



# Mise en Oeuvre

Calendrier de Réservation de Salles CRS

---

**Mariam BOUHAJJA**

StreamIO

date de création : 11/12/2023

dernière date de modification: 08/06/2024

## Sommaire :

<b>Sommaire :.....</b>	<b>2</b>
<b>Historique de modification :.....</b>	<b>2</b>
<b>Création de la base de données.....</b>	<b>3</b>
<b>Développement du frontend.....</b>	<b>3</b>
<b>Développement de L'API.....</b>	<b>4</b>
<b>Mise en relation entre le frontend et l'API.....</b>	<b>5</b>

## Historique de modification :

Réviseur	Date	Description
Mariem Bouhajja	11/12/2023	Création du document
Mariem Bouhajja	08/06/2024	Relecture et correction de l'orthographe

## Création de la base de données

Pour la conception de la base de données, j'ai utilisé l'outil Looping (<https://www.looping-mcd.fr/>). Cet outil se distingue par sa capacité à générer automatiquement le fichier SQL nécessaire à la création de la base de données. Il m'a suffi d'importer ce fichier dans phpMyAdmin (<https://www.phpmyadmin.net/>) pour créer ma base de données.

Remarque : Le MCD de la base de données se trouve dans le fichier d'analyse de la solution.

## Développement du frontend

Mon frontend est composé de deux parties : une page de connexion/inscription et une page principale accessible après authentification, qui comporte le calendrier de réservation.

### **Page de connexion :**

<http://127.0.0.1:5500/frontend/login.html>

Cette page offre deux fonctionnalités :

- La récupération des identifiants des utilisateurs et leur envoi à l'API. Si l'API répond avec un statut 200 (indiquant que l'utilisateur existe dans la base de données et que son mot de passe est correct), l'utilisateur sera redirigé vers la page principale.
- L'inscription de nouveaux utilisateurs en récupérant leurs informations et en les envoyant à l'API, qui les enregistrera dans la base de données.

## Page principale :

<http://127.0.0.1:5500/frontend/index.html>

Cette page permet :

- La vérification des disponibilités des salles.
- La réservation des salles après l'envoi des informations sélectionnées par l'utilisateur à l'API

## Développement de L'API

mon API comporte 5 routes principales :

- **/disponibilité** : cette route permet de récupérer les réservations déjà effectués pour une salle et une date précise
- **/réservation** : cette route permet d'effectuer une réservation en utilisant la salle, la date, l'heure et les informations sélectionnées par l'utilisateur.
- **/salles** : cette route permet de récupérer les noms des salles stockées dans la base de donnée
- **/signin** : cette route permet d'ajouter un utilisateur à la base de données et enregistre son mail, mot de passe, nom et prénom
- **/login** : cette route permet de récupérer un utilisateur de la bdd pour un mail et un mot de passe donnés

remarque : plus de détails sur les routes se trouve dans la documentation technique

## Mise en relation entre le frontend et l'API

Après la création de l'API avec node.js et le frontend avec HTML, CSS et JavaScript, j'ai relié les deux applications en utilisant des fonctions de type Fetch pour récupérer et/ou ajouter des données dans la bdd.

**Exemple** : j'utilise une fonction fetchdata qui va récupérer la date et la salle que l'utilisateur veut consulter et qui va faire appel à la route « /disponibilité ». Je récupère les réservations pour cette date et je les affiche dans le calendrier