

Ecole Centrale Marseille

Projet SISN

Etablir un serveur

QIAO Lei

ZHOU Zilong

ZHU Dan

ZHANG Tianrui

2018-5-25

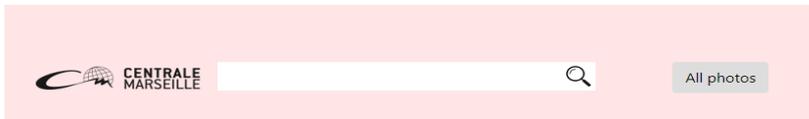
Fonctions de serveur

Avec des informations et des photos des étudiants, on a réussi à établir un serveur qui peut réaliser les fonctions suivantes:

id	numero	nom	prenom	email	photo
1	20160443	ABASQ	Maxence	maxence.abasq@centrale-marseille.fr	photo
2	20160524	ABBOUD	Baptiste	baptiste.abboud@centrale-marseille.fr	
3	20160487	AFRIT	Anouar	anouar.afrit@centrale-marseille.fr	
4	20160510	ALIC	Corentin	corentin.alic@centrale-marseille.fr	
5	20160633	AMARIR	Emilie	emilie.amarir@centrale-marseille.fr	
6	20160548	AMERIGIO	Quentin	quentin.amerigo@centrale-marseille.fr	
7	20160693	AMESTON	Vincent	vincent.amestoy@centrale-marseille.fr	
8	20160431	AMRAOU	Ilas	ilas.amraoui@centrale-marseille.fr	
9	20160600	ASSAMOUDI	Imane	imane.assamouzi@centrale-marseille.fr	
10	20160521	AUVRAY	Jeanne	jeanne.auvray@centrale-marseille.fr	
11	20160646	BABUR	Ellicot	ellicot.babur@centrale-marseille.fr	
12	20160601	BADRÉ	Aurore	aurore.badre@centrale-marseille.fr	
13	20160546	BAEHL	Antoine	antoine.baehl@centrale-marseille.fr	
14	20160499	Baldini	Pa Luigi	luigi.baldini_paulucci@centrale-marseille.fr	
15	20160660	BANCEL	Eve-Line	eve-line.bancel@centrale-marseille.fr	
16	20160598	BAO	Yuhang	yuhang.bao@centrale-marseille.fr	
17	20160552	BARRAU	Nathalie	nathalie.barrau@centrale-marseille.fr	
18	20160516	BASIRICO	Jordan	jordan.basirico@centrale-marseille.fr	
19	20160478	BASSETTE	Ines	ines.bassette@centrale-marseille.fr	
20	20160496	BATHLAN	Hugo	hugo.bathlana@centrale-marseille.fr	
21	20160504	BEGGI	Valentin	valentin.beggi@centrale-marseille.fr	
22	20160569	BERNIER	Lorenzo	lorenzo.bernier@centrale-marseille.fr	



1. Afficher les informations et les photos de tous les étudiants



Sur ce page home, on a un bouton "All photos" qui nous guide à page avec les informations et les photos de tous les étudiants.

All Records in Table user

Numero	Nom	PreNom	Email	Photo
20160443	ABASQ	Maxence	maxence.abasq@centrale-marseille.fr	
20160524	ABBOUD	Baptiste	baptiste.abboud@centrale-marseille.fr	
20160487	AFRIT	Anouar	anouar.afrit@centrale-marseille.fr	

2. Rechercher des informations d'un étudiant

Aussi sur le page home, on a un champ de recherche. D'un part, on peut faire un recherche exacte. Par exemple, quand on tape "QIAO LEI"(les lettres en majuscules ou en minuscules n'ont pas différences), on trouve le page suivant:

The screenshot shows the search results for the user LEI QIAO. At the top left is the CENTRALE MARSEILLE logo. A search bar is located at the top right. Below the search bar, the title "Search Results" is displayed in green. A table with five columns (Numero, Nom, Prenom, Email, Photo) contains one row of data. The "Photo" column contains a portrait of a woman with long black hair and glasses, wearing a dark blazer over a white collared shirt.

Numero	Nom	Prenom	Email	Photo
20160525	QIAO	LEI	lei.qiao@centrale-marseille.fr	

D'autre part, on peut faire une recherche floue. Par exemple, quand on entre "QUENTIN", on trouve le page suivant:

The screenshot shows the search results for the name "QUENTIN". The title "Search Results" is displayed in green. A table with five columns (Numero, Nom, Prenom, Email, Photo) contains four rows of data. Each row includes a small portrait photo of a man.

Numero	Nom	Prenom	Email	Photo
20160460	JEAN	Quentin	quentin.jean@centrale-marseille.fr	
20160548	AMERIGO	Quentin	quentin.amerigo@centrale-marseille.fr	
20160595	DEPAYRE	Quentin	quentin.depayre@centrale-marseille.fr	
20160573	LAMBERT	Quentin	quentin.lambert@centrale-marseille.fr	

Préparations

Afin de réaliser les fonctions ci-dessus, notre équipe a commencé à partir de zéro. Par conséquent, nous avons d'abord effectué de nombreuses préparations. D'une part,

nous devons installer des logiciels sur les ordinateurs; d'autre part, nous devons apprendre des connaissances utiles peu à peu.

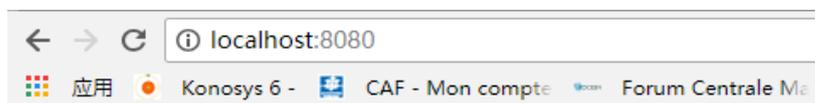
Tout d'abord, pour installer les logiciels dont nous avons besoin, nous utilisons le gestionnaire de paquets. Un gestionnaire de paquets est un outil automatisant le processus d'installation, désinstallation, mise à jour de logiciels installés sur un système informatique. Ici, nous avons choisi Scoop ou Chocolatey sur le système Windows selon nos ordinateurs. Grâce à cela, nous avons ensuite installé le kit de développement et les variables d'environnement de JAVA, ainsi que le compilateur IntelliJ IDEA. En outre, nous avons également installé le moteur de production Gradle.

En même temps, nous avons appris la langue JAVA à travers des cours en ligne sur les sites Web et écrit quelques programmes simples en utilisant JAVA. Ensuite, afin d'utiliser Spring Boot pour construire un serveur Web, nous avons appris à connaître quelques utilisations avec Spring. C'est un framework pour construire et définir l'infrastructure d'une application java et il peut faciliter les développements et les tests. Enfin, l'étude de Spring Boot nous a permis à la construction du serveur.

Réalisation

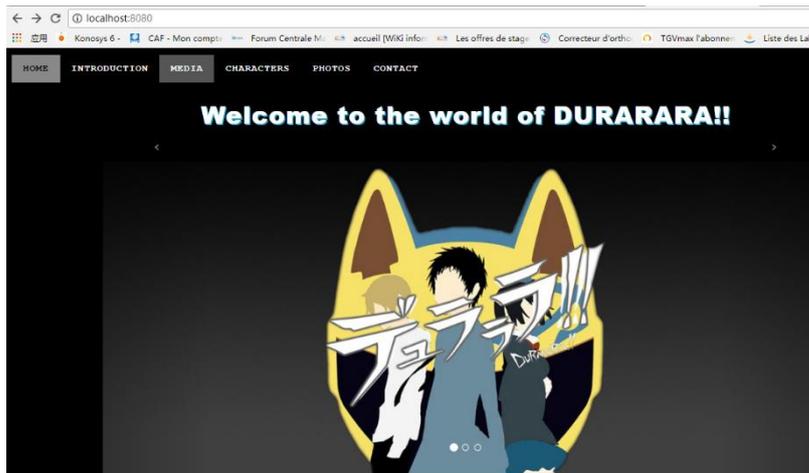
Pour la réalisation de notre application finale, on a fait plusieurs applications avec des fonctions différentes. Et on a trouvé le site <https://spring.io/guides> qui nous donne des exemples pour apprendre Spring boot rapidement.

Le premier programme en utilisant spring boot:



Cette application permet de construire un serveur avec Gradle. Grâce à cette application, on a également appris quelques utilisations avec Gradle et comment exécuter l'application et ajouter des sources pour les bibliothèques.

Une application qui peut rendre le contenu HTML:

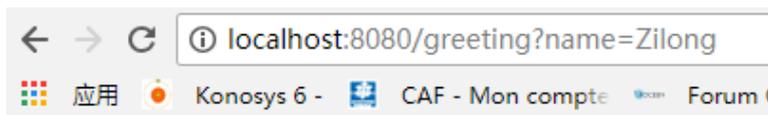


Pour rendre le contenu HTML, on a appliqué une technologie de visualisation:

Thymeleaf

Il est un outil qui réalise l'affichage des images.

Les ressources statiques comme CSS peut être servies par spring boot.

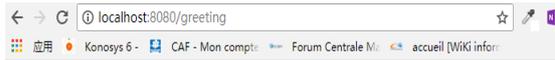


html...

Hello, Zilong!

Cette application permet aussi de recevoir des paramètres et les afficher. Comme dans cette image, on l'a donné un paramètre (name=Zilong), et elle réussi d'afficher mon prénom Zilong sur la page html.

La création d'une application Spring connectée à une base de données MySQL:



Form

Name

Age

Gender

Email

City

Result

Name: Zilong Zhou
Age: 2
Gender: m
Email: 123@com
City: Marseille

✔ Insert into MySQL successfully!

stocker des données en utilisant MySQL

All Records in Table user

NAME	Age	Gender	Email	City
qiaolei	18	M	qiaolei@123.com	Marseille
Zhou Zilong	1	f	zhou	i
Zilong Zhou	22	m	550983776@qq.com	Marseille
lea	12	f	asd	marseille
Zilong Zhou	2	m	123@com	Marseille
Zilong Zhou	2	s	s	s

Avec la technologie de ORM (object-relational mapping), on peut lier une classe java avec une table dans une base de donnée, MySQL par exemple.

On a aussi réaliser une application Spring avec une base de données SQLite.

(SQLite: Contrairement aux serveurs de bases de données traditionnels, comme MySQL, sa particularité est de ne pas reproduire le schéma habituel client-serveur mais d'être directement intégrée aux programmes. L'intégralité de la base de données est stockée dans un fichier indépendant de la plateforme.)

Avec SQLite, ce sera plus facile et pratique de stocker des informations dans un fichier.

Application finale:

Cette application fournit une fonction sur l'affichage des informations et des photos de tous les étudiants 2A et une fonction de recherche.

```
@Indexed
@Entity
public class Student {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @Field
    private Long numero;
    @Field
    private String nom;
    @Field
    private String prenom;
    @Field
    private String email;
    @Field
    private String photo;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public Long getNumero () {
        return numero;
    }
}
```

id	numero	nom	prenom	email	photo
1	20160443	ABASQ	Maxence	maxence.abasq@centrale-marseille.fr	maxence.abasq_centrale-marseille.fr
2	20160524	ABBODD	Baptiste	baptiste.abboud@centrale-marseille.fr	baptiste.abboud_centrale-marseille.fr
3	20160487	AFRIT	Anouar	anouar.afrit@centrale-marseille.fr	anouar.afrit_centrale-marseille.fr
4	20160510	Alix	Corentin	corentin.alix@centrale-marseille.fr	corentin.alix_centrale-marseille.fr
5	20160633	AMARIR	Émilie	emilie.amarir@centrale-marseille.fr	emilie.amarir_centrale-marseille.fr
6	20160548	AMERIGC	Quentin	quentin.amerigo@centrale-marseille.fr	quentin.amerigo_centrale-marseille.fr
7	20160693	AMESTOY	Vincent	vincent.amestoy@centrale-marseille.fr	vincent.amestoy_centrale-marseille.fr
8	20160431	AMRAOU	Ilias	ilias.amraoui@centrale-marseille.fr	ilias.amraoui_centrale-marseille.fr
9	20160600	assamouzi	imane	imane.assamouzi@centrale-marseille.fr	imane.assamouzi_centrale-marseille.fr
10	20160521	AUVRAY	Jeanne	jeanne.auvray@centrale-marseille.fr	jeanne.auvray_centrale-marseille.fr
11	20160646	BABUR	Elliot	elliot.babur@centrale-marseille.fr	elliot.babur_centrale-marseille.fr
12	20160601	BADRÉ	Aurore	aurore.badre@centrale-marseille.fr	aurore.badre_centrale-marseille.fr
13	20160546	BAEHL	Antoine	antoine.baehl@centrale-marseille.fr	antoine.baehl_centrale-marseille.fr
14	20160499	Baldini	Pa Luigi	luigi.baldini_paulucci@centrale-marseille.fr	luigi.baldini_paulucci_centrale-marseille.fr
15	20160660	BANCEL	Eve-Line	eve-line.bancel@centrale-marseille.fr	eve-line.bancel_centrale-marseille.fr
16	20160598	BAO	Yuhang	yuhang.bao@centrale-marseille.fr	yuhang.bao_centrale-marseille.fr
17	20160552	BARRAU	Nathalie	nathalie.barrau@centrale-marseille.fr	nathalie.barrau_centrale-marseille.fr
18	20160516	BASIRICO	Jordan	jordan.basirico@centrale-marseille.fr	jordan.basirico_centrale-marseille.fr
19	20160478	BASSETTE	Inès	ines.bassette@centrale-marseille.fr	ines.bassette_centrale-marseille.fr
20	20160496	BATTILAN	Hugo	hugo.battilana@centrale-marseille.fr	hugo.battilana_centrale-marseille.fr
21	20160504	BEGGI	Valentin	valentin.beggi@centrale-marseille.fr	valentin.beggi_centrale-marseille.fr
22	20160569	BERNIER	Lorenzo	lorenzo.bernier@centrale-marseille.fr	lorenzo.bernier_centrale-marseille.fr

C'est la classe Student qui définit des attributs (id, numéro, nom, prénom, email, nom de photo) pour stocker des informations des étudiants.

On a également créé une table dans la base de donnée SQLite pour stocker tous les informations des étudiants. Comme dans la deuxième image, on trouve tous les informations des étudiants en deuxième année.

Pour l'affichage des photos, on utilise le nom de photos dans la classe Student pour trouver les photos dans le fichier et les afficher sur la page html.

```
public List<Student> fuzzySearch(String searchTerm) {
    FullTextEntityManager fullTextEntityManager = Search.getFullTextEntityManager(entityManager);
    QueryBuilder qb = fullTextEntityManager.getSearchFactory().buildQueryBuilder().forEntity(Student.class).get();
    Query luceneQuery = qb.keyword().fuzzy().withEditDistanceUpTo(1).withPrefixLength(1).onFields("nom", "prenom")
        .matching(searchTerm).createQuery();
    javax.persistence.Query jpaQuery = fullTextEntityManager.createFullTextQuery(luceneQuery, Student.class);
}
```

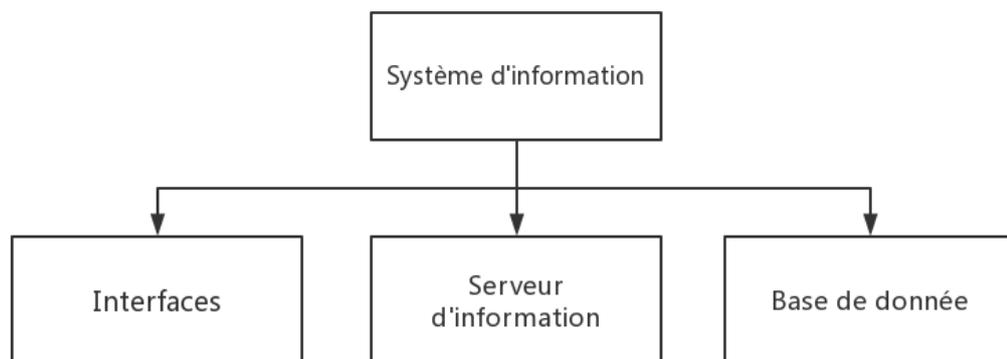
Pour la fonction de recherche, on a écrit une fonction qui s'appelle "fuzzysearch" et qui permet de chercher tous les noms ou prénoms dans la base de donnée. La fonction "fuzzysearch" retourne une liste d'étudiants. Avec cette liste d'étudiants, cela revient à l'étape dernière.

Développement futur

Comme on a réalisé le recherche exacte "fuzzysearch" dans notre service, on est en train de compléter les fonctions de recherche. D'abord, on va réaliser le recherche floue. Par exemple, pour le recherche exacte, si on code français, il donne seulement les étudiants qui s'appellent François, si on code fran, il y aura pas de résultat. Mais pour le recherche floue, si on code fran, il nous donne tous les étudiants qui s'appellent François(e). Ensuite, c'est le recherche par titre. Titre est le titre de la table dans la base

de donnée. Pour cette fonction, on peut choisir ce que on va rechercher comme nom, prénom, adresse e-mail etc. Puis, le nombre de résultat est aussi important. Tous les trois fonctions sont possible à ajouter dans notre serveur dans le futur.

Maintenant, on a créé un serveur d'information des étudiants. On a aussi codé un peu d'interfaces et la base de donnée. Alors, pour le développement futur, on peut étendre ce service comme un système informatique. Normalement, un système se divise par 3 parties, ce sont les interfaces, le serveur et la base de donnée. Comme on a codé certaine interface de HTML par java et on a utilisé SQLite comme la base de donnée. Je pense qu'on peut réussir à étendre ce serveur comme un système informatique.



Dans ce système informatique, il y a beaucoup de fonctions d'extension. Maintenant, dans notre serveur, on peut chercher et inspecter les informations personnelles des étudiants. Quand il devient un système informatique, on peut ajouter plus d'information dans le système. Par exemple, les cours, les projets, les notes etc. Et on peut séparer les utilisateurs par deux parties, l'administrateur et les professeurs. Les deux types d'utilisateurs ont compétences différentes. Finalement, ça peut devenir un système éducatif pour bien manager l'information de l'école.

Bibilographies

spring guide:

<https://spring.io/guides>

<https://spring.io/guides/gs/reactive-rest-service/>

<https://docs.spring.io/spring-boot/docs/current-SNAPSHOT/reference/htmlsingle/>

<https://springframework.guru/spring-boot-web-application-part-2-using-thymeleaf/>

<https://memorynotfound.com/category/spring-framework/spring-boot/>

<https://www.callicoder.com/categories/spring-boot/>

Thymeleaf + Spring:

<https://www.thymeleaf.org/documentation.html>

<https://www.thymeleaf.org/doc/tutorials/2.1/thymeleafspring.html>

<http://www.baeldung.com/thymeleaf-in-spring-mvc>

https://blog.csdn.net/qq_31971675/article/details/51853295

Hibernate:

<http://wiki.jikexueyuan.com/project/hibernate/>

<https://www.tutorialspoint.com/hibernate/index.htm>

http://docs.jboss.org/hibernate/orm/5.2/quickstart/html_single/#obtaining

<https://www.jmdoudoux.fr/java/dej/chap-hibernate.htm>

elasticsearch:

<https://www.elastic.co/downloads/elasticsearch>

<https://github.com/elastic/elasticsearch>

D'autres sites:

<https://www.mk Yong.com/spring-boot/spring-boot-hibernate-search-example/>

https://blog.csdn.net/qq_37355731/article/details/62222971

<http://www.imooc.com/article/19782>

maven to gradle:

<https://blog.csdn.net/qq275949034qq/article/details/40978847>

org.hibernate.search: (fuzzysearch)

<https://www.programcreek.com/java-api-examples/?class=org.hibernate.search.jpa.FullTextEntityManager&method=createFullTextQuery>