

Collaboration de microservices

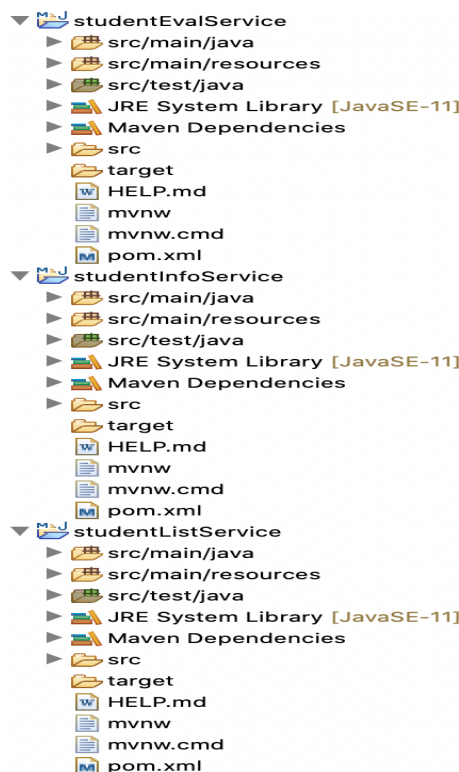
a. Compétences visées

Précédemment, vous avez vu comment créer un microservice avec Spring Boot. L'objectif maintenant est de pouvoir créer et de faire collaborer un ensemble de microservices. Comme les microservices doivent être indépendant les uns des autres, il faut créer un projet par microservices. Ainsi, vous pourrez lancer les microservices les uns indépendamment des autres, instancier certains services, ...etc.

b. Création de trois microservices

Créez les trois projets Spring Boot suivants qui vont correspondre à trois microservices :

- *StudentsListService*: Le rôle de ce microservice est d'obtenir la liste complète des étudiants d'une promotion. Ce microservice a comme entrée le code de la promotion (e.g., 3MIC) et retourne une liste des identifiants (id) des étudiants.
- *StudentInfoService*: Ce microservice permet de récupérer les informations d'un étudiant donnée. Ce microservice a comme entrée un id et retourne le nom, prénom, et la date de naissance de l'étudiant correspondant
- *StudentEvalService*: Ceci est dédié à la recherche de l'évaluation d'un étudiant. Ce microservice a comme entrée un id et retourne la moyenne de l'étudiant correspondant.



Les étapes que vous allez suivre sont comme suit :

1. Créer l'API du microservice *StudentsListService*
2. Créer l'API du microservice *StudentInfoService*

3. Créer l'API du microservice *StudentEvalService*
4. Implémenter les appels entre les microservices

Vous commencez par implémenter le modèle de *StudentsListService*. Son API attend un String (code de la promo) et retourne une liste des id des étudiants (une liste de Integer).

Créez la classe StudentIDList dans le package dédié au modèle :

```
package fr.insa.msa.studentslistservice.model;

import java.util.ArrayList;
import java.util.List;

public class StudentIDList {
    List<Integer> studentList;

    public StudentIDList() {
        this.studentList= new ArrayList<Integer>();
    }

    public List<Integer> getStudentList() {
        return studentList;
    }

    public void saddStudentToList(Integer studentId) {
        this.studentList.add(studentId);
    }
}
```

Créez le contrôleur (microservice) dans un package dédié. Dans ce microservice, vous simulez un ensemble d'identifiants d'étudiants en utilisant StudentIDList que vous avez précédemment créée.

```
package fr.insa.msa.studentslistservice.resources;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import fr.insa.msa.studentslistservice.model.*;

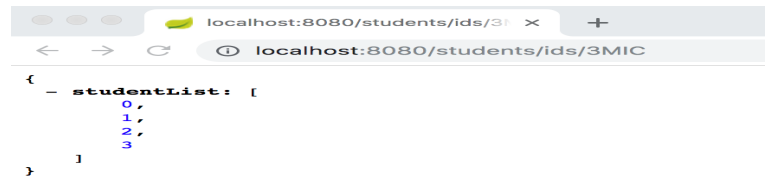
@RestController
@RequestMapping("/students")
public class StudentListResource {

    @GetMapping("ids/{idSpeciality}")
    public StudentIDList getIDStudents(@PathVariable("idSpeciality") String speciality){

        //Simulate the database using an object StudentIDList, which contains a list of integers
        StudentIDList studentsId=new StudentIDList();
        studentsId.saddStudentToList(0);
        studentsId.saddStudentToList(1);
        studentsId.saddStudentToList(2);
        studentsId.saddStudentToList(3);

        return studentsId;
    }
}
```

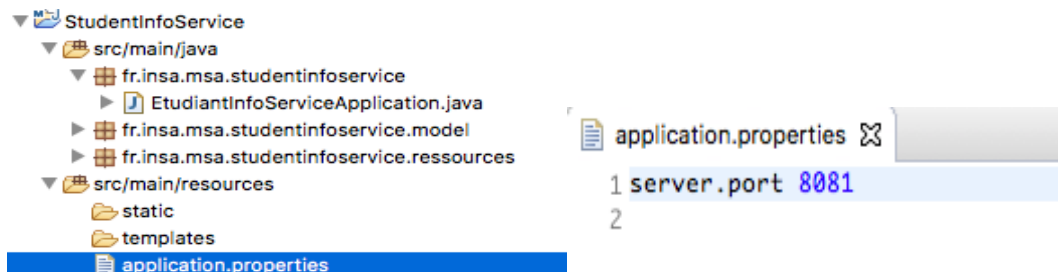
Lancez votre application et testez là.



Vous allez revenir sur ce microservice plus tard.

Maintenant, vous allez créer le microservice qui permet de récupérer les informations personnelles d'un étudiant à partir de son id.

Puisque vous aurez besoin de lancer plusieurs microservices sur votre machine (en local), vous allez devoir associer des ports de serveur différents pour chaque microservice. Rappelez-vous que pour changer le numéro de port de votre serveur, il vous suffit d'indiquer dans le fichier *application.properties* de votre projet *StudentInfoService* sous *src/main/resources* : **server.port 8081**



L'API de votre *StudentInfoResource* attend un id d'un étudiant (int) et retourne un objet *StudentInfo* avec l'id, le nom, le prénom, et la date de naissance de l'étudiant.

Créez la classe *StudentInfos* (le modèle) :

```
package fr.insa.msa.studentinfoservice.model;

public class StudentInfos {
    private int id;
    private String birthDate;
    private String lastName;
    private String firstName;

    public StudentInfos(int id, String birthDate, String lastName, String firstName) {
        this.id = id;
        this.birthDate = birthDate;
        this.lastName = lastName;
        this.firstName = firstName;
    }

    public int getId() {
        return id;
    }

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

    public String getBirthDate() {
        return birthDate;
    }

    public void setBirthDate(String birthDate) {
        this.birthDate = birthDate;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
}
```

Créez ensuite votre microservice (contrôleur) comme suit :

```
package fr.insa.msa.studentinfoservice.ressources;

import java.util.Arrays;
import java.util.List;

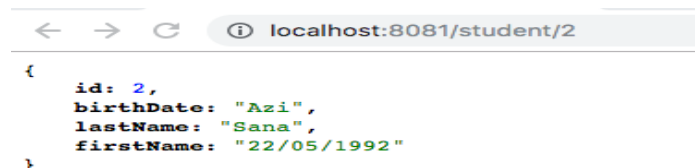
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import fr.insa.msa.studentinfoservice.model.StudentInfos;

@RestController
@RequestMapping("/student")
public class StudentInfoResource {

    @GetMapping("/{idStudent}")
    public StudentInfos getInfoEtudiant(@PathVariable("idStudent") int id){
        //Simulate the DB with a list
        List<StudentInfos> etudInfos=Arrays.asList(
            new StudentInfos(0,"Godart","Noemie","12/12/1992"),
            new StudentInfos(1,"Perrin","Ania","10/02/1993"),
            new StudentInfos(2,"Azi","Sana","22/05/1992"),
            new StudentInfos(3,"Yala","Nelia","12/06/1994")
        );
        //Get the student that corresponds to the id
        return etudInfos.get(id);
    }
}
```

Testez-le.



```
{
  id: 2,
  birthDate: "Azi",
  lastName: "Sana",
  firstName: "22/05/1992"
}
```

A la fin, implémentez le troisième microservice. Ce microservice retourne l'évaluation d'un étudiant donné en spécifiant son id. Attribuer le port 8082 à ce microservice.

Créer la classe Evaluation (le modèle):

```
package fr.insa.mas.studentEvalService.model;

public class Evaluation {
    private int idStudent;
    private float avreage;

    public Evaluation(int idStudent, float avreage) {
        this.idStudent = idStudent;
        this.avreage = avreage;
    }

    public int getIdStudent() {
        return idStudent;
    }

    public void setIdStudent(int idStudent) {
        this.idStudent = idStudent;
    }

    public float getAvreage() {
        return avreage;
    }

    public void setAvreage(float avreage) {
        this.avreage = avreage;
    }
}
```

Implémentez le microservice comme suit :

```
package fr.insa.msa.studentevalservice.resources;

import java.util.Arrays;
import java.util.List;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import fr.insa.msa.studentevalservice.model.Evaluation;

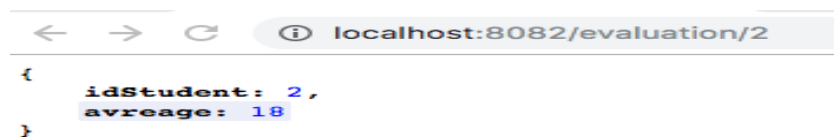
@RestController
@RequestMapping("/evaluation")
public class StudentEvaResource {

    @GetMapping("/{idStudent}")
    public Evaluation getStudenEval(@PathVariable("idStudent") int idEtud){

        //Simulate the DB with a list that contains the marks of 4 students whose id are 0, 1, 2, 3
        List<Evaluation> evallist=Arrays.asList(
            new Evaluation(0, 13F),
            new Evaluation(1, 10.65F),
            new Evaluation(2, 18F),
            new Evaluation(3, 18F)
        );

        //Return the evaluation of the specified idStudent
        return evallist.get(idEtud);
    }
}
```

Testez-le :



```
<  >  ↻  ⓘ localhost:8082/evaluation/2
{
  idStudent: 2,
  avreage: 18
}
```

Maintenant vos microservices sont prêts. Vous allez passer à l'étape suivante qui est la mise en place de la collaboration entre ces 3 microservices.

c. Collaboration des microservices

Vous allez implémenter les appels des microservices comme suit :

1. Le Microservice *StudentListResource* récupère la liste des id des étudiants d'une promo donnée.
2. Pour chaque id, il sollicite le microservice *StudentInfoResource* pour avoir les informations personnelles de l'étudiant concerné
3. Aussi, il sollicite le microservice *StudentEvalResource* pour avoir l'évaluation de l'étudiant correspondant
4. Finalement, il retourne toutes les informations récupérées pour chaque étudiant (ses informations et son évaluation)

Remarque :

Pour des raisons de simplicité, on implémente pour l'exemple les appels dans le microservice *StudentListResource*. Une autre manière plus propre est d'implémenter le scénario dans un autre microservice dédié justement à la coordination de vos microservices (orchestration).

Reprenez le microservice *StudentListResource*. Ce microservice récupère les informations des étudiants (id, date of birth, last name and first name) en appelant le deuxième microservice. Donc, vous avez aussi besoin de définir le modèle *StudentInfos*. Copiez cette classe du projet *StudentInfoService*. Puisque le microservice *StudentListResource* récupère les informations des étudiants en JSON et les déséréalise en des objets java, alors vous avez aussi besoin d'ajouter un constructeur vide comme suit :

```
package fr.insa.msa.studentslistservice.model;

public class StudentInfos {

    private int id;
    private String birthDate;
    private String lastName;
    private String firstName;

    public StudentInfos(int id, String birthDate, String lastName, String firstName) {
        this.id = id;
        this.birthDate = birthDate;
        this.lastName = lastName;
        this.firstName = firstName;
    }

    //Put an empty constructor. It is necessary for JSON to Java deserialization
    public StudentInfos(){}

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getBirthDate() {
        return birthDate;
    }
    public void setBirthDate(String birthDate) {
        this.birthDate = birthDate;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
}
```

En outre, le premier microservice appelle le 3ème microservice pour récupérer l'évaluation de chaque étudiant. Copiez la classe *Evaluation* (du projet *StudentEvalService*) et ajoutez-y un constructeur vide.

Finalement, le microservice retourne toutes les informations (informations personnelles et les évaluations). Pour cela, ajoutez au modèle la classe *Student* qui est caractérisée par les 4 attributs (id, prénom, nom, et moyenne).

```
package fr.insa.msa.studentslistservice.model;

public class Student {
    private int id;
    private String firstName;
    private String lastName;
    private float average;

    public Student(int id, String firstName, String lastName, float average) {
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.average = average;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public float getAverage() {
        return average;
    }
    public void setAverage(float average) {
        this.average = average;
    }
}
```

Retournez à *StudentListResource*. Ajoutez une nouvelle méthode *getStudents* qui retourne pour une promo donnée, la liste des étudiants inscrits (i.e., une liste d'objets *Student*). Pour les appels Rest, vous utilisez *RestTemplate*. Les commentaires dans le code suivante expliquent chaque étape.

```
@GetMapping("/all/{idSpeciality}")
public List<Student> getStudents(@PathVariable("idSpeciality") String speciality){

    //Simulate the database using a list of integers
    StudentIDList students=new StudentIDList();
    students.saddStudentToList(0);
    students.saddStudentToList(1);
    students.saddStudentToList(2);
    students.saddStudentToList(3);

    //Instantiate RestTemplate for Rest calls
    RestTemplate restTemplate =new RestTemplate();
    int i=0;
    List<Student> listStudents=new ArrayList<Student>();

    while (i<students.getStudentList().size()){
        //Call the MS to get student's information.
        //The result is deserialized into StudentInfos java object
        StudentInfos etudInfos=restTemplate.getForObject("http://localhost:8081/student/"+i, StudentInfos.class);
        //Call the MS to get the student's evaluation.
        //The result is deserialized into Evaluation java object
        Evaluation eval=restTemplate.getForObject("http://localhost:8082/evaluation/" + i,Evaluation.class);
        //Instantiate a student with his id, his first name, last name, average and store it in the list
        listStudents.add(new Student(i,etudInfos.getFirstName(),etudInfos.getLastName(),eval.getAvreage()));
        i++;
    }
    return listStudents ;
}
```

Testez votre nouvelle méthode. Vous devriez obtenir le résultat de la collaboration des trois microservices.