Distributed system fundamentals & technologies web in Java

LE Do Thanh Long
Software Engineer

Université Paris Dauphine PSL
Master 2 Miage IF - Apprentissage

Course 2 : Introduction Java Spring

Roadmap today:

- Introduction Spring framework
- Some feature of Spring
- Don’t forget about Maven
- Let’s “Hello World” in Java Spring
- Let’s discovery API
- Connect to database ?
- Let’s CRUD
- Conclusion

What is Spring Framework

By wikimedia : The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform.

-> hmm !

Question today

Create a service to manage Student and Book.

A book is only rent by a student.

One student can rent many books.

What is Spring Framework

It's just : **open-source application framework that provides infrastructure support for developing Java applications**.

We can :

- Make it easy with connecting to database
- Make it security
- Make it like a batch
- Make it like web
- Make it like web services
- Make it like stream
- Make it more and more …

What is Spring Framework

It's just : **open-source application framework that provides infrastructure support for developing Java applications**.

We can :

- Make it easy with connecting to database - Spring JPA
- Make it security - Spring security
- Make it like a batch - Spring batch
- Make it like web - Spring MVC
- Make it like web services - Spring … guess what??
- Make it like stream - Spring Kafka
- Make it more and more …

What is Spring Framework

It's just : **open-source application framework that provides infrastructure support for developing Java applications**.

We can :

- Make it easy with connecting to database - Spring JPA
- Make it security - Spring security
- Make it like a batch - Spring batch
- Make it like web - Spring MVC
- Make it like web services - Spring … guess what??
- Make it like stream - Spring Kafka
- Make it more and more …

Let's boot everything

-> Spring Boot

Why Spring ?

- Because you love java
- Because you will create a java enterprise application
- Because it will boot everything you need
- Because it’s flexible, modular, supported, innovate
- Hmm because Java Developer is well-paid ;)

Don’t forget about Maven

Maven is a popular open-source build tool developed by the Apache Group to build, publish, and deploy several projects at once for better project management. The tool provides allows developers to build and document the lifecycle framework

Don’t forget about Maven

Maven focuses on the simplification and standardization of the building process, taking care of the following:

- Builds
- Documentation
- Dependencies
- Reports
- SCMs
- Distribution
- Releases
- Mailing list

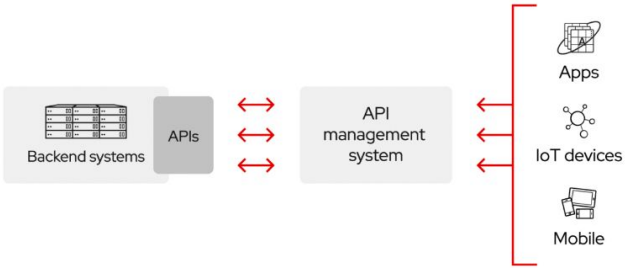
“Hello World” with Spring Boot

Let’s wrap into a demo !!

API topic

API stands for application programming interface, which is a set of definitions and protocols for building and integrating application software.

- Product or service communicate with other products
- Help business and IT teams collaborate
- Connect to the infrastructure & share the data with customers and other external users

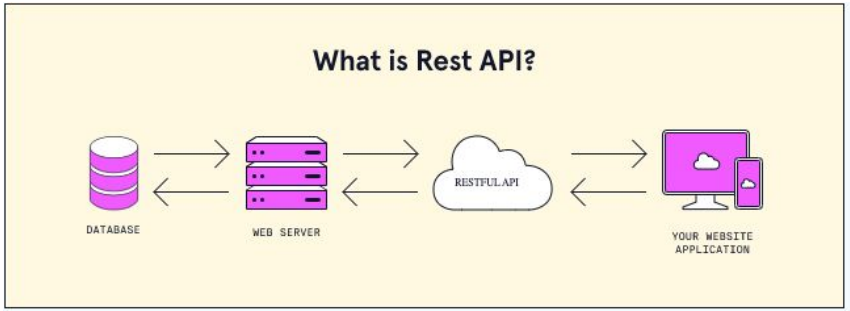


REST API

A REST API is an API that conforms to the design principles of the REST, or representational state transfer architectural style.

REST API strictly operates on the web concept of Client and Server

Rest API is a kind of web-service which stores and retrieves necessary data

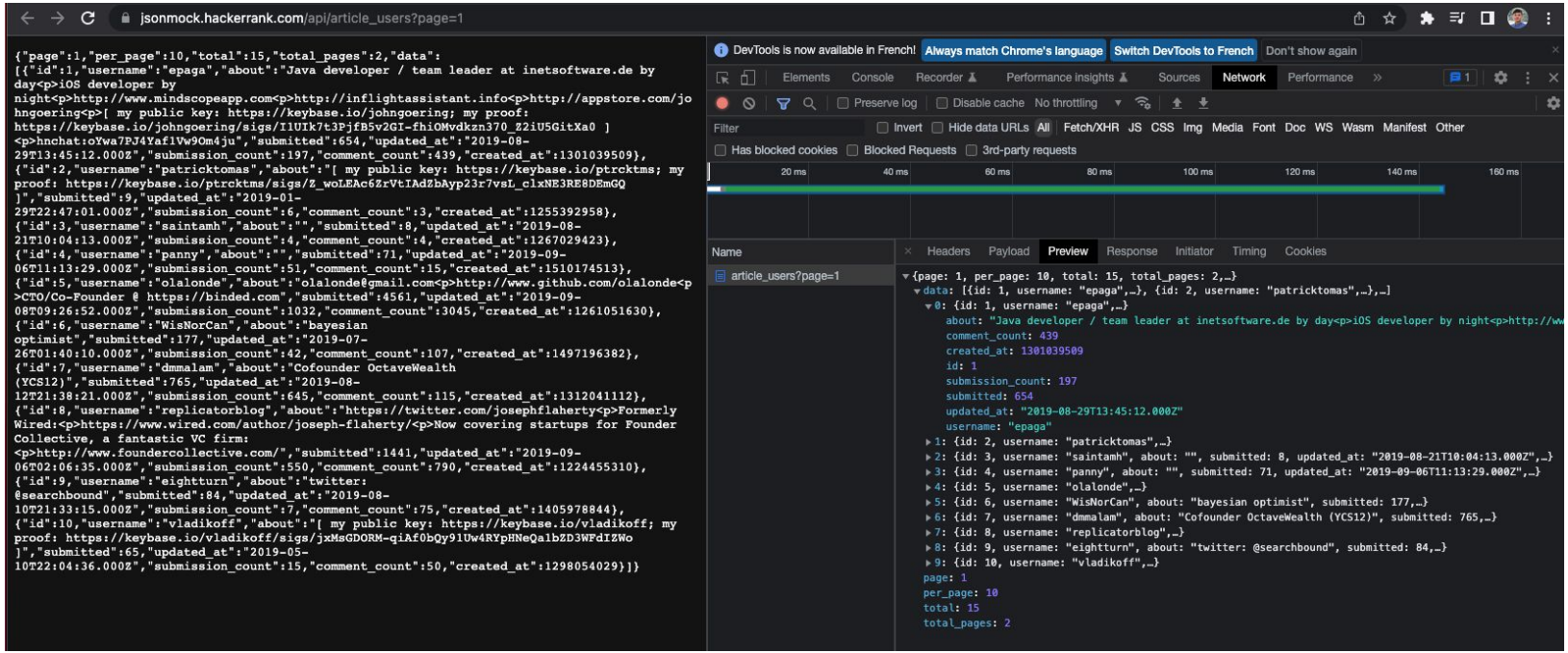


REST API

The basic :

- Use of a uniform interface (UI): guide the behavior of components
- Client-server based: the uniform interface separates user concerns from data storage concerns
- Stateless operations: the client request should contain all the information necessary to respond
- RESTful resource caching: cacheable or not.Caching improves performance
- Layered system: the requesting client need not know whether it’s communicating with the actual server, a proxy, or any other intermediary.
- Code on demand: servers can also send executable codes to the client (often XML or JSON)

REST API example



What are REST APIs used for?

- Cloud applications
- Cloud services
- Web use

The Benefits of Using REST APIs

- Scalability
- Flexibility & Portability
- Independence

Challenges of REST APIs

- Versioning
- Authentification
- Security
- Multiple Requests and Unnecessary Data
- etc…

How to connect to database ?

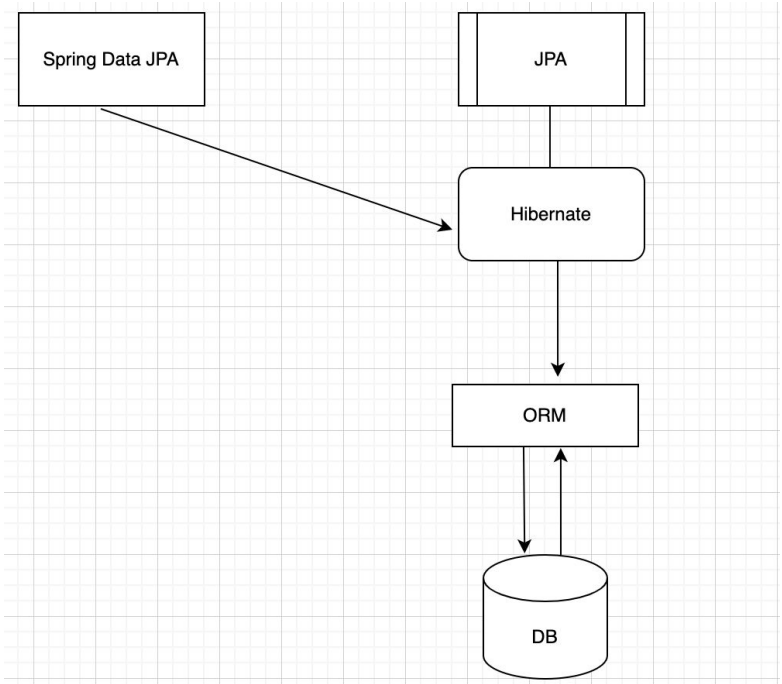
Add spring data jpa dependency into pom.xml

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>


  <dependency>
          <groupId>org.postgresql</groupId>
          <artifactId>postgresql</artifactId>
          <scope>runtime</scope>
  </dependency>
```

How to connect to database ? (application.yml)

```yaml
server:
 port: 52001
 servlet:
   context-path: /

spring:
 jpa:
   generate-ddl: true
   hibernate:
     ddl-auto: none
   show-sql: false
   properties:
     hibernate.default_schema: public
     hibernate.format_sql: true
     hibernate.jdbc.time_zone: UTC
 datasource:
   driver-class-name: org.postgresql.Driver
   url: jdbc:postgresql://localhost:5432/ifmiage
   username: postgres
   password:
```

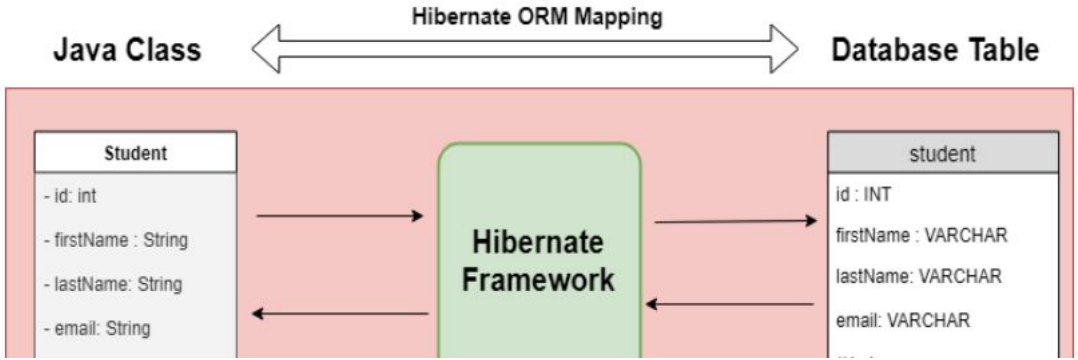
Relation



ORM - Object-Relational Mapping

ORM is object relational mapping, that maps the data in the database to the Java Class which is called an Entity

The most popular ORM framework is **Hibernate**



Why Hibernate ?

- Open Source
- High performance
- HQL (Hibernate query language)
- Caching
- Lazy loading

JPA - Java Persistence API

The Java Persistence API provides a specification for persisting, reading, and managing data from your Java object to relational tables in the database.

A link between an object-oriented model and a relational database system

Hibernate is the standard implementation of JPA

Spring Data JPA provides a set of interfaces for easily creating data access repositories.

Spring Data JPA is not exact JPA !!

JPA is a standard for defining the persistence layer

Spring Data JPA is a sub-project (abstraction) under the Spring Framework which allows Spring applications to integrate with JPA

Remember, Spring Data JPA always requires the JPA provider such as Hibernate

@Entity

Class Student

```
public class Student {

    private Integer id;

    private String firstName;

    private String lastName;

    private String email;

    private SexEnum sex;

}
```

@Entity

```java
@Entity
@Table(name = "student")
@Data
public class Student {
    @Id
    @GeneratedValue
    private Integer id;
    @Column(name = "first_name")
    private String firstName;
    @Column(name = "last_name")
    private String lastName;
    @Column(name = "email")
    private String email;
    @Column(name = "age")
    private Integer age;
}
```

@Entity

```java
@Entity
@Table(name = "course")
public class Course {
    @Id
    @GeneratedValue
    private Integer id;
    @Column(name = "code")
    private String code;
    @Column(name = "name")
    private String name;
}
```

Repository

```
public interface StudentEntityRepository  extends JpaRepository<Student,Integer> {

}
```

Repository

```
public interface StudentEntityRepository  extends JpaRepository<Student,Integer> {

   public List<Student> findAll();

   public Optional<Student> findById(Integer id);

   public Optional<Student> findByEmail(String email);

}
```

JPQL + native query

We can write the query code following two approaches:

JPQL

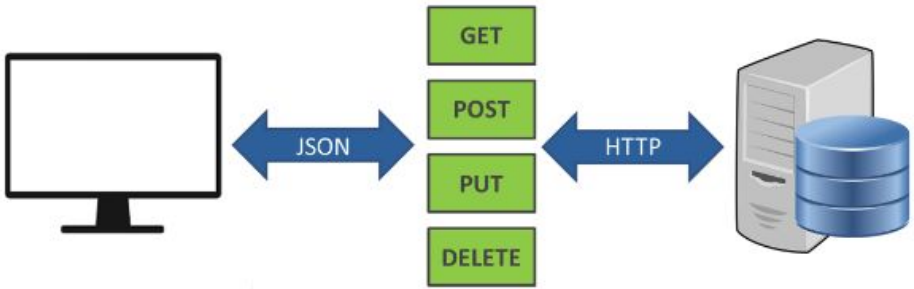
```
@Query("SELECT u FROM User u WHERE u.status = 1")
Collection<User> findAllActiveUsers();
```

Native query

```
@Query(
  value = "SELECT * FROM USERS u WHERE u.status = 1",
  nativeQuery = true)
Collection<User> findAllActiveUsersNative();
```

CRUD

CRUD = Create + Read + Update + Delete



@RestController

@RestController annotation is a special controller used in RESTful Web services

Add @RequestMapping to map the resource

Ex: @RequestMapping(“/student”) -> will map all requests starting by “/student”

@RestController

@RestController annotation is a special controller used in RESTful Web services

For CRUD, we can use with @GetMapping, @PostMapping, PutMapping, @DeleteMapping

Some annotations

@PathVariable:

```
@DeleteMapping("students/email/{email}")

public void delete(@PathVariable String email){  // students/email/david@gmail.com
```

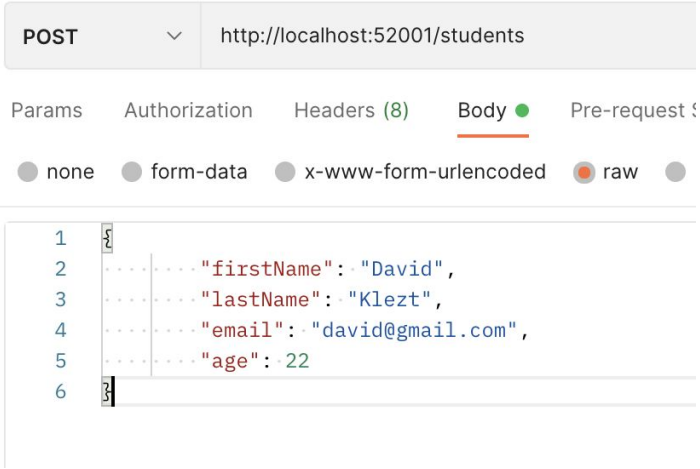
Some annotations

@RequestBody

```
@PostMapping("students")

public Student save(@RequestBody Student s)
```

Some annotations

@RequestBody



Some annotations

@RequestParam

```
@PutMapping("students")

public Student update(@RequestBody Student s, @RequestParam String email)

// students?email=david@gmail.com
```

Exemple

```java
@RestController()
@AllArgsConstructor
public class StudentApi {
    private final StudentService studentService;
    private final StudentEntityRepository studentEntityRepository;

    @GetMapping("students")
    public List<Student> getAll(){
        return studentService.getAll();
    }
    @PostMapping("students")
    public Student save(@RequestBody Student s){
        return studentEntityRepository.save(s);
    }

    @PutMapping("students")
    public Student update(@RequestBody Student s, @RequestParam String email){ // students?email=david@gmail.com
        Optional<Student> student = studentEntityRepository.findById(s.getId());
        student.ifPresent(value -> value.setEmail(email));
        return studentEntityRepository.save(student.orElse( other: null));
    }

    @DeleteMapping("students/email/{email}")
    public void delete(@PathVariable String email){ // students/email/david@gmail.com
        Optional<Student> student = studentEntityRepository.findByEmail(email);
        studentEntityRepository.delete(student.orElse( other: null));
    }
}
```

You need to know about “Mapstruct”

MapStruct is a code generator that greatly simplifies the implementation of mappings between Java bean types based on a convention over configuration approach.

The generated mapping code uses plain method invocations and thus is fast, type-safe and easy to understand.

-> Why do we need mapstruct ?

You need to know about “Mapstruct”

MapStruct is a code generator that greatly simplifies the implementation of mappings between Java bean types based on a convention over configuration approach.

The generated mapping code uses plain method invocations and thus is fast, type-safe and easy to understand.

-> Why do we need mapstruct ?

- Mapping between different object models (ex: entities <->dtos)
- Generating methods from interfaces definition

You need to know about “Mapstruct”

Adding into pom.xml

```
<dependency>
    <groupId>org.mapstruct</groupId>
    <artifactId>mapstruct</artifactId>
    <version>1.4.2.Final</version>
</dependency>
```

You need to know about “Mapstruct”

Car.java | CarDto.java

```java
public class Car {

    private String make;
    private int numberOfSeats;
    private CarType type;

    //constructor, getters, setters etc.
}
```

Car.java | CarDto.java

```java
public class CarDto {

    private String make;
    private int seatCount;
    private String type;

    //constructor, getters, setters etc.
}
```

You need to know about “Mapstruct”

CarMapper.java

```
1. @Mapper 1
2. public interface CarMapper {
3.
4.     CarMapper INSTANCE = Mappers.getMapper( CarMapper.class ); 3
5.
6.     @Mapping(source = "numberOfSeats", target = "seatCount")
7.     CarDto carToCarDto(Car car); 2
8. }
```

You need to know about “Mapstruct”

CarMapperTest.java

```java
@Test
public void shouldMapCarToDto() {
    //given
    Car car = new Car( "Morris", 5, CarType.SEDAN );

    //when
    CarDto carDto = CarMapper.INSTANCE.carToCarDto( car );

    //then
    assertThat( carDto ).isNotNull();
    assertThat( carDto.getMake() ).isEqualTo( "Morris" );
    assertThat( carDto.getSeatCount() ).isEqualTo( 5 );
    assertThat( carDto.getType() ).isEqualTo( "SEDAN" );
}
```

Homework

Read about @OneToMany, @OneToOne, @ManyToOne, @ManyToMany

Will be discussed for the next time