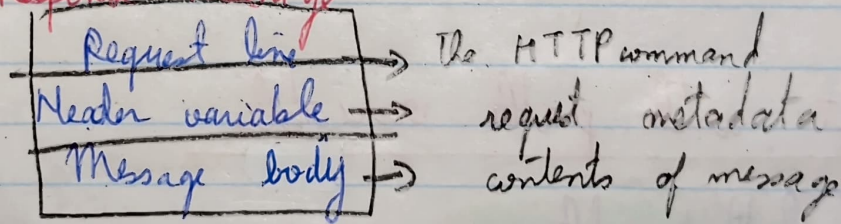


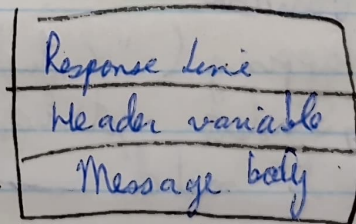
REST Gnd API.

- REST over HTTP → POST, GET, PUT, DELETE
- XML and JSON

HTTP Request Message



HTTP Response Message



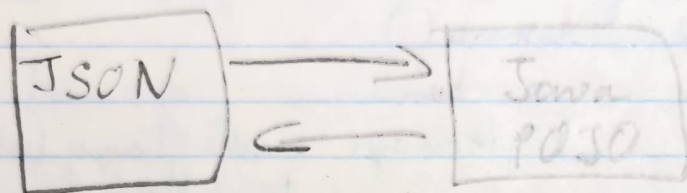
MIME Content Types

ex: - text/html

- application/json
- application/xml

Java JSON Data binding

Convert from JSON to Java POJO.



Synonyms: serializer, deserializer ☺

Jackson Project:

- support data binding
- used by Spring
- by default, call appropriate getter/setter

- JSON → POJO : use setter method
- POJO → JSON : getter method

@RestController

@RequestMapping("/api") → prefix

```
public class MyRestController { }
```

Rest with path variable

@PathVariable

@GetMapping("/students/{studentId}")

```
public Student getStudent(@PathVariable int studentId)
```

need to match

Exception Handling

- create Java POJO for error handling. Used when student not found
- @ExceptionHandler class

Global Exception Handling

@ControllerAdvice → use case quand tu veux que ce soit handle de manière global.
On crée une classe avec l'annotation @ControllerAdvice et mettre les méthodes @ExceptionHandler dedans.

API Best Practice

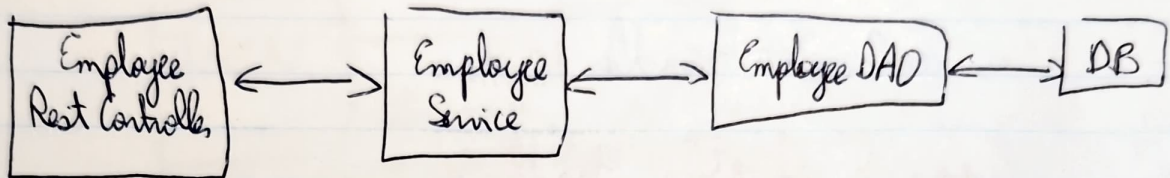
- Pluriel pour endpoints : /employees
- Ne pas mettre les verbes dans le nom des endpoints,
→ utiliser les verbes HTTP
ex: api/addEmployee X
POST api/employees ✓

crud

@Autowired
@Repository

Define Services with @Service

- Integrating info from * sources ex: employeeDAO, skillsDAO, Payroll DBA



- Spring will auto register Service implementation
- Transactional boundaries in Service
↳ @Transactional

entity Manager . merge (the Employee)

↳ if id == 0
then save/insert
else update

Spring Data JPA

Permet avoir les méthodes CRUD sans écrire classe Impl

Jpa Repository expose :

findAll()
findById()

primary key Type

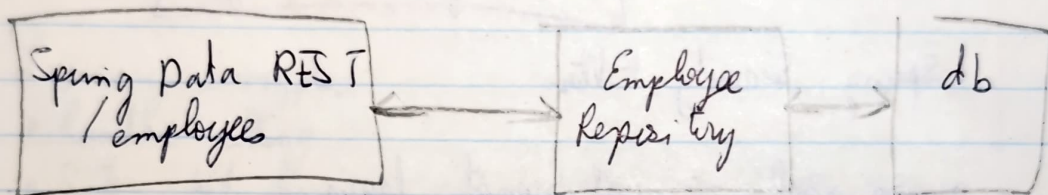
extends JpaRepository < Employee, Integer >

- Can also remove @Transactional

Spring Data REST

- Get endpoints with no coding
- Ça que tu as besoin :

1. Your entity : Employee
2. Jpa Repository : EmployeeRepository extends JpaRepository
3. Maven POM dependency : spring-boot-starter-data-rest



HATEOS → meta-data for REST data

- Le return sera la response avec d'autres choses comme page size, total elements, pages, etc..

- You can add custom queries with JPQL

Custom base url → applicati. properties
spring.data.rest.base-path = /magic-api

* PUT : met le id dans url parameter, le autres dans body

Pluralized Form :

- complex language. Default add 's'. What about Goose → Geese?
- Solution : use @RepositoryRestResource(path = "members")

Pagination :

- Return first 20 elements. Modifier dans properties
- Solution : .?page=0

o Sorting
by last Name
first name Descending

} dans url
? sort = last Name
? sort = last Name, desc

Page number are 0 sized