="stylesheet"

      href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"

    />

    <link rel="stylesheet" type="text/css" href="{% static "css/index.css" %}"

    />

  </head>

  <body>

    <div id="main">

      <div id="app"></div>

    </div>

    <script src="{% static "frontend/main.js" %}"></script>

  </body>

</html>

**Etape 17 : Travailler views du front end pour pouvoir lancer le fichier index.html**

from django.shortcuts import render

# Create your views here.

def index(request,\*args, \*\*kwargs):

    return render(request,'frontend/index.html')

**Etape 18 : créer un URL pour lancer cette views qui va lancer le fichier HTML**

1. créer le files urls.py dans frontend

2. dans le projet : music\_controller ajouter dans les urls le path du fichier url crée dans front

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',include('api.urls')),

    path('',include('frontend.urls'))

]

3. remplir le fichier urls juste crée

from django.urls import path

from .views import index

urlpatterns = [

    path('', index)

]

**Etape 19 : Coder un component js**

1. Dans src/component créer le fichier App.js

2.Ecrire le code :

import React, { Component } from "react";

import { render } from "react-dom";

import HomePage from "./HomePage";

export default class App extends Component {

  constructor(props) {

    super(props);

  }

  render() {

    return (

      <h1>

        Testing React

      </h1>

    );

  }

}

const appDiv = document.getElementById("app");

render(<App />, appDiv);

3.Completer le fichier index.js pour le relier a App.js

import App from "./components/App";

**Etape20 :** **Ajouter le frontend aux settings du projet dans installed apps**

 'frontend.apps.FrontedConfig'

**Etape 21 : lancer le projet**

Il faut déjà lancer le serveur de l’app (bien se placer cd mucis\_controller) : python manage.py runserver

Ouvrir un autre terminal : se placer dans frontend et taper : npm run dev

Si le nav ne se met pas a jour Ctrl Shft R