

Site des sports

Rapport de projet S8 SISN



Étudiants :

- Adèle BOURGEIX
- Gauthier HÉNON
- Antoine MIRANDE

Tuteur : François BRUCKER

Client : Jean-Luc BLANCHON

Table des matières :

Introduction au projet	2
Enjeux du site des sports	2
Les technologies	3
Front End	3
Back End	3
Fonctionnement de l'application	3
Les fonctionnalités	5
Élèves	5
Professeurs	6
Administrateur	7

1. Introduction au projet

Le site des sports est un site relatif au fonctionnement de l'enseignement à Centrale Marseille. Celui-ci doit permettre aux professeurs d'entrer les présences des étudiants dans les différents sports, aux étudiants de vérifier leurs présences, et au responsable de l'UE (Monsieur Jean-Luc Blanchon) d'évaluer les élèves au vu des présences effectuées et des notes attribuées.

Le site a été développé au cours des années par des étudiants en Java, mais quelques bugs restaient à corriger et certaines fonctionnalités à ajouter pour faciliter la notation par le responsable de l'UE. Notre travail consistait, au début de ce projet, à corriger, en Java, les bugs les plus critiques du site web puis, plus tard, en l'ajout de nouvelles fonctionnalités. Très rapidement, comme suggéré par Monsieur Brucker, un recodage en NodeJS a été envisagé.

Après une période d'un mois, qui consistait à réaliser l'application la plus basique qui remplirait la majorité des exigences de notre "cahier des charges" (explicité plus loin dans le rapport), nous devons choisir de nous pencher sur la correction en Java du site ou sur le recodage en NodeJS. Nous avons finalement décidé de nous concentrer sur un recodage entier de l'application.

Tout le code et les instructions sont disponibles sur le [Repository GitHub](#) que nous avons utilisé pour mettre en place le projet. Un fichier README.md contient toutes les instructions pour bien faire fonctionner l'application.

2. Enjeux du site des sports

Le premier enjeu de notre étude était de distinguer les différentes fonctionnalités qui sont utilisées par chaque profil d'utilisateur : élève, professeur, administrateur du site (responsable de l'UE).

Il fallait aussi distinguer les deux modes de connexion sur l'application : celle pour les membres déjà présents dans le répertoire (base de données LDAP) de l'école c'est à dire les élèves et les enseignants de sport qui enseignent à l'école et celle des enseignants extérieurs. Dans le premier cas, les utilisateurs se connectent via le CAS (Central Authentication Service) de l'école relié à sa base de données. Dans le second cas, il se connectent via une plateforme extérieure qui stocke les données relatives au professeur dans une base de données autre.

D'autre part, un élève ne doit pouvoir avoir accès qu'au suivi des séances des différents sports auxquels il participe. Cette visualisation doit être la plus claire possible pour lui permettre de réagir en cas d'erreur.

Un professeur, une fois connecté, doit pouvoir rentrer les séances de ses élèves seulement dans la semaine où se déroule le cours, ni avant, ni après. Il doit aussi pouvoir noter les élèves selon deux critères : comportement et engagement physique.

Finalement l'administrateur doit être capable de tout voir et de toute faire sur l'application.

3. Les technologies

3.1. Front End

Pour la mise en forme, nous avons utilisé le framework CSS [Bootstrap](#), dans sa version 4, nous permettant ainsi d'ajouter le minimum de lignes de code pour mettre en page l'application.

Pour le dynamisme des pages, nous avons utilisé le framework JavaScript [Angular](#), dans sa version 5.

3.2. Back End

Nous avons utilisé [Node.js](#) dans sa dernière version stable et le framework [Express.js](#) pour l'environnement côté serveur.

Nous avons aussi utilisé différents modules complémentaires:

- [Mocha](#), [Chai](#), et [Supertest](#) pour les tests
- [Sequelize](#) pour administrer nos deux bases de données PostgreSQL et SQLite

4. Fonctionnement de l'application

L'application est séparée en deux parties distinctes : celle relative au front-end (la partie visible sur le navigateur web) et la partie back-end (côté serveur).

La partie front est contenue dans le dossier **sport-front**, développée entièrement grâce à Angular et Bootstrap. L'application compilée se situe dans le dossier **ng**. Chaque fonctionnalité est servie par un **composant** particulier. Les routes pour passer d'une page à l'autre sont fixées dans le fichier **app-routing**.

La partie serveur de l'application est contenue dans le fichier **app.js**. Cette application serveur permet essentiellement 2 choses :

- L'authentification des utilisateurs grâce à l'initialisation d'une session express à la connexion. Session qui est enregistrée chez le client dans un cookie ayant une durée d'expiration finie. Le serveur sert une page d'authentification à l'utilisateur qui choisit son mode d'authentification et entre ses identifiants, si l'authentification est validée la session est créée.
- La définition des routes de notre API. Chaque route a un but précis et est appelée par l'application Angular. Par exemple : la route '/api/students/abourgeix' sert au client les informations de l'utilisateur ayant pour identifiant 'abourgeix'. Ces routes ont, pour la majorité d'entre elles, pour but de faire appel à la base de données, de sélectionner les informations demandées par le client et de les servir au format JSON dans la réponse. Si une erreur survient, les routes renvoient une erreur avec le code HTTP approprié.

Toutes les routes de l'API sont protégées par un middleware qui empêche par exemple un utilisateur non administrateur de modifier le profil d'un enseignant.

Dans le dossier **config** on retrouve les fichiers qui fixent les paramètres de configuration de l'application serveur. Le fichier **server.js** régit notamment : la connexion au CAS, la connexion au serveur LDAP, la création de la session express, la complexité du mot de passe lors d'une génération aléatoire ou encore les différents administrateurs de l'application. Les fichiers **dbTest.js** et **db.js** contiennent les paramètres de configuration des deux bases de données.

Le dossier **models** contient les fichiers qui décrivent la structure de chaque table de la base de données. Les différentes tables sont relatives à l'utilisateur, aux présences en cours, aux notes, aux semestres, aux sports et aux différentes séances de sport.

Le dossier **LDAP** contient des scripts permettant la connexion à la base de données LDAP de l'école qui contient toutes les informations des étudiants. Deux fichiers sont présents : l'un permet la recherche d'un seul étudiant à la fois, l'autre permet de renvoyer tous les élèves de l'école. Ces requêtes sont très importantes puisque les informations élèves ne sont pas stockées dans notre base de données : dès que nous avons besoin d'appeler un élève, une requête LDAP est effectuée directement.

Les parties front et back interagissent via un service Angular nommé **api** dans lequel chaque fonction fait appel à une route de notre API serveur pour recueillir ou envoyer des données grâce aux 4 méthodes HTTP usuelles.

Un fichier **README.md** est également fourni pour servir de guide du fonctionnement du site des sports.

```
# sport
Sport management app project (Node.js with Angular 5)

# Guide
How to use :

1. You'll first need to install the last stable version of NodeJS available here: https://nodejs.org/en/

2. Parameters need to be changed in files server.js and db.js of the config folder: the dbTest.js file must never be modified because it contains the config of the tests' database.

3. Once in the project's folder, you'll need to get the Node packages with the command: npm install

4. The Angular App is already compiled in the ng folder. The uncompiled Angular App is stored in the sport-front folder if you need to change anything to the front-end app.

5. You just need to launch the app with the command: node app

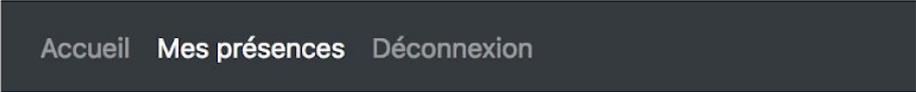
#Modules used
Angular Confirmation Popover by mattlewis92
Docs : https://mattlewis92.github.io/angular-confirmation-popover/docs/
Angular datatables
Docs : https://l-lin.github.io/angular-datatables/
```

5. Les fonctionnalités

5.1. Élèves

Les élèves se connectent via le CAS.

Ils peuvent accéder à l'onglet "Mes présences" :



Accueil **Mes présences** Déconnexion

Ils peuvent y voir leurs **présences** pour les différents **sports** et les différents **semestres**. Pour chaque sport et chaque semestre, leur **note**, si elle a été rentrée, est affichée.

S2 2017/2018		Liste des présences	
Sabre laser			
Note :			
<ul style="list-style-type: none"> • Assiduité : 8 • Comportement : 4 • Physique : 4 • Total : B 			
Show	10	entries	Search: <input type="text"/>
Date	↑↓	État	↑↓
2018-04-30		Présent	
2018-05-14		Présent	
2018-05-28		Présent	

profil d'un élève

5.2. Professeurs

Les professeurs extérieurs se connectent via "Login Exté" avec leur mot de passe.

Ils peuvent accéder aux onglets "Mon profil" et "Mes sports" :

Accueil Mon profil Mes sports Déconnexion

Dans leur profil, ils peuvent modifier leur **adresse mail** et leur **mot de passe**.

Ils peuvent **gérer les sports** qu'ils enseignent. Pour chaque sport, ils peuvent :

- **ajouter une séance** (dans les 7 derniers jours)
- **sélectionner le semestre** à visualiser
- avoir une **vue globale du semestre** dans laquelle figurent les états des élèves pour chaque séance du semestre
- accéder à la **notation** des élèves pour ce sport, pour ce semestre
- accéder à la **gestion** de chacune des séances (dans les 7 derniers jours ou si un administrateur l'a rendue accessible)

Ajouter une séance
Une date valide doit être sélectionnée

Vue globale du semestre Noter

Show entries Search:

Séance	Actions
2018-05-03	Supprimer Gestion
2018-05-10	Supprimer Gestion
2018-05-17	Supprimer Gestion

gestion d'un sport

Dans la page de notation, ils peuvent **noter** le **physique** et le **comportement**.

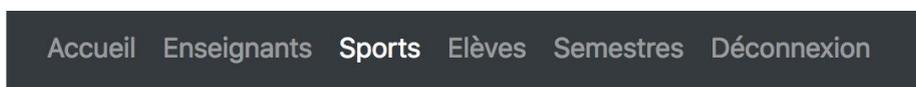
Dans la page de gestion d'un séance, ils peuvent :

- **ajouter des étudiants**
- leur attribuer un statut : **présent** ou **absent**

5.3. Administrateur

L'administrateur (Jean-Luc Blanchon) se connecte via le CAS.

Il peut accéder à tous les onglets :



Dans "Enseignants", il peut **ajouter un enseignant** et **modifier son profil**.

Dans "Sports", il peut :

- **ajouter un sport**
- modifier l'enseignant attribué à un sport
- supprimer un sport
- accéder à la **gestion** d'un des sports

Dans la page de **gestion d'un sport**, il peut, en plus de tout ce que peut faire un professeur, supprimer, **débloquer ou bloquer une séance**.

Dans une la page de **gestion d'une séance**, il peut, en plus de tout ce que peut faire un professeur, attribuer le statut "**compétition**" ou "**dispensé**" à un étudiant, et accéder au profil de l'élève (correspondant à ce que l'élève peut voir depuis son compte).

Dans la page de **notation** d'un semestre, accessible depuis la page de gestion d'un sport, il peut noter l'**assiduité** en se basant sur le **récapitulatif des présences** sous forme de ratio. Il attribue également la **note finale**.

Dans "Élèves", il peut accéder aux **profils des élèves**.

Dans "Semestres", il peut **ajouter, modifier, ou supprimer** un **semestre** (défini par une date de début et une date de fin).