

API REST avec Spring MVC (sans Spring Boot)

Concepts Clés

DispatcherServlet

Routeur central qui dirige les requêtes HTTP vers les contrôleurs appropriés.
`org.springframework.web.servlet.DispatcherServlet`

HandlerMapping

Détermine quel contrôleur gère une requête entrante. Ex: `@RequestMapping("/path")`

Concepts Clés

HttpMessageConverter

Transforme les objets Java en JSON/XML et vice versa. Ex: `MappingJackson2HttpMessageConverter`

Exemple Illustratif : Service Métro

Fichier `src/main/webapp/WEB-INF/web.xml`

```
<!-- Configuration du contexte Spring -->
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/applicationContext.xml</param-value>
</context-param>

<!-- Initialisation du contexte -->
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

<!-- Configuration MVC -->
<servlet>
  <servlet-name>mvc</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/mvc-servlet.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

```
<!-- Mapping des URLs -->
<servlet-mapping>
    <servlet-name>mvc</servlet-name>
    <url-pattern>/api/*</url-pattern>
</servlet-mapping>
```

Fichier `src/main/webapp/WEB-INF/mvc-servlet.xml`

```
<!-- Activation des annotations -->
<mvc:annotation-driven/>

<!-- Scan des contrôleurs -->
<context:component-scan base-package="com.example.controller"/>
```

Contrôleur Météo

`src/main/java/com/example/controller`

```
@RestController
public class WeatherController {

    @GetMapping("/weather")
    public Weather getWeather() {
        return new Weather("Paris", 22.5, "Ensoleillé");
    }
}
```

Modèle de Données

`src/main/java/com/example/model`

```
public class Weather {
    private String city;
    private double temperature;
    private String condition;

    // Constructeur + Getters
}
```

Flux de Traitement

1. Requête `GET /api/weather` arrive au serveur

2. `DispatcherServlet` trouve le contrôleur via `@GetMapping`
3. Méthode `getWeather()` exécutée
4. Conversion automatique en JSON via Jackson
5. Réponse `200 OK` avec :

```
{
  "city": "Paris",
  "temperature": 22.5,
  "condition": "Ensoleillé"
}
```

Test avec cURL

`curl -H "Accept: application/json" http://localhost:8080/api/weather`

Avantages/Inconvénients

Configuration Manuelle !

- ✖ Contrôle total des composants
- ✖ Compréhension approfondie de Spring
- ✖ Plus verbeux que Spring Boot

Avantages/Inconvénients

Déploiement WAR !

- ✖ Compatible avec tout serveur Servlet
- ✖ Isolation des dépendances
- ✖ Nécessite rebuild après modifications