



# Documentation technique

Calendrier de Réservation de Salles CRS

---

**Mariam BOUHAJJA**

StreamIO

date de création : 18/12/2023

dernière date de modification: 08/06/2024

## Sommaire :

<b>Sommaire :</b> .....	<b>2</b>
<b>Historique de modification :</b> .....	<b>2</b>
<b>Dictionnaire de données :</b> .....	<b>3</b>
<b>Les fonctions :</b> .....	<b>5</b>
<b>L'API :</b> .....	<b>8</b>

## Historique de modification :

Réviseur	Date	Description
Mariem Bouhajja	18/12/2023	Création du document
Mariem Bouhajja	08/06/2024	Relecture et correction de l'orthographe

## Dictionnaire de données :

Fichier	Nom	Type	Description
index.js	timeData	object	Contient les données de disponibilité des blocs de temps récupérées via l'API.
index.js	dataDate	string	Représente la date sélectionnée par l'utilisateur.
index.js	Id_salle	string	Identifiant de la salle sélectionnée.
index.js	firstDayOfMonth	number	Le premier jour du mois courant.
index.js	lastDayOfMonth	number	Le dernier jour du mois courant.
index.js	lastDateOfMonth	number	La dernière date du mois courant.
index.js	lastDateofLastMonth	number	La dernière date du mois précédent.
index.js	Id	string	Stocke l'identifiant de la salle extraite du textContent du bouton cliqué.
index.js	calendar	HTMLElement	Référence à l'élément du calendrier dans le DOM.
index.js	equipement	string	Informations sur l'équipement sélectionné.
index.js	details	string	Détails supplémentaires saisis par l'utilisateur.
index.js	date	Date	Objet de date représentant la date et l'heure actuelles.
index.js	currYear	number	Année actuelle.
index.js	currMonth	number	Mois actuel.
index.js	months	Array<string>	Tableau des noms des mois.
index.js	liTag	string	Contient les éléments <code>li</code> générés pour le calendrier.
index.js	daysTag	HTMLElement	Référence à l'élément contenant les jours du calendrier.
index.js	currentDate	HTMLElement	Référence à l'élément affichant la date actuelle.
index.js	prevNextIcon	NodeList	Liste des éléments des icônes de navigation (précédent/suivant).
index.js	blocks	NodeList	Liste des blocs de temps disponibles.

Fichier	Nom	Type	Description
<code>index.js</code>	<code>idTime</code>	<code>string</code>	Identifiant du bloc de temps sélectionné.
<code>index.js</code>	<code>bookingTime_modal</code>	<code>HTMLElement</code>	Référence à l'élément modal de sélection de temps de réservation.
<code>index.js</code>	<code>booking_modal</code>	<code>HTMLElement</code>	Référence à l'élément modal de formulaire de réservation.

Fichier	Nom	Type	Description
<code>login.js</code>	<code>signupForm</code>	<code>HTMLElement</code>	Référence au formulaire d'inscription dans le DOM.
<code>login.js</code>	<code>nom</code>	<code>string</code>	Nom saisi par l'utilisateur dans le formulaire d'inscription.
<code>login.js</code>	<code>prenom</code>	<code>string</code>	Prénom saisi par l'utilisateur dans le formulaire d'inscription.
<code>login.js</code>	<code>email</code>	<code>string</code>	Email saisi par l'utilisateur dans le formulaire d'inscription.
<code>login.js</code>	<code>mdp</code>	<code>string</code>	Mot de passe saisi par l'utilisateur dans le formulaire d'inscription.
<code>login.js</code>	<code>object</code>	<code>object</code>	Objet contenant les paramètres <code>nom</code> , <code>prenom</code> , <code>email</code> , et <code>mdp</code> pour l'inscription.
<code>login.js</code>	<code>loginForm</code>	<code>HTMLElement</code>	Référence au formulaire de connexion dans le DOM.
<code>login.js</code>	<code>emailLogin</code>	<code>string</code>	Email saisi par l'utilisateur dans le formulaire de connexion.
<code>login.js</code>	<code>mdpLogin</code>	<code>string</code>	Mot de passe saisi par l'utilisateur dans le formulaire de connexion.
<code>login.js</code>	<code>object</code>	<code>object</code>	Objet contenant les paramètres <code>email</code> et <code>mdp</code> pour la connexion.

## Les fonctions :

Fichier	Nom de la fonction	Paramètres d'entrée	Sortie	Description
index.js	<code>fetchdata</code>	<code>date</code> (string), <code>Id_salle</code> (string)	<code>timeData</code> (object)	Envoie une requête POST à l'API pour obtenir les disponibilités des salles pour une date donnée et met à jour <code>timeData</code> .
index.js	<code>fetchreservation</code>	<code>details</code> (string), <code>dateDate</code> (string), <code>Id_salle</code> (string), <code>idTime</code> (string), <code>equipement</code> (string)	None	Envoie une requête POST à l'API pour effectuer une réservation avec les détails fournis.
index.js	<code>selectRoom</code>	<code>Id</code> (string)	None	Affiche le calendrier lorsqu'une salle est sélectionnée et stocke l'ID de la salle.
index.js	<code>handleClick</code>	<code>event</code> (Event)	None	Gère le clic sur un bouton de salle, extrait et stocke l'ID de la salle sélectionnée, et appelle <code>selectRoom</code> .
index.js	<code>renderCalendar</code>	None	None	Génère dynamiquement le calendrier pour le mois en cours.
index.js	<code>handleDates</code>	None	<code>dataDate</code> (string)	Détermine et stocke la date sélectionnée par l'utilisateur et appelle <code>fetchdata</code> .
index.js	<code>selectBlock</code>	None	None	Détermine et stocke l'ID du bloc de temps sélectionné, met à jour la disponibilité des

Fichier	Nom de la fonction	Paramètres d'entrée	Sortie	Description
				blocs et appelle <code>fetchreservation</code> .
index.js	<code>openBookingTimeModal</code>	None	None	Ouvre la modal de sélection des blocs de temps pour la réservation.
index.js	<code>closeBookingTimeModal</code>	None	None	Ferme la modal de sélection des blocs de temps.
index.js	<code>openBookingModal</code>	None	None	Ouvre la modal de formulaire de réservation.
index.js	<code>closeBookingModal</code>	None	None	Ferme la modal de formulaire de réservation.
index.js	<code>handelAvailablity</code>	<code>timeData</code> (object)	None	Met à jour l'apparence des blocs de temps disponibles ou indisponibles selon les données de disponibilité.
index.js	<code>recupereInfo</code>	None	None	Récupère les informations du formulaire de réservation, vérifie leur validité et appelle <code>fetchreservation</code> pour réserver.
index.js	<code>fetch</code>	<code>url</code> (string), <code>options</code> (object)	<code>Promise</code> (response)	Envoie une requête HTTP à l'URL spécifiée avec les options fournies. Utilisée dans <code>fetchdata</code> et <code>fetchreservation</code> .

Nom de la fonction	Paramètres	Description
<code>submitSignupForm</code>	<code>event</code>	Gère la soumission du formulaire d'inscription. Envoie les données au serveur et redirige l'utilisateur.
<code>submitLoginForm</code>	<code>event</code>	Gère la soumission du formulaire de connexion. Envoie les informations d'identification au serveur et redirige l'utilisateur.

## L'API :

### Documentation de l'API de Gestion des Réservations

Cette API fournit des fonctionnalités pour la réservation des salles. Elle permet de vérifier la disponibilité des salles, de créer de nouvelles réservations, et de récupérer des informations sur les salles disponibles.

---

#### Authentification [↗](#)

##### Créer un Compte Utilisateur (Signin) [↗](#)

- **URL:** `/signin`
- **Méthode:** POST
- **Description:** Crée un nouveau compte utilisateur avec les informations fournies.
- **Paramètres du corps de la requête:**
  - `nom` (string): Nom de l'utilisateur.
  - `prenom` (string): Prénom de l'utilisateur.
  - `email` (string): Adresse e-mail de l'utilisateur.
  - `mdp` (string): Mot de passe de l'utilisateur.
- **Réponses:**
  - 201 : Succès. Le compte utilisateur a été créé avec succès.
  - 400 : L'utilisateur existe déjà.
  - 500 : Erreur serveur.

##### Authentification Utilisateur (Login) [↗](#)

- **URL:** `/login`
  - **Méthode:** POST
  - **Description:** Authentifie un utilisateur en vérifiant l'e-mail et le mot de passe fournis.
  - **Paramètres du corps de la requête:**
    - `email` (string): Adresse e-mail de l'utilisateur.
    - `mdp` (string): Mot de passe de l'utilisateur.
  - **Réponses:**
    - 201 : Succès. L'utilisateur est authentifié avec succès.
    - 401 : Utilisateur non trouvé ou mot de passe incorrect.
    - 500 : Erreur serveur.
- 

#### Endpoints [↗](#)

##### Test de Connexion [↗](#)

- **URL:** `/test`
- **Méthode:** GET
- **Description:** Endpoint de test pour vérifier la connexion au serveur.
- **Réponses:**



- 200 : Succès. Renvoie "Connecté".

### Vérifier la Disponibilité d'une Salle [↗](#)

- **URL:** `/disponibilite`
- **Méthode:** POST
- **Description:** Vérifie la disponibilité d'une salle à une date donnée.
- **Paramètres du corps de la requête:**
  - `date` (string): Date de la réservation (format YYYY-MM-DD).
  - `id` (int): Identifiant de la salle.
- **Réponses:**
  - 200 : Succès. Renvoie les heures disponibles pour la salle à la date spécifiée.
  - 500 : Erreur serveur.

### Créer une Réservation [↗](#)

- **URL:** `/reservation`
- **Méthode:** POST
- **Description:** Crée une nouvelle réservation pour une salle.
- **Paramètres du corps de la requête:**
  - `id_user` (int): Identifiant de l'utilisateur effectuant la réservation.
  - `details` (string): Détails de la réservation.
  - `date` (string): Date de la réservation (format YYYY-MM-DD).
  - `id` (int): Identifiant de la salle réservée.
  - `idTime` (int): Heure de la réservation.
  - `equipement` (string): Équipement utilisé lors de la réservation.
- **Réponses:**
  - 200 : Succès. La réservation a été créée avec succès.
  - 500 : Erreur serveur.

### Obtenir la Liste des Salles [↗](#)

- **URL:** `/salles`
  - **Méthode:** GET
  - **Description:** Récupère la liste des salles disponibles.
  - **Réponses:**
    - 200 : Succès. Renvoie la liste des numéros de salle disponibles.
    - 500 : Erreur serveur.
-