

Jogenda

Moser Keith - Sans Nathan - Villalard Kelig - Desablens Maëva

IAI 2

Sommaire

Sommaire	2
Partie I : Contexte	3
1. Mission	3
2. Consigne	3
Partie II : Choix techniques	4
1. Framework BackEnd	4
2. Framework FrontEnd	4
Partie III : Mise en place de la base de données	5
1. La recherche en équipe	5
2. Production de la table	5
Partie IV : Mise en place du CRUD	6
Partie V : Cookie de session	7

Partie I : Contexte

Les jeux olympiques de Paris 2024 approchent à grands pas !

Et c'est dans cette optique que le Comité International Olympique vous confie la création d'une application web permettant aux organisateurs de gérer le planning des épreuves.

1. Mission

Notre mission était donc la conception d'un site web en Java/Javascript, permettant de gérer les différents Catégories, Sites, Disciplines, Épreuves, Sessions et type d'épreuves.

2. Consigne

Si le désigne nous était libre nous avions tout de même quelque contrainte à respecter :

- La session est caractérisée par un code de session = 3 première lettres de la discipline + XX (numérique sur 2 position)
- Pour les dates :
 - Par de recouvrement entre 2 sessions d'une même discipline.
 - Les organisateurs d'un site doivent avoir une pause d'une heure entre 2 sessions.

Le reste des consignes étant les informations importantes à trouver dans chaque table.

Partie II : Choix techniques

1. Framework BackEnd

Afin de commencer la prise en main du projet, nous avons commencé par nous renseigner sur la technologie TomCat imposée pour le projet et nous nous sommes rapidement penché sur Spring, framework populaire en Java back-end utilisant tomcat. Après plusieurs recherches, nous en avons conclu qu'il était adapté au projet et qu'on avait la capacité technique de le prendre en main. Suite à ça, nous nous sommes formés en autodidacte sur Spring, et nous avons continué tout au long du projet.

2. Framework FrontEnd

Nous avons choisi de réaliser le front sans frameworks particuliers, de peur de surcharger notre charge de travail avec l'apprentissage de trop de nouveaux concepts. C'est pourquoi nous avons opté pour du Javascript natif, simple à manipuler et basique. En outre ce côté non surchargé nous a permis d'utiliser nos connaissances en HTML et css pour pouvoir mettre facilement en place un début de front agréable.

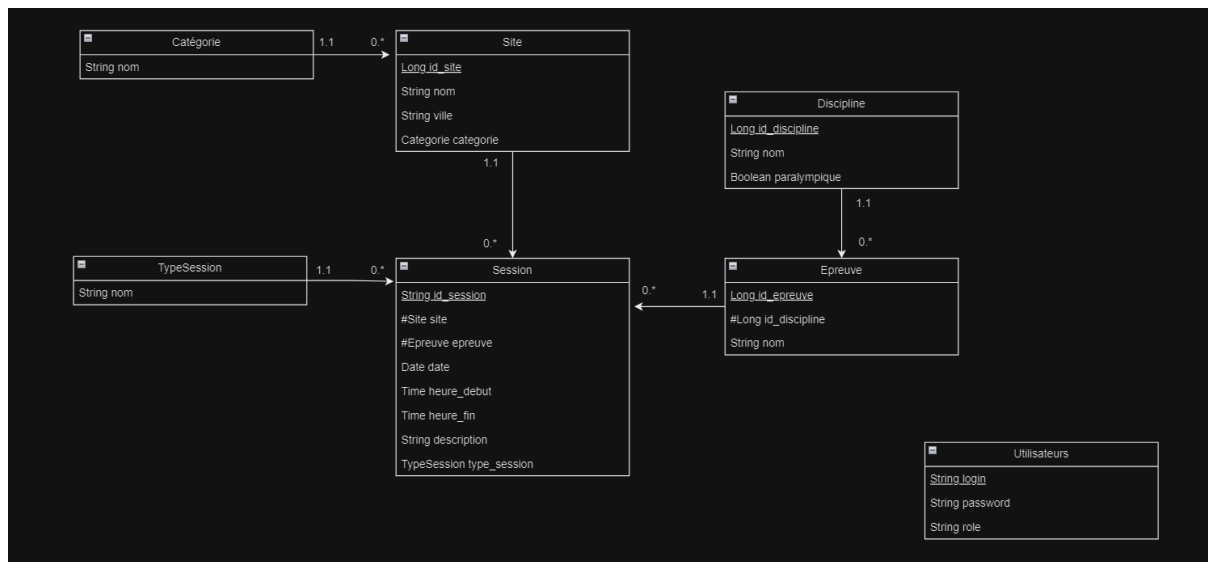
Ainsi, nous avons une partie back-end fonctionnant en partie en api rest et en serveur de routing, et une partie front native.

Partie III : Mise en place de la base de données

Pour commencer le développement de l'application, il était primordial d'avoir une base de données fonctionnelle.

1. La recherche en équipe

Nous avons commencé la conception de la base à tâtons. Dans un premier temps nous avons listé les attributs obligatoires et primordiaux pour chaque table selon nous. Puis après avoir posé à l'écrit ce qui nous semblait primordial nous avons chacun conçu la table de notre côté. Ainsi nous avons pu comparer chacune de nos tables et comparer chacune de nos choix.



2. Production de la table

Pour créer la table, nous avons utilisé les outils que nous fournissent Spring. En effet, avec Spring, il est possible de créer des modèles qui seront directement utilisés comme base pour générer la base de données ensuite. Cela nous a permis de simplifier le développement et ainsi augmenter notre productivité. Ainsi ils nous ont juste fallu transcrire nos recherches en équipe en modèle Spring pour avoir notre table créée.

Au fur et à mesure du développement, nous avons créé des scripts de génération d'enregistrements de test dans la base de données.

Partie IV : Mise en place du CRUD

La base de données étant prête, nous pouvions commencer le développement de chaque CRUD. Après s'être mis d'accord sur la base de l'organisation de nos CRUD nous avons pu chacun développer un CRUD en parallèle.

Développer les CRUD en parallèle nous a permis de gagner beaucoup de temps, sachant que les CRUD bien que liés, peuvent être créés séparément. Mais si séparer le backend du côté CRUD nous a permis de gagner du temps d'une certaine manière, c'est vers la fin que nous avons remarqué que nous aurions pu faire un CRUD assez général tous ensemble et le personifier en fonction de la table à appliquer.

Mais une fois le backend fait, il suffisait de le lier avec le front et de s'assurer d'avoir un affichage attirant l'œil.

Partie V : Cookie de session

La mise en place de Cookie de session nous a paru importante. En plus de permettre de garder une personne connectée, elle nous permet de s'assurer que l'utilisateur a accès uniquement à certaines pages.

Donc après plusieurs recherches et échange la meilleur solution a donc été la création de Token web lié à spring. C'est token fonctionne en trois temps, leur création qui permettent d'enregistrer les données de connexion et le rôle de la personne. La vérification de son état de connexion, s'assurer que l'identifiant et mot de passe n'ont pas été changés, mais aussi de vérifier son rôle. Et enfin la suppression, la suppression revient techniquement à lancer un token vide avec le même nom pour le remplacer et un temps faible pour qu'il se supprime a la chaine.