Documentation utilisateur

Application MRBS

Salle & Poste informatique : Observé – Modifier – Crée

Table des matières

[Présentation de l’application 2](#_Toc72638972)

[I. Architecture et application 2](#_Toc72638973)

[II. Base de données 5](#_Toc72638974)

[III. Fonction PHP 6](#_Toc72638975)

[IV. Fonction Javascript 8](#_Toc72638976)

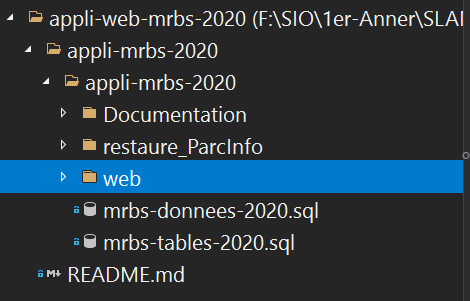
# Présentation de l’application

## Architecture et application

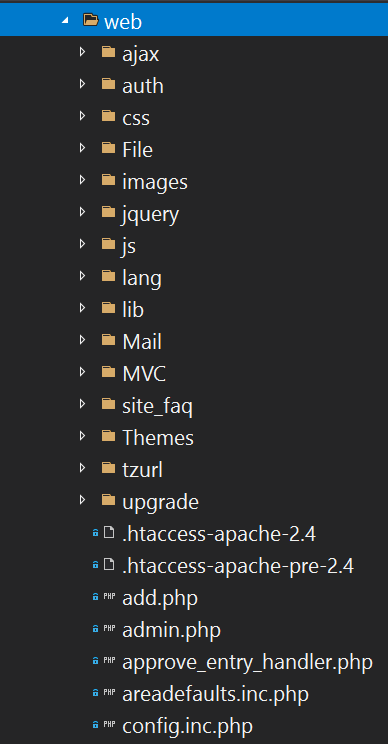
L’application web tourne grâce à l’aide du serveur **UwAmp** ( Un [WAMP](https://fr.wikipedia.org/wiki/WAMP) portable Apache/MySQL/PHP).

Elle réunit l’application **MRBS** (Meeting Room Booking System) : version 1.9.2

Avec un sous base (dossier d’architecture) au **format MVC** dédié **aux salles et poste informatique.**



le fichier « Web » qui contient tous les fichiers permettant le fonctionnement de l’application web.

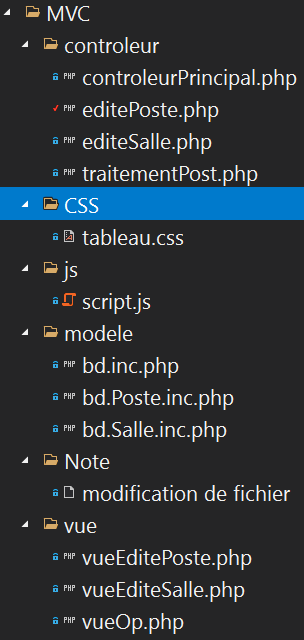
C’est également lui qui contient le dossier « MVC » qui contient l’architecture du même nom, dédié **aux salles et poste informatique.**

Le index.php du dossier MVC est appelé par le fichier :

****

Qui se trouve en dehors du dossier MVC, au niveau hiérarchique des autres fichier.php de l’application MRBS afin de pouvoir profiter de toutes les fonctionnalités offertes par l’application.

( Web/MVC.php ).

 On voit ici tous les fichiers et sous dossier contenu dans le fichier MVC :

Le dossier « **Contrôleur** » permet de rediriger entre les différents pages de l’application.

Et transformer et gérer les informations récupérées en bdd ou inscrites en brute.

Le dossier « **CSS** » permet de mettre en forme les informations.

Le dossier « **JS** » permet de gérer les appels réalisés grâce au langage Script (ici il s’agit de JavaScript ).

Le « **modèle** » gère les appels à la base de données « [mrbs\_web](http://localhost/mysql/db_structure.php?server=1&db=mrbs_web&token=1b74710fa0fea684cc598c9acefcd979" \o "Structure) ».

« **Vue** » affiche l'interface graphique des données.

## Base de données

« Mrbs\_web » est la base de données qui gérer les données de l’application.



Il s’agit d’une Base de données fonctionnant sous MySQL.

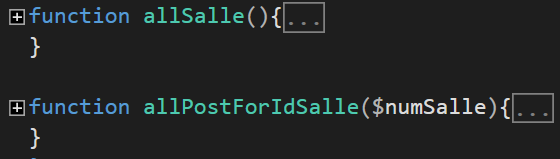


On y retrouve toutes les tables déjà existantes dans l’application MRBS.

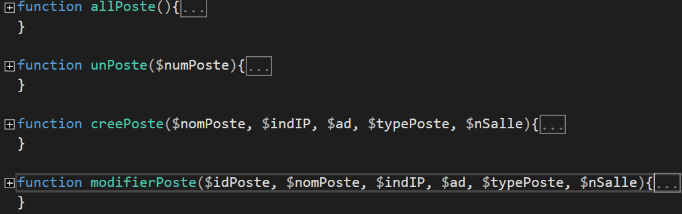
Plus la table « **mrbs\_poste** » ou sont renseigné toutes les informations sur les postes dans les salles informatiques.

## Fonction PHP

1. Fonction BDD

allSalle : récupère toutes les informations de la table « mrbs\_salle » pour les salles informatiques uniquement.

allPostForIdSalle : fait comme allSalle mais uniquement pour la salle dont l’id est $numSalle.



**allPoste :** récupère la liste de toutes les informations de la table « **mrbs\_poste** ».

**unPoste :** fait comme **allPoste**mais uniquement pour la salle dont l’id est $numPoste.

c**reePoste**: Permets d’ajouter une nouvelle entrée dans la table « **mrbs\_poste** »

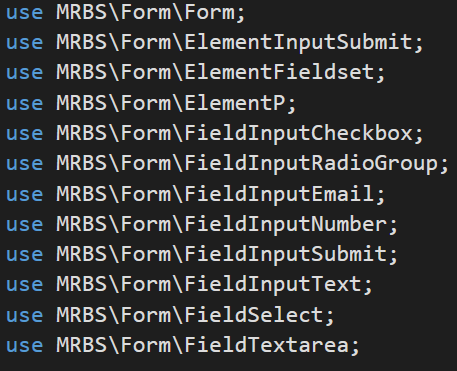
avec **id** = auto-incrémenté, le **nom** = $nomPoste, l’**IP** = $indIP, **ad** = $ad,

le **type** = $typePoste, et le **numéro de la salle** = $nSalle.

modifierPoste : fait comme **creePoste,** mais a la place de crée un poste il update juste les données. La fonction récupère aussi l’Id pour savoir quelle poste et concerné, mais il s’agit de la seule donnée inchangeable.

1. Fonction de traitement des données

Les données générées par la fonction que nous allons voir vont servir à alimenter un formulaire de donnée créée par l’application MRBS en utilisant une de ses librairies :





Cette fonction se trouvant dans « editeSalle.php » et « editePoste.php » et elle permet de gérer les champs que le formulaire va utiliser.

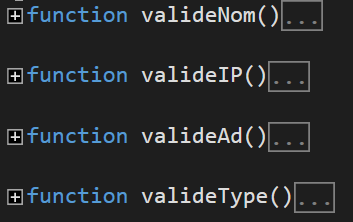
Soit avec des données inscrit en dure (ce qui est principalement le cas)

Ou des données transmisse par la variable $data

## Fonction Javascript

Setup tous les écouteurs d'événement de la page.

Charge toutes les informations du poste dans le formulaire.

Charge la liste des postes de la salle dans le tableau récapitulatif des postes de la salle

Toutes les fonctions qui vérifient la validité de chaque champ du formulaire, et permettent de l’envoyer ou non.

ValideNom : vérifie qu’il y a moins de 20 caractères.

valideIP : vérifie que l’adresse IP contient « xxx.xxx.xxx » ou les X sont remplacés par des chiffres uniquement

valideAd : vérifie qu’il y a moins de 3 caractères.

valideType : vérifie qu’il y a moins de 9 caractères.