

Fichiers essentiels pour un projet web/Python

Sommaire

-  Fichiers essentiels pour un projet web/Python
 - Sommaire
 - 1. `.htaccess` – Pour un serveur Apache
 - Utilité
 - Exemples d'usages
 - Exemple
 - 2. `.gitignore` – Pour Git
 - Bonnes pratiques
 - Exemple `.gitignore`
 - 3. `requirements.txt` – Pour Python
 - Exemple `requirements.txt`
 - 4. `deploy.sh` – Pour le déploiement d'un projet
 - Exemple `deploy.sh`
 - 5. `README.md`
 - 6. Fichiers de configuration et exemples
 - 6.1 Exemple de `db_connect.sample.php`
 - 6.2 Exemple de script SQL (`bad_sample.sql`)
 - 6.3 Script PHP pour importer un SQL
 - 6.4 Initialisation MySQL en Python (Flask)
 - 7. Intégration dans le projet
 - 8. Exemple de `readme.md`
 - Utilisation
 - Initialisation de la base
 - Contribution
 - Résumé

1. `.htaccess` – Pour un serveur Apache

Utilité

Fichier de configuration local pour **Apache**. Il permet de modifier le comportement du serveur dans un dossier sans toucher à la config globale.

Exemples d'usages

- Redirections et réécriture d'URL
- Contrôle d'accès (auth, IP, interdiction de fichiers)
- Forcer le HTTPS
- Gestion du cache

Exemple

```
# Rediriger tout vers HTTPS
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*?) https://%{HTTP_HOST}%{REQUEST_URI}

# Interdire l'accès à un fichier sensible
<Files ".env">
    Order allow,deny
    Deny from all
</Files>
```

⚠ À utiliser seulement si `.htaccess` est activé sur l'hébergement Apache.

2. `.gitignore` – Pour Git

Indique à Git quels fichiers/dossiers **ignorer** (non versionnés).

Bonnes pratiques

- Exclure fichiers temporaires, secrets, configs locales, dépendances, etc.
- Garder un dépôt propre et sécurisé.

Exemple `.gitignore`

```
# Python
__pycache__/
*.pyc

# Environnements virtuels
venv/

# Fichiers de config secrets
.env

# Logs et fichiers générés
*.log

# PHP/MySQL
config/db.php
*.sql
```

On peut générer un `.gitignore` adapté sur le site gitignore.io.

3. `requirements.txt` – Pour Python

Liste toutes les dépendances Python du projet. Permet une installation rapide avec :

```
pip install -r requirements.txt
```

Exemple `requirements.txt`

```
flask==2.2.5  
numpy>=1.23  
scikit-learn
```

On peut le générer avec la commande `pip freeze > requirements.txt` (attention à ne pas inclure de paquets inutiles).

4. `deploy.sh` – Pour le déploiement d'un projet

Script shell pour **automatiser le déploiement** (build, install, lancement, copie, etc.).

Exemple `deploy.sh`

```
#!/bin/bash  
echo "Déploiement en cours..."  
  
# Activer l'environnement virtuel  
source venv/bin/activate  
  
# Installer les dépendances  
pip install -r requirements.txt  
  
# Lancer l'application  
python app.py
```

Le rendre exécutable avec la commande `chmod +x deploy.sh`.

5. `README.md`

Documenter le projet : installation, usage, contribution, structure, bonnes pratiques, etc.

C'est la **vitrine** du projet.

6. Fichiers de configuration et exemples

6.1 Exemple de `db_connect.sample.php`

Ne pas versionner les vrais codes et programmes d'accès (de connexion).

On doit utiliser un fichier exemple :

```
<?php
// db_connect.sample.php
$host = 'localhost';
$dbname = 'nom_de_la_base';
$username = 'utilisateur';
$password = 'mot_de_passe';

try {
    $dsn = "mysql:host=$host;dbname=$dbname;charset=utf8mb4";
    $pdo = new PDO($dsn, $username, $password, [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    ]);
} catch (PDOException $e) {
    die("Erreur de connexion : " . $e->getMessage());
}
```

A ajouter dans `.gitignore`: (pour ignorer le vrai fichier de connexion : `db_connect.php`)

```
# Ne pas versionner les fichiers sensibles
backend_php/db_connect.php
```

6.2 Exemple de script SQL (`bad_sample.sql`)

```
-- bad_sample.sql
DROP TABLE IF EXISTS utilisateurs;

CREATE TABLE utilisateurs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(100) NOT NULL,
    email VARCHAR(150) UNIQUE NOT NULL,
    mot_de_passe VARCHAR(255) NOT NULL,
    date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

INSERT INTO utilisateurs (nom, email, mot_de_passe) VALUES
('Charles', 'charles@carnus.fr', 'motdepasse1'),
('Carnus', 'carnus@carnus.fr', 'motdepasse2'),
('Bts', 'bts@carnus.fr', 'motdepasse3');
```

6.3 Script PHP pour importer un SQL

```
<?php
// import_db.php
```

```
require 'db_connect.php';

$sqlFile = __DIR__ . '/bad_sample.sql';
if (!file_exists($sqlFile)) die("Fichier SQL non trouvé : $sqlFile");

try {
    $sql = file_get_contents($sqlFile);
    $pdo->exec($sql);
    echo "Import SQL réussi !\n";
} catch (PDOException $e) {
    die("Erreur lors de l'import SQL : " . $e->getMessage());
}
```

6.4 Initialisation MySQL en Python (Flask)

Installer la librairie :

```
pip install mysql-connector-python
```

Exemple :

```
import mysql.connector

def init_db():
    cnx = mysql.connector.connect(
        user='user', password='mdp', host='localhost', database='base'
    )
    cursor = cnx.cursor()
    cursor.execute("DROP TABLE IF EXISTS utilisateurs")
    cursor.execute("""
        CREATE TABLE utilisateurs (
            id INT AUTO_INCREMENT PRIMARY KEY,
            nom VARCHAR(100) NOT NULL,
            email VARCHAR(150) UNIQUE NOT NULL,
            mot_de_passe VARCHAR(255) NOT NULL,
            date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        )
    """)
    users = [
        ('Charles', 'charles@carnus.fr', 'motdepasse1'),
        ('Carnus', 'carnus@carnus.fr', 'motdepasse2'),
        ('Bts', 'bts@carnus.fr', 'motdepasse3')
    ]
    cursor.executemany(
        "INSERT INTO utilisateurs (nom, email, mot_de_passe) VALUES (%s, %s, %s)", users
    )
    cnx.commit()
```

```
cursor.close()
cnx.close()
print("Base initialisée !")

if __name__ == '__main__':
    init_db()
```

7. Intégration dans le projet

- **PHP :**

Copier `db_connect.sample.php` → `db_connect.php` et renseigner les vrais codes d'accès
Puis lancer :

```
php backend_php/import_db.php
```

- **Python (Flask) :**

Lancer :

```
python backend_flask/init_db.py
```

8. Exemple de `readme.md`

```
# MonProjet

Projet pour ...

## Installation

```bash
git clone https://github.com/mon_utilisateur/mon_projet.git
cd mon_projet
pip install -r requirements.txt
```

## Utilisation

```
python app.py --image "image.jpg"
```

## Initialisation de la base

- **PHP :**

Copier `db_connect.sample.php` → `db_connect.php`

puis :

```
php backend_php/import_db.php
```

- **Python :**

- 

```
python backend_flask/init_db.py
```

## Contribution

Les PR sont les bienvenues !

---

## Résumé

Fichier	Rôle	Projet concerné
<code>.htaccess</code>	Config Apache	Web
<code>.gitignore</code>	Exclusion Git	Tous
<code>requirements.txt</code>	Dépendances Python	Python
<code>deploy.sh</code>	Script de déploiement	DevOps
<code>README.md</code>	Documentation de projet	Tous

---