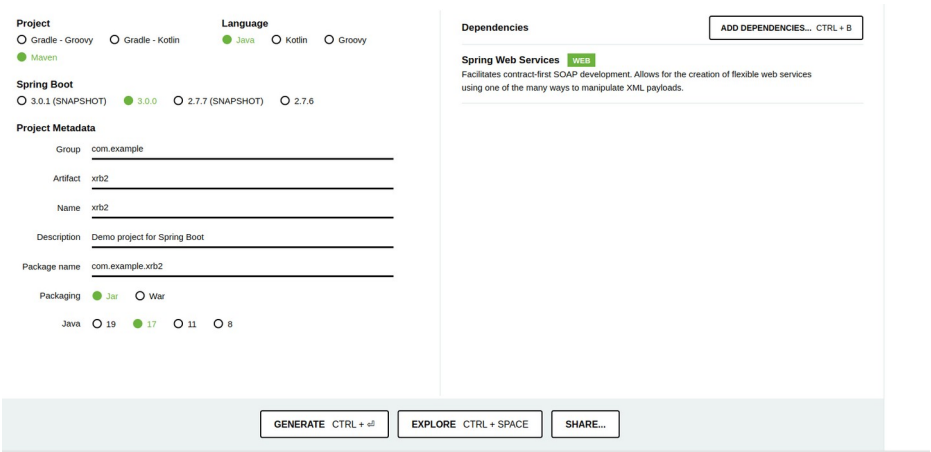
TP Web Service API REST SpringBoot

Prérequis

- jdk 1.17
- serveur
- eclipse
- le fichier de test HTML

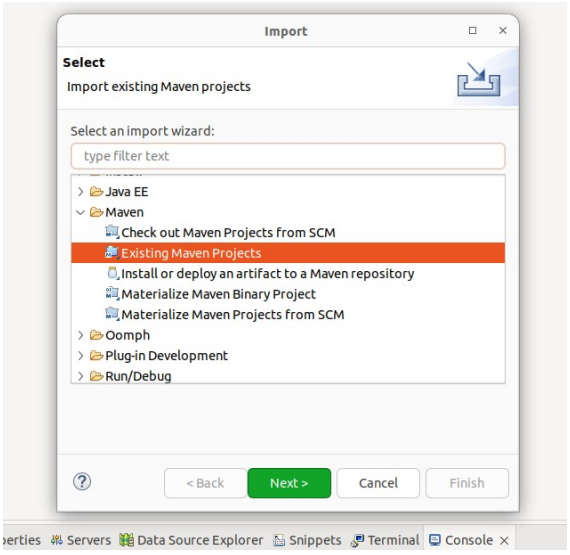
Consignes :

créez un nouveau worskpace dans eclipse
installez un server Tomcat10
créez un projet dans spring initilzr https://start.spring.io/



décompressez le zip créé par spring dans le work space eclipse

importez ce projet dans eclipse en choisissant ‘existing maven project’



configurez le 'run as'

en choisissant 'maven build'

en indiquant le goal 'spring-boot:run'

cliquer le bouton 'run'

le build doit s'exécuter

dans votre projet (là où est RestServiceApplication.java)

ajoutez la class controleur

```
import java.util.concurrent.atomic.AtomicLong;

import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.RestController;

import jakarta.servlet.http.HttpServletRequest;

@RestController
public class MyOneController {

	private final AtomicLong counter = new AtomicLong();

	@GetMapping("/bonjour")
	public Hello hello(@RequestParam(value = "name", defaultValue = "Inconnu") String name)
	{
		return new Hello(counter.incrementAndGet(), String.format( "Hello, %s!", name));
	}


	@RequestMapping(value = "/hello/**", method = RequestMethod.GET)
	@ResponseBody
	public String hello(HttpServletRequest request)
	{
	    String requestURL = request.getRequestURL().toString();
	    String arg[]          = requestURL.split("/");
	    String prenom   = arg[4];
	    String nom            = arg[5];
	    return String.format( "bonjour %s %s", prenom, nom.toUpperCase() );
	}

	//@CrossOrigin(origins = "http://localhost:8080")
	@CrossOrigin
	@GetMapping("/meteo")
	public String ajax(HttpServletRequest request)
	{
	    return "brrr !!!! fait froid...";
	}
}
```

et la class Hello

```
public class Hello {

	private final long id;
	private final String name;

	public Hello(long id, String name) {
		this.id = id;
		this.name = name;
	}

	public long getId() {
		return id;
	}

	public String getName() {
		return name;
	}
}
```

n’oubliez pas d’indiquer le package pour toutes vos classes

cliquez 'run as' et vérifiez le résultat :

http://localhost:8080/bonjour
http://localhost:8080/bonjour?name=toto
http://localhost:8080/meteo
http://localhost:8080/hello/tom/cat

PARTIE 2

ajouter une class Livre

```
public class Livre {
	private String titre;
	private String auteur;

	public Livre(String titre, String auteur) {
		this.titre = titre;
		this.auteur = auteur;
	}

	public String getTitre() {
		return titre;
	}

	public String getAuteur() {
		return auteur;
	}
}
```

ajouter un contrôleur dans lequel vous:
créez une route qui instancie un livre et le retourne en json
créez une route qui prend deux entiers en argument et retourne leur addition