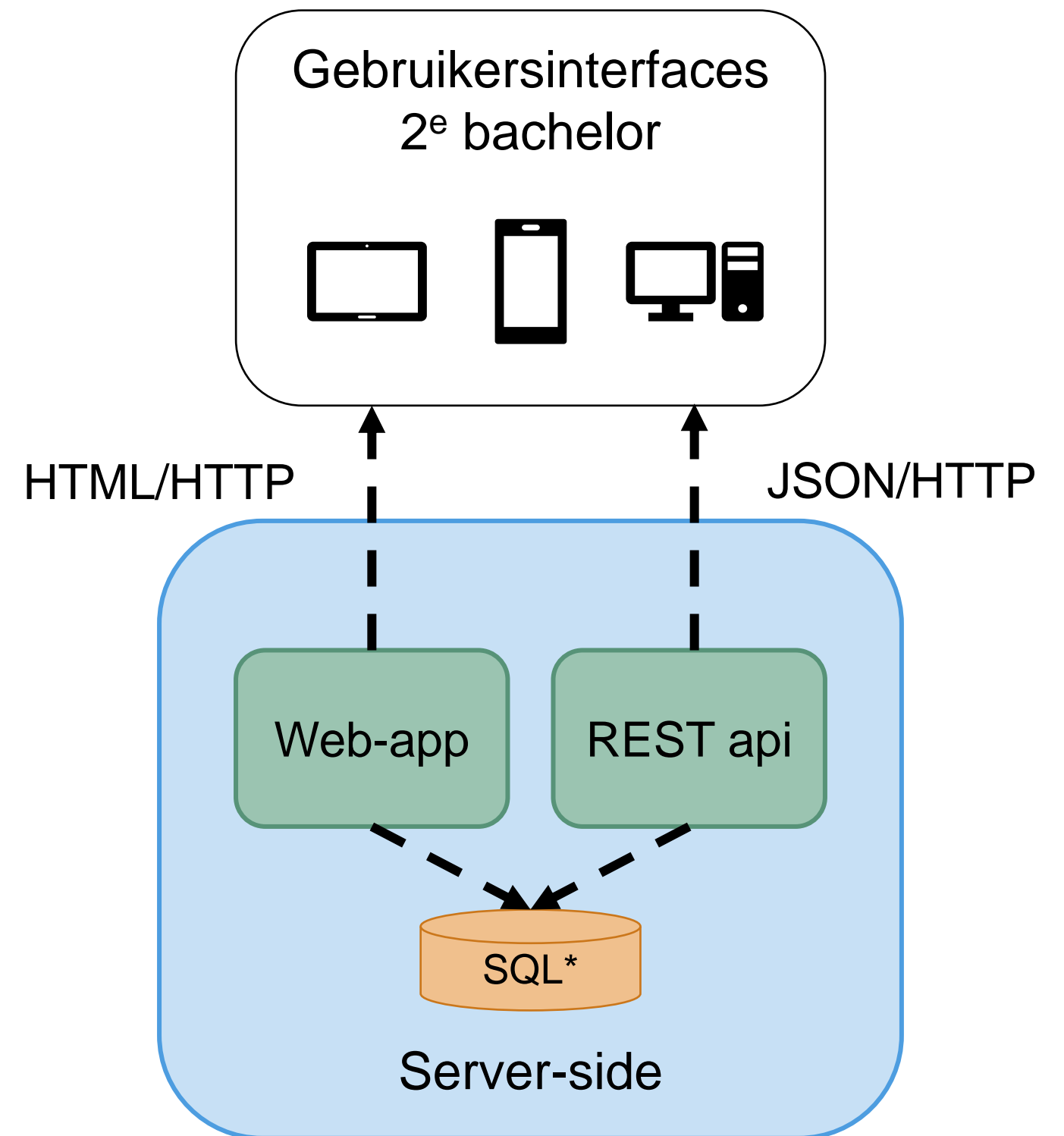


LABO 1: SPRING BOOT

Prof. Dr. Veerle Ongenae – Miel Verkerken

OVERZICHT

Datum	Labo
4 oktober	REST webservices in Java (Spring)
11 oktober	REST webservices met JPA
18 oktober	REST webservices met beveiliging
25 oktober	Groepsopdracht Node.js
1 november	GEEN LABO
8 november	Groepsopdracht Node.js
15 november	Reactive REST webservices (Spring)
22 november	Java JDBC
Week 10	TEST
29 november	ADO.NET
6 november	REST webservices met Web API
11 december	ASP.NET Core
13 december	ASP.NET Core met beveiliging



* Relationele gegevensbanken - 3^e bachelor

CREATE PROJECT (1)



Project

☐ Gradle - Groovy ☐ Gradle - Kotlin

☒ Maven

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 3.2.0 (SNAPSHOT) ☐ 3.2.0 (M3) ☐ 3.1.5 (SNAPSHOT) ☒ 3.1.4

☐ 3.0.12 (SNAPSHOT) ☐ 3.0.11 ☐ 2.7.17 (SNAPSHOT) ☐ 2.7.16

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☒ 21 ☐ 17 ☐ 11 ☐ 8

Dependencies

ADD DEPENDENCIES... CTRL + B

Spring Boot DevTools DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

GENERATE CTRL + G

EXPLORE CTRL + SPACE

SHARE...



CREATE PROJECT (2)

New Project

Empty Project

Generators

Maven Archetype

Jakarta EE

Spring Initializr

JavaFX

Quarkus

Micronaut

Ktor

Kotlin Multiplatform

Compose Multiplatform

HTML

React

Express

Angular CLI

IDE Plugin

Android

Play 2

Server URL: start.spring.io

Name: reeks1

Location: ~\Documents\onderwijs\serverframeworks\labo\2324\Reeks1

Project will be created in: ~\Documents\ond...rserverframeworks\labo\2324\Reeks1\reeks1

Create Git repository

Language: Java Kotlin Groovy

Type: Gradle - Groovy Gradle - Kotlin Maven

Group: com.example

Artifact: reeks1

Package name: com.example.reeks1

JDK: openjdk-21 Oracle OpenJDK version 21

Java: 21

Packaging: Jar War

Next Cancel

New Project

Spring Boot: 3.1.4

Download pre-built shared indexes for JDK and Maven libraries

Dependencies:

Developer Tools

GraalVM Native Support

Spring Boot DevTools

Lombok

Spring Configuration Processor

Docker Compose Support

Spring Modulith

Web

Spring Web

Spring Reactive Web

Spring for GraphQL

Rest Repositories

Spring Session

Rest Repositories HAL Explorer

Spring HATEOAS

Spring Web Services

Spring Web

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Building a RESTful Web Service

Serving Web Content with Spring MVC

Building REST services with Spring

Added dependencies:

Spring Boot DevTools

Spring Web

Previous Create Cancel

PROJECT STRUCTURE

Many Spring Boot developers like their apps to use auto-configuration, component scan and be able to define extra configuration on their "application class". A single `@SpringBootApplication` annotation can be used to enable those three features, that is:

- `@EnableAutoConfiguration`: enable Spring Boot's auto-configuration mechanism
- `@ComponentScan`: enable `@Component` scan on the package where the application is located (see [the best practices](#))
- `@Configuration`: allow to register extra beans in the context or import additional configuration classes

The `@SpringBootApplication` annotation is equivalent to using `@Configuration`, `@EnableAutoConfiguration`, and `@ComponentScan` with their default attributes, as shown in the following example:

```
package be.ugent.reeks1;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication // same as @Configuration @EnableAutoConfiguration @ComponentScan
public class Reeks1Application {

    public static void main(String[] args) { SpringApplication.run(Reeks1Application.class, args); }

}
```

POM.XML

- Project Configuration
 - modelVersion
 - groupId
 - artifactId
 - version
 - packaging
- Dependencies
 - groupId: organisation
 - artifactId: specific library
 - version
 - scope

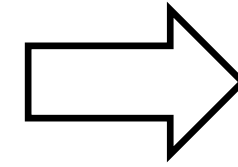
```
<modelVersion>4.0.0</modelVersion>
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>3.1.4</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>be.ugent</groupId>
<artifactId>reeks1</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>reeks1</name>
<description>Demo project for Spring Boot</description>
<properties>
  <java.version>21</java.version>
</properties>
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
```

SPRING BOOT STARTER

- Multiple dependencies
- Auto-configured

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



```
<dependencies>  
  <dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter</artifactId>  
    <version>3.1.4</version>  
    <scope>compile</scope>  
  </dependency>  
  <dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-json</artifactId>  
    <version>3.1.4</version>  
    <scope>compile</scope>  
  </dependency>  
  <dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-tomcat</artifactId>  
    <version>3.1.4</version>  
    <scope>compile</scope>  
  </dependency>  
  <dependency>  
    <groupId>org.springframework</groupId>  
    <artifactId>spring-web</artifactId>  
    <version>6.0.12</version>  
    <scope>compile</scope>  
  </dependency>  
  <dependency>  
    <groupId>org.springframework</groupId>  
    <artifactId>spring-webmvc</artifactId>  
    <version>6.0.12</version>  
    <scope>compile</scope>  
  </dependency>  
</dependencies>
```


DEPENDENCY INJECTION

```
@RestController
public class BlogController {
    private BlogPostDAOMemory postDAO;

    public BlogController(BlogPostDAOMemory postDAO) {
        this.postDAO = postDAO;
    }

    @GetMapping("/posts")
    public List<BlogPost> getPosts() {
        return postDAO.getAllPosts();
    }
}
```

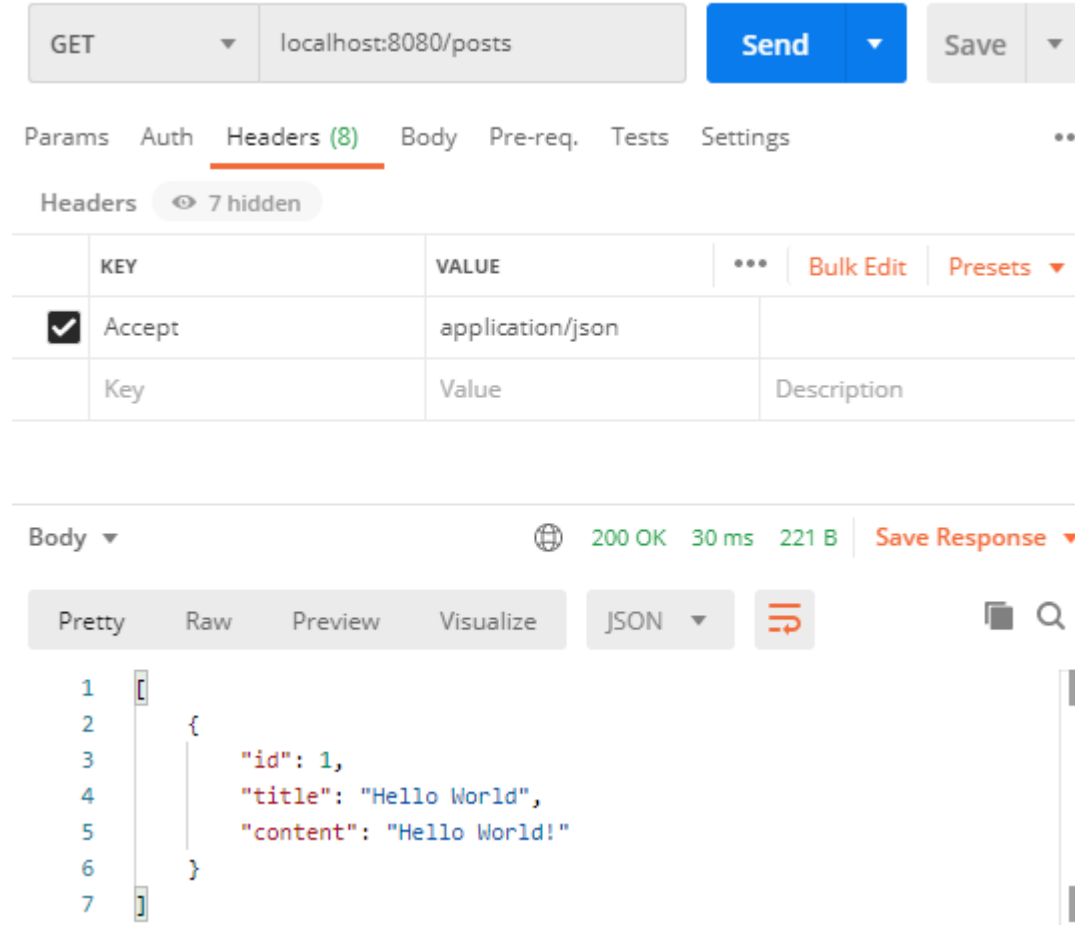
```
@RestController
public class BlogController {
    @Autowired
    private BlogPostDAOMemory postDAO;

    public BlogController() {}

    @GetMapping("/posts")
    public List<BlogPost> getPosts() {
        return postDAO.getAllPosts();
    }
}
```


JSON/XML ACCEPT HEADERS

- Json conversion default included (Jackson)
<https://codeboje.de/spring-mvc-rest-api-xml-response/>
- XML add *[jackson-dataformat-xml](#)* dependency
- Or use default JAXB but annotations are required



GET localhost:8080/posts

Send Save

Params Auth Headers (8) Body Pre-req. Tests Settings

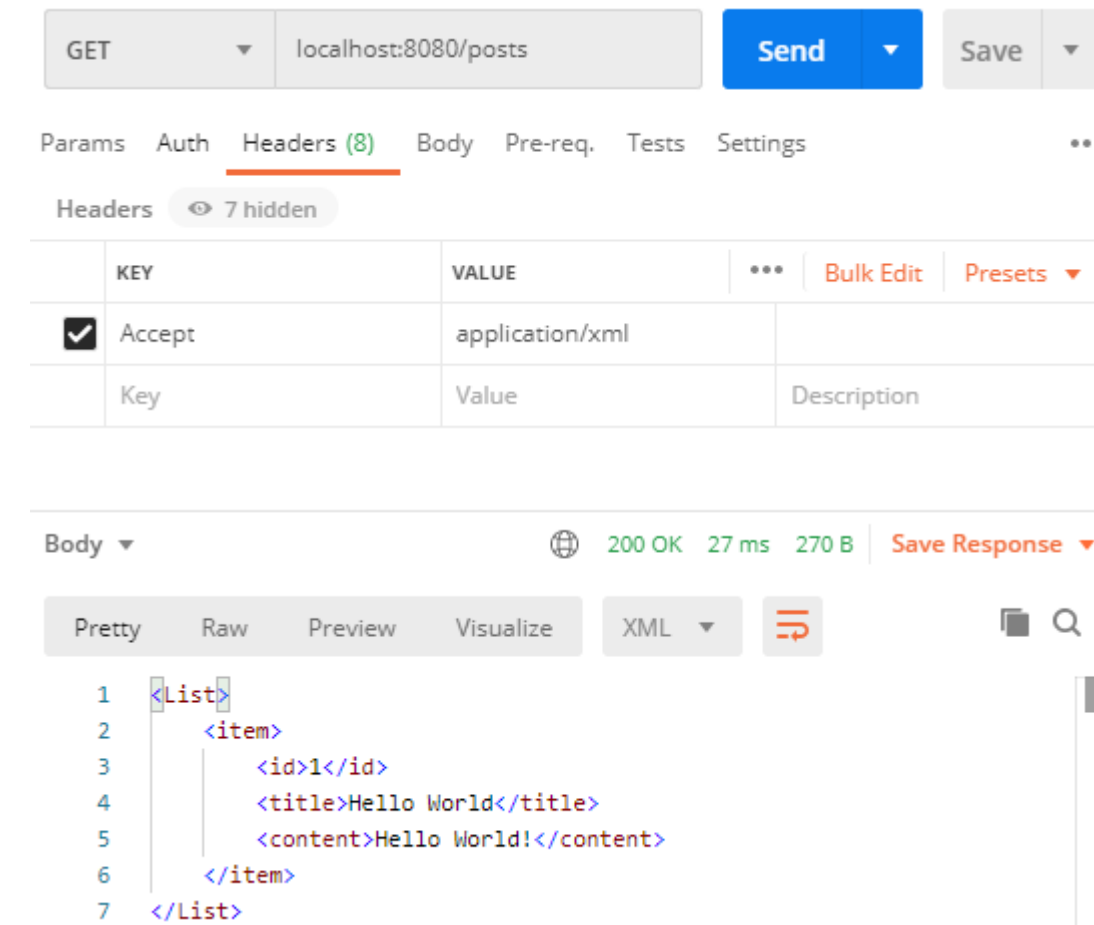
Headers 7 hidden

KEY	VALUE	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Accept	application/json			
Key	Value	Description		

Body 200 OK 30 ms 221 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "title": "Hello World",
5     "content": "Hello World!"
6   }
7 ]
```



GET localhost:8080/posts

Send Save

Params Auth Headers (8) Body Pre-req. Tests Settings

Headers 7 hidden

KEY	VALUE	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Accept	application/xml			
Key	Value	Description		

Body 200 OK 27 ms 270 B Save Response

Pretty Raw Preview Visualize XML

```
1 <List>
2   <item>
3     <id>1</id>
4     <title>Hello World</title>
5     <content>Hello World!</content>
6   </item>
7 </List>
```

CUSTOM ERROR HANDLING

```
/**
 * Provide the details of a blogPost with the given id. Throw PostNotFoundException if id doesn't exist.
 */
@GetMapping("/posts/{id}")
public BlogPost getPost(@PathVariable("id") long id) {
    return postDAO.getPost(id).orElseThrow(() -> new PostNotFoundException(id));
}

/**
 * Explicit exception handler to map PostNotFoundException to a 404 Not Found HTTP status code.
 */
@ResponseStatus(HttpStatus.NOT_FOUND)
@ExceptionHandler(PostNotFoundException.class)
public void handleNotFound(Exception ex) {
    logger.warn("Exception is: " + ex.getMessage());
    // return empty 404
}
```

Controller.class

```
public class PostNotFoundException extends RuntimeException {
    public PostNotFoundException(Long id) {
        super("Could not find post with id=" + id);
    }
}
```

Miel Verkerken

Ph.D. Student

IDLAB - IMEC

E Miel.Verkerken@ugent.be

www.ugent.be



Universiteit Gent



@ugent



@ugent



Ghent University

