**Initiation à la programmation Web**

TPs : MyBand :) (1)

2024

Auteur : Emilien MICARD

# WEB1 – TPs : Initiation à la programmation Web

## *Pré-requis*

Au cours des prochains TPs, nous allons essayer de créer un site web dynamique relativement simple : **My Band**

Il s'agira d'un site vitrine pour votre groupe de Rock (comment cela vous n'avez pas de groupe de rock ? Tout le monde à un groupe de Rock !) et ainsi passer par les différentes notions de base de la programmation WEB.

Mais avant tout chose il faut mettre en place votre environnement et pour cela nous allons avoir besoin de quelques outils indispensables.

- **XAMPP pour faire tourner un serveur web en local sur votre machine**
- **Visual Studio Code comme éditeur de code. (Non obligatoire)**
- **Un navigateur Web comme Chrome par exemple**
- **Git**

**Note importante :**

C'est volontairement que je ne vous donnerai pas toutes les instructions (installation, utilisation des outils, fonctions PHP, ...) ; **ceci n'est pas un didacticiel pas à pas** mais bien un **énoncé** de ce que vous devrez réaliser.

Aussi, je vous invite à utiliser la plus extraordinaire des ressources à votre disposition :

**INTERNET ...**

En outre, vous devrez faire valider votre travail (à la mode CheckPoints) lorsque vous verrez le cachet suivant :



### Pré-requis conseillé 1 : WAMP

Pour que votre ordinateur puisse lire du PHP, il faut qu'il se comporte comme **un serveur**.

*Rassurez-vous* : vous n'avez pas besoin d'acheter une machine spéciale pour cela !

Il suffit simplement d'installer les mêmes programmes que ceux que l'on trouve sur les serveurs qui délivrent les sites web aux internautes :



- **Apache** : C'est notre serveur web. Il s'agit du plus important de tous les programmes, car c'est lui qui est chargé de délivrer les pages web aux visiteurs.
- **PHP** : Ce sera notre langage de programmation Backend.
- **MySQL** : Pour la gestion de la Base De Données.

Tous ces éléments sont libres et gratuits mais un peu fastidieux à tous installer si vous deviez le faire à la main pour chacun d'entre eux ; Nous allons utiliser un « pack » adapté à votre OS :



Normalement **WAMP est déjà installé sur les machines de l'IUT**.

Si vous préférez travailler avec votre **propre machine**, je vous laisse le soin **d'installer la distribution correspondant à votre OS**.

### TO DO (1) :



Pour l'utiliser, il vous faudra déterminer les éléments suivants :

- **Où devrez vous placer le code PHP de votre site (qui s'appellera « myband ») ?**
- **Quel fichier PHP sera le point d'entrée de votre site ?**
- **Quelle adresse devrez-vous saisir dans votre navigateur web pour accéder à votre site ?**

## Pré-requis conseillé 2 : Visual Studio Code

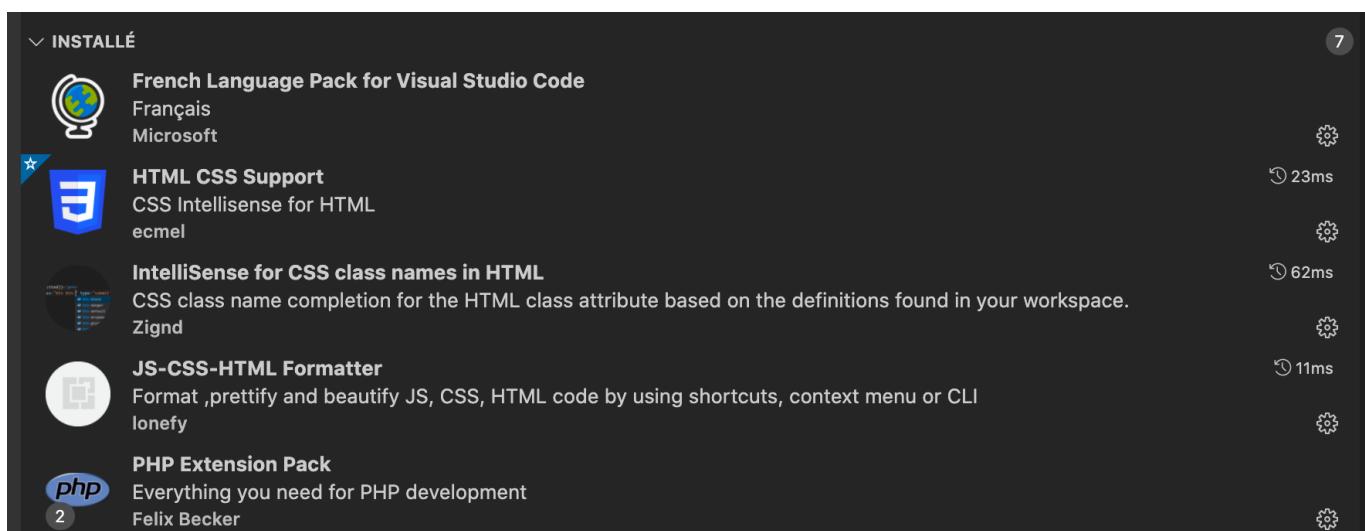


Je ne vous oblige en rien mais **VS Code** présente de nombreux **avantages** comme :

- Son faible poids
- Son faible coût (gratuit ☺ qui dit mieux ?)
- La facilité d'ajout **d'extensions** (PHP, HTML/CSS/JS, ...)

Si ce n'est pas déjà fait rendez-vous donc sur <https://code.visualstudio.com/download> pour télécharger et installer **VSCODE**.

Ensuite, je vous recommande d'installer les extensions suivantes :



## Le navigateur

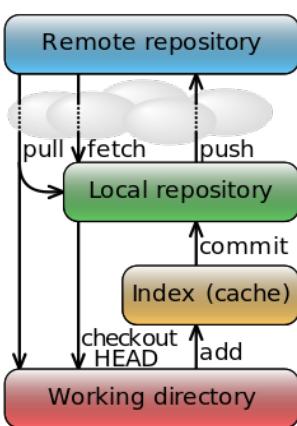


Là encore, vous faites comme vous voulez, mais si vous n'avez aucune appétence particulière, je vous recommande d'utiliser **Chrome** qui est rapide, efficace et contient plein d'outils pratiques pour le débogage.

## GIT c'est quoi ?



Une bonne pratique du développement consiste à « versionner » votre code. Et ce, afin de le sauvegarder, de travailler en collaboration et d'assurer le suivi de version de vos projets.



GIT, créé par M. Linus Thorvald lui-même, est un logiciel de gestion de version utilisable en ligne de commande ou via des outils tiers.

Il repose sur une gestion de versions décentralisé, c'est-à-dire que chaque client possède le dépôt complet en local, ainsi que l'historique intégral du projet et peut « commit » toutes les modifications sur les fichiers sources en local.

Bien sûr, on peut « push » les modifications locales sur le serveur GIT en remote (le GitHub par exemple). Avant cela, si vous êtes plusieurs à travailler sur les mêmes sources, il est recommandé de vérifier les dernières mises à jour du serveur en faisant un « fetch » et éventuellement de télécharger ces mises à jour en faisant un « pull »

Vous trouverez un guide complet pour l'utilisation de Git ici : <https://fr.wikibooks.org/wiki/Git>

## *Configuration de l'environnement*

### Configuration de votre repo GIT

Dans le cadre de ce TP et du projet nous allons utiliser les fonctionnalités Git du GitLab de l'UL. A savoir que GitLab qui est une plateforme plus riche de DevOps qui peut faire bien plus que serveur distant pour Git.

Création d'un repo sur le [GitLab](#) de l'Univ :

Rendez-vous sur <https://gitlab.univ-lorraine.fr/> et connectez-vous avec vos identifiants UL.

*Par exemple, pour moi :*

#### GitLab

##### A complete DevOps platform

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security.

This is a self-managed instance of GitLab.

LDAP	Standard
LDAP Username <input type="text" value="micard1"/>	Standard
Password <input type="password" value="***** Tu pensais avoir mon mdp ? Rêves pas trop !"/>	
<input type="checkbox"/> Remember me	
<a href="#">Sign in</a>	

Ensuite, il vous reste à créer un projet privé :

 [New project](#) > [Create blank project](#)

**Create blank project**

Project name

Project URL  Project slug

Visibility Level  Private Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.  
 Internal The project can be accessed by any logged in user except external users.  
 Public The project can be accessed without any authentication.

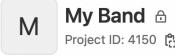
Project Configuration

Initialize repository with a README Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Enable Static Application Security Testing (SAST) Analyze your source code for known security vulnerabilities. [Learn more](#).

[Create project](#) [Cancel](#)

Et copiez le chemin du repo dans votre presse papier ([https://gitlab.univ-lorraine.fr/\\*\\*\\*\\*\\*\\*\\*\\*\\*/my-band.git](https://gitlab.univ-lorraine.fr/*********/my-band.git)) :

  
**My Band**   
 Project ID: 4150 

0 Commits 1 Branch 0 Tags 10 KB Project Storage

**Auto DevOps**  
 It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.  
[Learn more in the Auto DevOps documentation](#)

[Enable in settings](#)

main my-band / + 

**Initial commit**  
 MICARD Emilien authored just now

 README  Add LICENSE  Add CHANGELOG  Add CONTRIBUTING 

**Clone with HTTPS**  
<https://gitlab.univ-lorraine.fr/micard1>

**Open in your IDE**  
 Visual Studio Code (HTTPS)  IntelliJ IDEA (HTTPS) 

## *Configuration de l'environnement*

### Configuration générale :

Dans un terminal exécutez les commandes suivantes pour configurer les paramètres globaux at minima :

Dans Git, les contributeurs à un projet sont identifiés par leur nom et leur adresse courriel, il faut donc fournir à Git ces deux informations.

```
>> git config --global user.email nom.prenom@univ-lorraine.fr  
>> git config --global user.name "Prenom Nom"
```

Pour voir la configuration :

```
>> git config -l
```

### Initialisation du repo local :

Rendez-vous dans le répertoire du **dossier public du serveur web** :

Puis « **Clonez** » le repo de votre projet « **My Band** » (URL copiée précédemment) :

```
emilien$ git clone https://gitlab.univ-lorraine.fr/micard1/my-band.git  
Cloning into 'my-band'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.  
Checking connectivity... done.
```

Cela va copier/télécharger le contenu du dépôt remote en local – Il ne devrait y avoir qu'un fichier pour le moment README.md.

### Let's play with Git :

Sans passer trop de temps là-dessus ;), essayez de vous amuser un peu à créer/modifier/supprimer des fichiers de votre repo et de commit les changements en local.

Puis push tout cela sur le github remote

### **Voici quelques liens utiles :**

[https://fr.wikibooks.org/wiki/Git/Premiers\\_pas](https://fr.wikibooks.org/wiki/Git/Premiers_pas)

[https://fr.wikibooks.org/wiki/Git/Synchroniser\\_le\\_dépôt\\_local\\_avec\\_le\\_dépôt\\_distant](https://fr.wikibooks.org/wiki/Git/Synchroniser_le_dépôt_local_avec_le_dépôt_distant)

Et voici les commandes dont vous devriez avoir besoin :

```
git status  
git add –all  
git commit -m “comment”
```

```
git fetch  
git pull ]
```

Assez peu utiles si vous travaillez seul sur votre repo

```
git push
```

Dans cette première partie, nous allons **vérifier la configuration de notre environnement**

### TO DO (2) :



- Commencez par créer un **simple point d'entrée PHP** (*fichier .php appelé par default lorsque vous demanderez l'URL root de votre site*) qui **affichera la configuration PHP**.  
*Indice : Chercher du côté de `phpinfo()` ;*
- Vérifier que les clés de configuration suivantes sont aux bonnes valeurs :
  - `error_reporting` a la valeur **E\_ALL**
  - `display_errors` a la valeur **On***Si ce n'est pas le cas, voyez avec le prof pour la modifier 😊*



... et faire **quelques essais pour rigoler**

### TO DO (3) :



- Écrivez un script **PHP** qui **générera une première page HTML/CSS** statique au contenu souhaité.



### TO DO (4) :



- Écrivez un script **PHP** qui affichera les éléments suivants :
  - La **méthode HTTP** correspondant à votre requête (GET, POST, PUT, ...)
  - **L'URL** complète de votre requête*Indice : Chercher du côté de `$_SERVER` ;*



## Première page

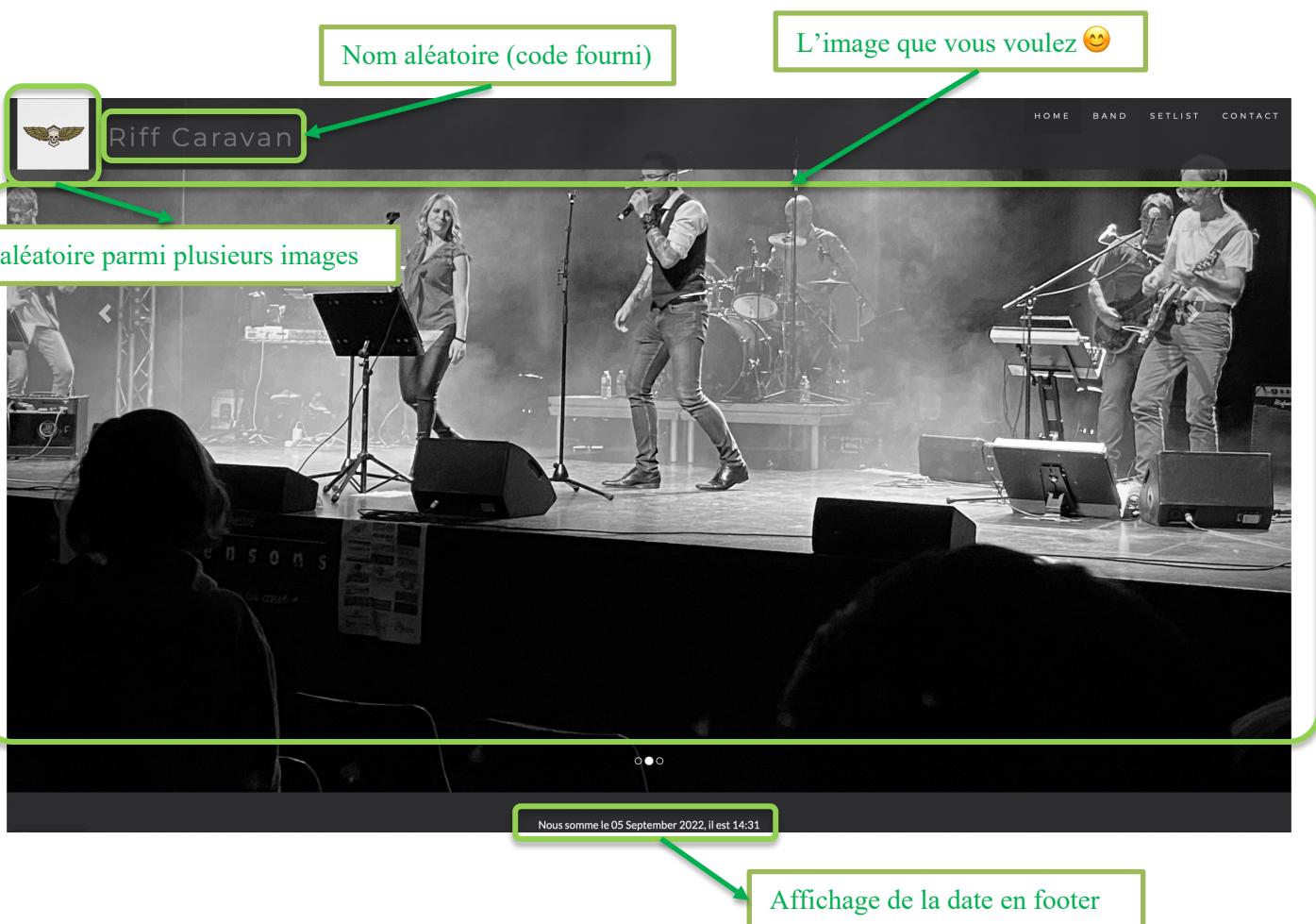
Jusqu'à présent, nous n'avons pas fait grand-chose de bien utile ; ni de bien dynamique d'ailleurs.

Vous allez donc commencer à créer la page d'accueil de votre site.

Et pour y **ajouter** un brin de **dynamisme**, vous allez faire en sorte de **générer aléatoirement** (à chaque chargement) :

- **Le nom du groupe**
- **Le logo**

Un peu dans ce style :



Pour cela vous disposerez (sur Arche) d'un fichier **band\_generators.php** contenant :

- La fonction **generate\_bandname** déjà codée et qui **retourne un nom de groupe aléatoire**
- La fonction **generate\_bandlogo** à écrire et qui devra **retourner le chemin d'une image aléatoirement choisi dans le dossier logos**

Ainsi qu'un dossier « **logos** » contenant quelques logos dans lesquelles il faudra choisir.

TO DO (5) :



- Écrivez le script **PHP** qui générera **cette page d'accueil**
- Écrivez aussi la page « **band.php** » qui reprendra le même entête et le même pied de page mais avec une description fictive ou réelle 😊 de votre Groupe. Vous ferez **pointer le lien de navigation** « Band » vers cette page.



*That's all Folks!*