UAX FP

Universidad
Alfonso X el Sabio

Desarrollo con API

MÓDULO: ACCESO A DATOS

PROFESOR: CARLOS RUFIÁNGEL GARCÍA

CONCEPTOS CLAVE

¿Qué es una API?

API (Application Programming Interface) es un conjunto de reglas y definiciones que permiten que diferentes programas se comuniquen entre sí.

Una API **expone funcionalidades o datos** para que otros programas los usen, sin necesidad de conocer los detalles internos de su implementación.

CONCEPTOS CLAVE

¿Qué es @GetMapping?

@GetMapping es una anotación de Spring Boot que **define un endpoint** que responde a **solicitudes HTTP GET**.

Se usa dentro de una clase controladora (@RestController) para especificar una URL que responderá con datos.

¿Qué es una API REST?

API REST (Representational State Transfer) es un estilo de arquitectura para diseñar servicios web.

Usa métodos HTTP estándar para interactuar con recursos, como:

- GET: Obtener datos
- POST: Enviar datos
- PUT: Actualizar datos
- DELETE: Eliminar datos

CONCEPTOS CLAVE

¿Qué es "Parse" y en concreto parseJson?

"Parse" significa "analizar" o "convertir" datos de un formato a otro.

En nuestro caso, parseJson(String json) toma un JSON en formato texto y lo convierte en objetos Java.

UAX FP

Universidad
Alfonso X el Sabio

LABORATORIO PRÁCTICO

LABORATORIO

Configuración del Proyecto en IntelliJ

- **Abrir IntelliJ IDEA** y seleccionar **"New Project"**.

- **Elegir "Maven" o "Java"** según la configuración que prefieras.

- **Si usas Maven**, asegúrate de que el pom.xml incluya la dependencia.

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <version>2.7.0</version>
  </dependency>
</dependencies>
```

LABORATORIO

Crear el Servidor API con Spring Boot

- **Crear el paquete y clase principal** (DemoApplication.java).

- **Añadir la anotación @RestController** para indicar que es una API REST.

- **Implementar un @GetMapping("/personas")** que lea datos de un XML (personas.xml).

- **Colocar el archivo personas.xml en src/main/resources/**.

```
<personas>
  <persona>
    <id>1</id>
    <nombre>Juan</nombre>
    <edad>30</edad>
  </persona>
  <persona>
    <id>2</id>
    <nombre>Ana</nombre>
    <edad>25</edad>
  </persona>
</personas>
```

LABORATORIO

Crear el Cliente en Java

- **Crear una clase Persona** con id, nombre y edad.
- **Crear Main.java** con HttpClient para consumir la API.
- **Implementar parseJson(String json)** para extraer datos.
- **Ejecutar primero el servidor Spring Boot** (DemoApplication).
- Abrir el navegador y visitar http://localhost:8080/api/personas para ver el JSON.
- **Ejecutar Main.java y ver los resultados en consola**.

LABORATORIO

Pruebas y Validación

•

• Ejecutar el cliente Java y verificar que se imprimen los datos correctamente.

• Probar con datos diferentes en personas.xml para validar cambios.