

Spring framework exemple 2

- Injection de dépendances
- Architecture de l'application
- REST : introduction

1. Créer un projet spring Boot

- Name : spring-exemple2
- Java 17
- Group : iir4.g5.cours
- Artifact :spring-exemple2
- Package: iir4.g5.cours.spring.exemple2
- Dépendances : Spring Web, Spring boot Devtools (l'application redémarre automatiquement après chaque modification dans l'application)

2. Créer les sous packages model, repository, services et controllers

```
spring-exemple2 [boot]
├── src/main/java
│   └── iir4.g5.cours.spring.exemple2
│       ├── iir4.g5.cours.spring.exemple2.controllers
│       ├── iir4.g5.cours.spring.exemple2.model
│       ├── iir4.g5.cours.spring.exemple2.repository
│       └── iir4.g5.cours.spring.exemple2.services
```

3. Dans le sous package model, ajouter la classe Projet

```
package iir4.g5.cours.spring.exemple2.model;

import java.util.List;

public class Projet {
    private int id;
    private String description;

    public int getId() {
        return id;
    }

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

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }
}
```

4. Ajouter l'interface ProjetRepository dans le sous package repository

```
package iir4.g5.cours.spring.exemple2.repository;

import java.util.List;

public interface ProjetRepository {
    void ajouter(Projet projet);
    void modifier(Projet projet);
    void supprimer(int idProjet);
    List<Projet> get();
    Projet get(int idProjet);
}
```

5. Ajouter dans le même package la classe ProjetRepositoryImpl qui implémente l'interface ProjetRepository

```

import java.util.List;

import org.springframework.stereotype.Repository;

import iir4.g5.cours.spring.exemple2.model.Projet;

@Repository
public class ProjetRepositoryImpl implements ProjetRepository {
    // liste des projets
    static private List<Projet> projets = new ArrayList<>();

    @Override
    public void ajouter(Projet projet) {
        projets.add(projet);
    }

    @Override
    public void modifier(Projet projet) {
        int id = projet.getId(); // id du projet à modifier
        /*
         * Recherche et modification du projet s'il existe dans la collection des
         * projets.
         */
        for (Projet p : projets)
            if (p.getId() == id) {
                p.setDescription(projet.getDescription());
                break;
            }
    }

    @Override
    public void supprimer(int idProjet) {
        // suppression du projet s'il existe
        for (Projet p : projets)
            if (p.getId() == idProjet) {
                projets.remove(p);
                break;
            }
    }

    @Override
    public List<Projet> get() {
        return projets;
    }

    @Override
    public Projet get(int idProjet) {
        for (Projet p : projets)
            if (p.getId() == idProjet)
                return p;
        return null;
    }
}

```

6. Ajouter l'interface ProjetService et la classe ProjetServiceImpl dans le sous package services

```

package iir4.g5.cours.spring.exemple2.services;

import java.util.List;

import iir4.g5.cours.spring.exemple2.model.Projet;

public interface ProjetService {
    void ajouter(Projet projet);

    void modifier(Projet projet);

    void supprimer(int idProjet);

    List<Projet> get();

    Projet get(int idProjet);
}

```

```
package iir4.g5.cours.spring.exemple2.services;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import iir4.g5.cours.spring.exemple2.model.Projet;
import iir4.g5.cours.spring.exemple2.repository.ProjetRepository;

@Service
public class ProjetServiceImpl implements ProjetService {
    @Autowired
    ProjetRepository projetRepository;

    @Override
    public void ajouter(Projet projet) {
        projetRepository.ajouter(projet);
    }

    @Override
    public void modifier(Projet projet) {
        projetRepository.modifier(projet);
    }

    @Override
    public void supprimer(int idProjet) {
        projetRepository.supprimer(idProjet);
    }

    @Override
    public List<Projet> get() {
        return projetRepository.get();
    }

    @Override
    public Projet get(int idProjet) {
        return projetRepository.get(idProjet);
    }
}
```

7. Ajouter le contrôleur ProjetController dans le sous package controllers

```

package iir4.g5.cours.spring.exemple2.controllers;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;
import iir4.g5.cours.spring.exemple2.model.Projet;
import iir4.g5.cours.spring.exemple2.services.ProjetService;

@RestController
public class ProjetController {
    @Autowired
    private ProjetService projetService;

    @GetMapping("/projets")
    public List<Projet> get() {
        return projetService.get();
    }

    @GetMapping("/projets/{id}")
    public Projet get(@PathVariable int id) {
        return projetService.get(id);
    }

    @PostMapping("/projets")
    public void ajouter(@RequestBody Projet p) {
        System.out.println("Projet " + p.getDescription());
        projetService.ajouter(p);
    }

    @PutMapping("/projets")
    public void modifier(@RequestBody Projet p) {
        projetService.modifier(p);
    }

    @DeleteMapping("/projets/{id}")
    public void supprimer(@PathVariable int id) {
        projetService.supprimer(id);
    }
}

```

8. Pour tester les services REST, il faut installer un client http comme postman, yarc (Chrome) ou RESTED (firefox), dans la suite des étapes l'extension RESTED sous firefox est utilisée

9. Lancer l'application Spring boot (run as / spring boot App)

Pour exécuter RESTED une fois installé cliquer sur l'icône de l'application située dans le coin supérieur droit de firefox :



Pour ajouter un projet :

Annotations on the screenshot:

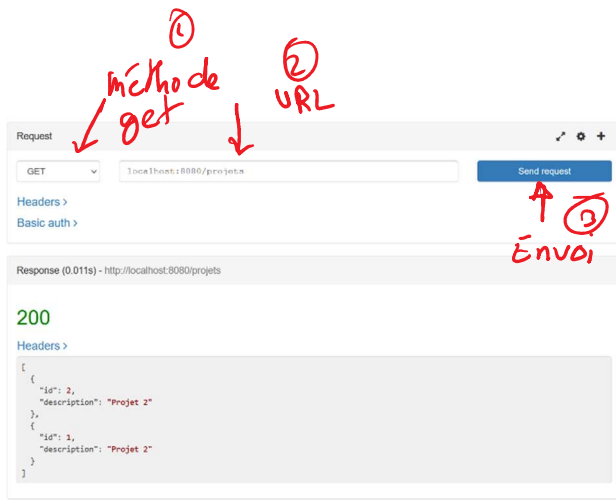
- 1: Method dropdown (POST)
- 2: URL input field (localhost:8080/projets)
- 3: Request body input field
- 4: 'le projet' (referring to the JSON body)
- 5: 'Envoyer la requête' (Send request button)

ou bien

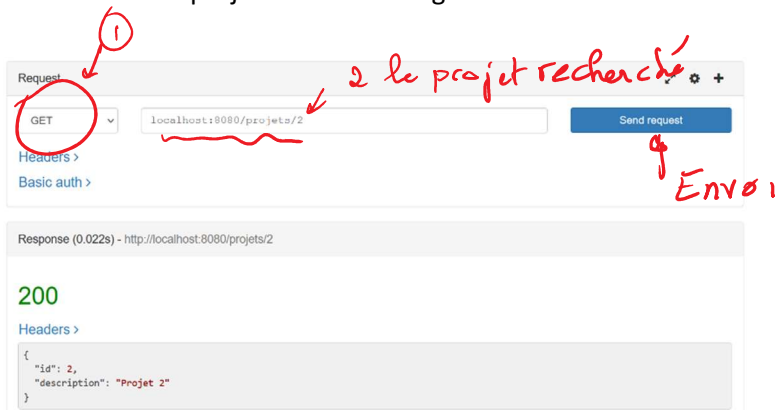
Annotation on the screenshot:

- 1: 'Ecrire le projet au format JSON' (referring to the custom request body)

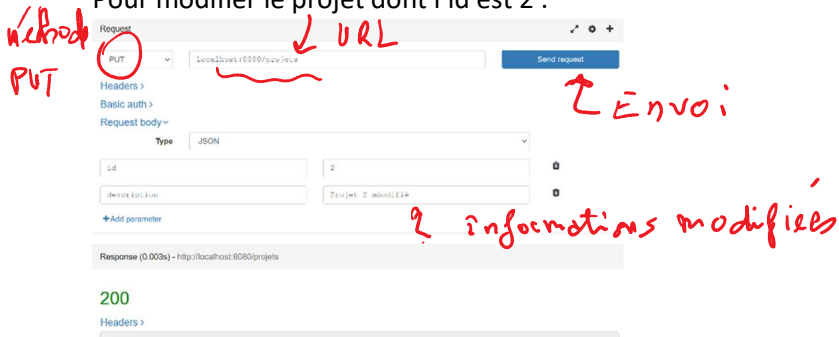
Pour afficher tous les projets



Pour obtenir le projet dont l'id est égal à 2



Pour modifier le projet dont l'id est 2 :



Pour Supprimer le projet dont l'id est 2

