# Microservices mit Spring Boot

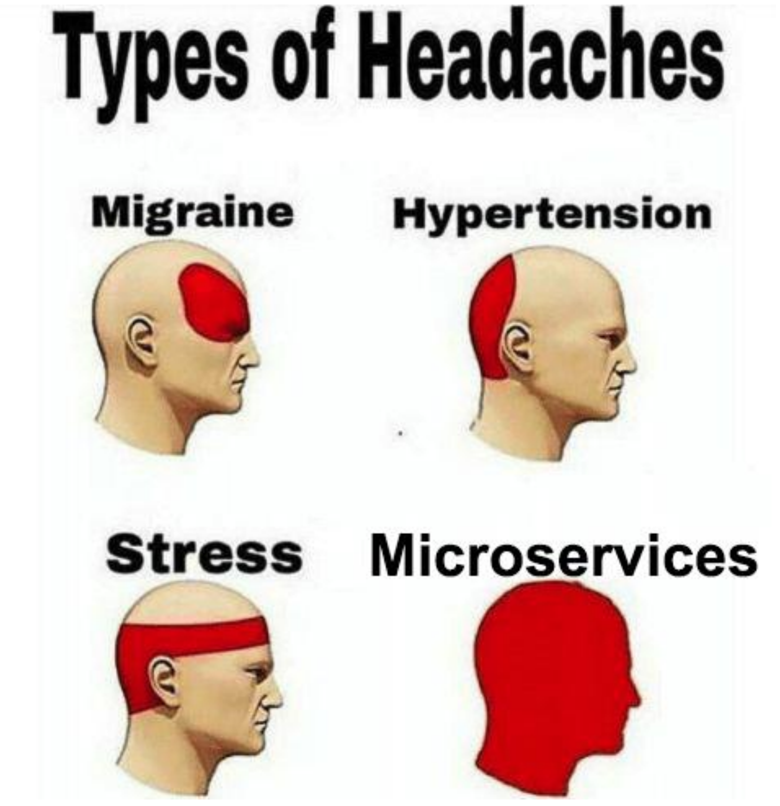
# Microservices mit Spring Boot

## Typische Technische Herausforderungen bei der Entwicklung von Microservices

- Konfigurationsmanagment
- Monitoring & Logging
- Sicherheit
- Service-Discovery
- Deployment & Testing
- Load-Balancing
- Kommunikation zwischen den Services

# Typische Technische Herausforderungen bei der Entwicklung von Microservices

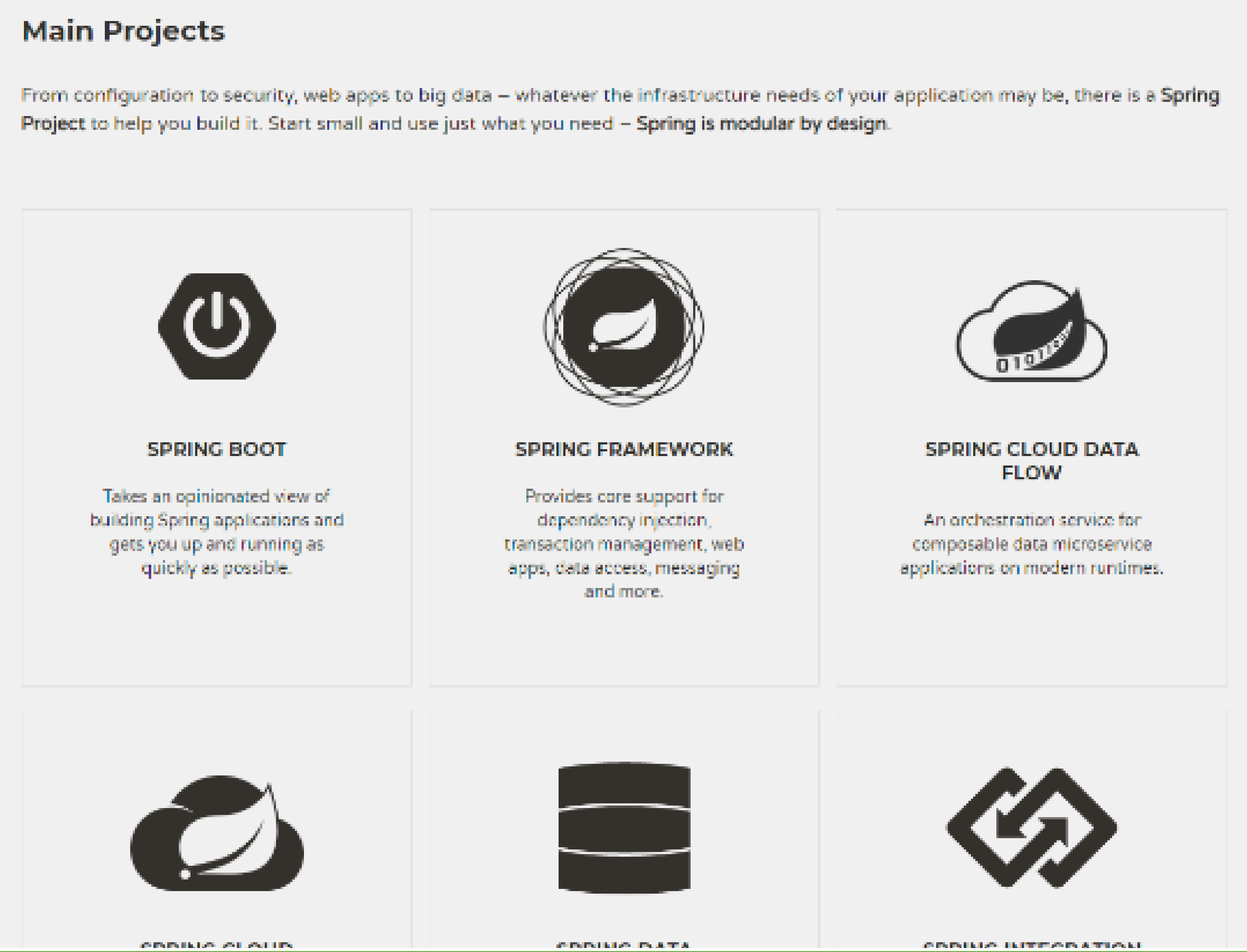
- Konfigurationsmanagment
- Monitoring & Logging
- Sicherheit
- Service-Discovery
- Deployment & Testing
- Load-Balancing
- Kommunikation zwischen den Services



# Microservices mit Spring Boot

## Was ist Spring?

- Großes open source java framework



## Main Projects

From configuration to security, web apps to big data – whatever the infrastructure needs of your application may be, there is a **Spring Project** to help you build it. Start small and use just what you need – **Spring is modular by design.**

**SPRING BOOT**

Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.

**SPRING FRAMEWORK**

Provides core support for dependency injection, transaction management, web apps, data access, messaging and more.

**SPRING CLOUD DATA FLOW**

An orchestration service for composable data microservice applications on modern runtimes.

## Was ist Spring?

- Großes open source java framework
- Ziel: die Entwicklung mit Java einfacher machen
- Basiert auf den Prinzipien Dependency Injection und AOP
- Ermöglicht ein POJO-basiertes Programmiermodell durch Annotationen
- Umfangreiche Einsatzmöglichkeiten
- Viele Konfigurationsschritte notwendig vor dem ersten Start

# Microservices mit Spring Boot

## Was verspricht Spring Boot?

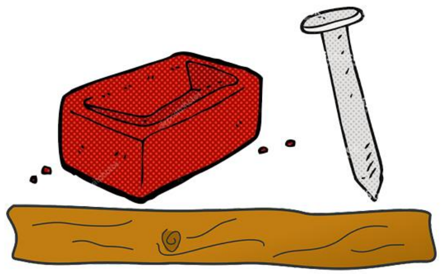
”Spring Boot makes it easy to create stand-alone, production-grade Spring-based Applications that you can run.”

## Was verspricht Spring Boot?

”Spring Boot makes it easy to create stand-alone, production-grade Spring-based Applications that you can run.”

## Was ist Spring Boot?

- Spring Projekt
- Convention-over-configuration Lösung
- Eine Zusammenstellung von Elementen der Spring Plattform und Third-Party libraries
- Voreingestellt mit den „besten“ Konfigurationen für die enthaltenen Elemente


Spring

$\longrightarrow$

Spring Boot

## Was bietet Spring Boot?

- SQL und NoSQL Unterstützung
- Embedded Server (Tomcat, Jetty, Undertow)
- Automatische Konfiguration
- Integrierte Metriken und Health Status
- Ermöglicht fat jar-Erstellung
- Devtools (auto-restart, liveReload)
- Vielzahl an Spring Boot Starter

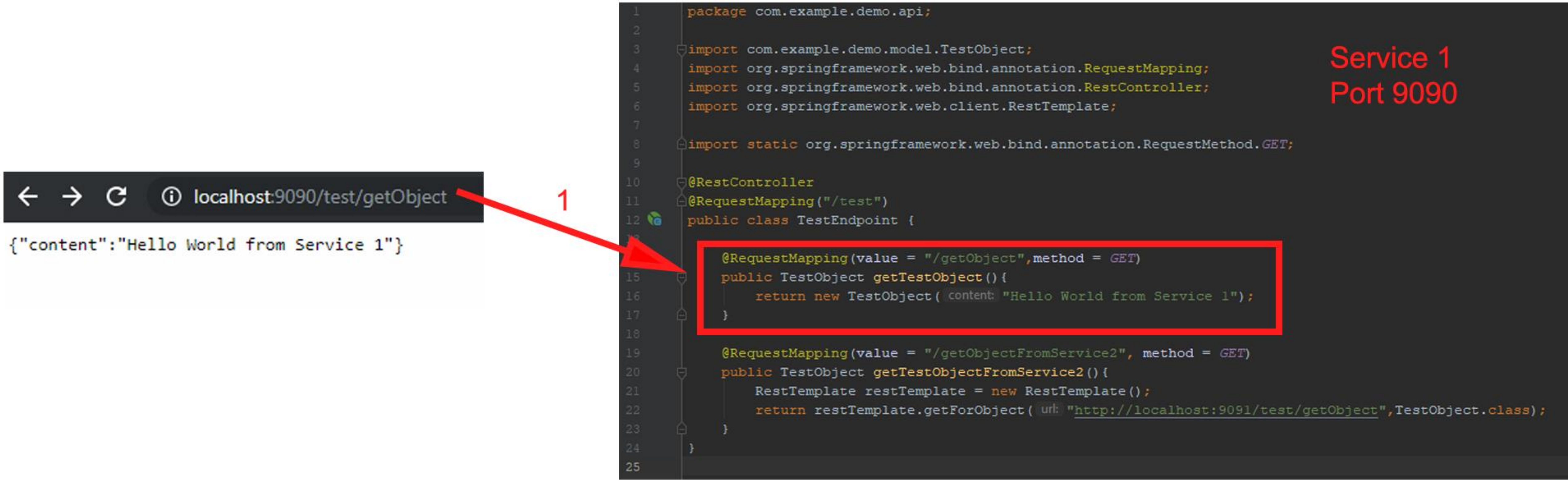
## Was ist ein Spring Boot Starter?

- Spring Boot Lösung für einen typischen Anwendungsfall
- Ad-hoc Skelettprojekt
- Beinhaltet alle notwendigen Dependencies
- Über 60 verschiedene Starter (Data,Logging,Mail,Web,...)
- Mehrere Starter können in einem Service genutzt werden
  -> viele weitere Anwendungsfälle können abgedeckt werden

## Wie erstelle ich ein Spring Boot Projekt?

- Spring Boot Projekte besitzen eine bestimmte Struktur und können von vielen IDE‘s (IntelliJ,Eclipse,…) oder durch den Spring Initializr erstellt werden
- Spring Initializer: https://start.spring.io/

# Wie kann die Kommunikation zwischen Microservices aussehen?



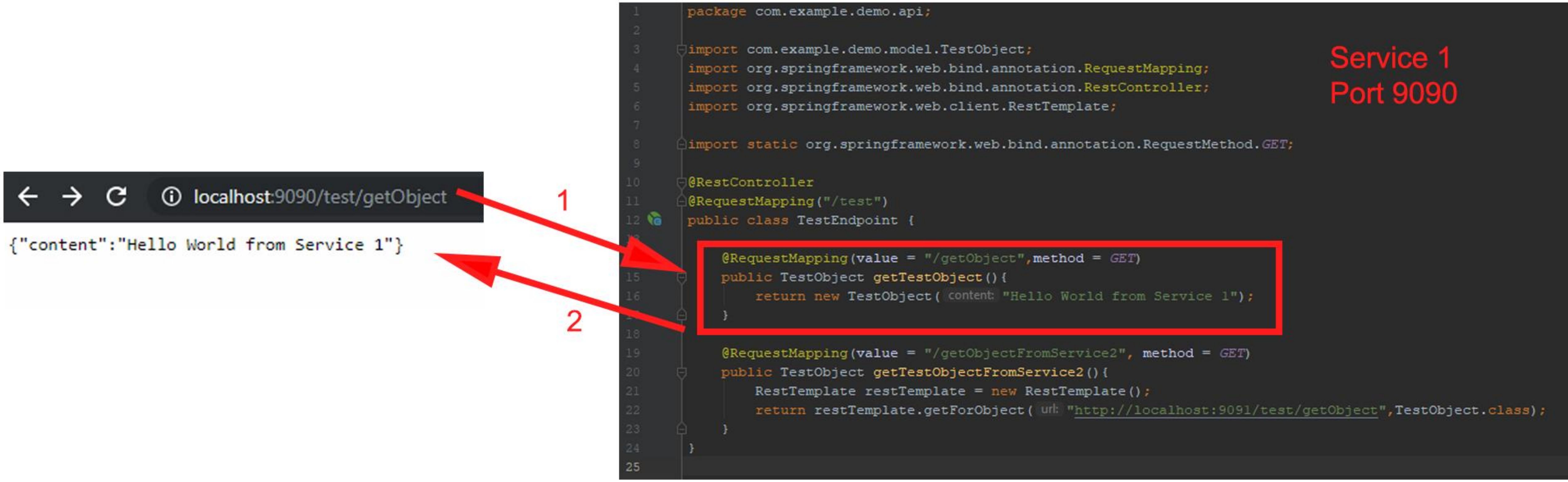
```
localhost:9090/test/getObject
```

```
{"content":"Hello World from Service 1"}
```

```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject(){
        return new TestObject( content: "Hello World from Service 1");
    }

    @RequestMapping(value = "/getObjectFromService2", method = GET)
    public TestObject getTestObjectFromService2(){
        RestTemplate restTemplate = new RestTemplate();
        return restTemplate.getForObject( url: "http://localhost:9091/test/getObject",TestObject.class);
    }
}
```

Benötigte Dependency: Spring Boot Starter Web

# Wie kann die Kommunikation zwischen Microservices aussehen?

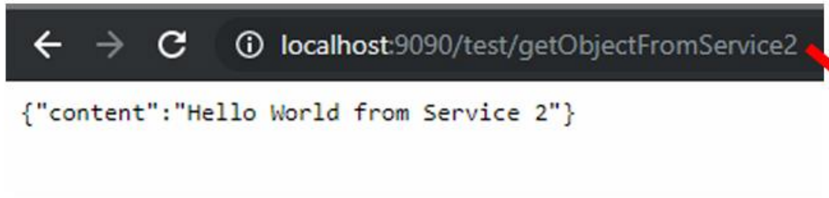


```
localhost:9090/test/getObject
```

```
{"content":"Hello World from Service 1"}
```

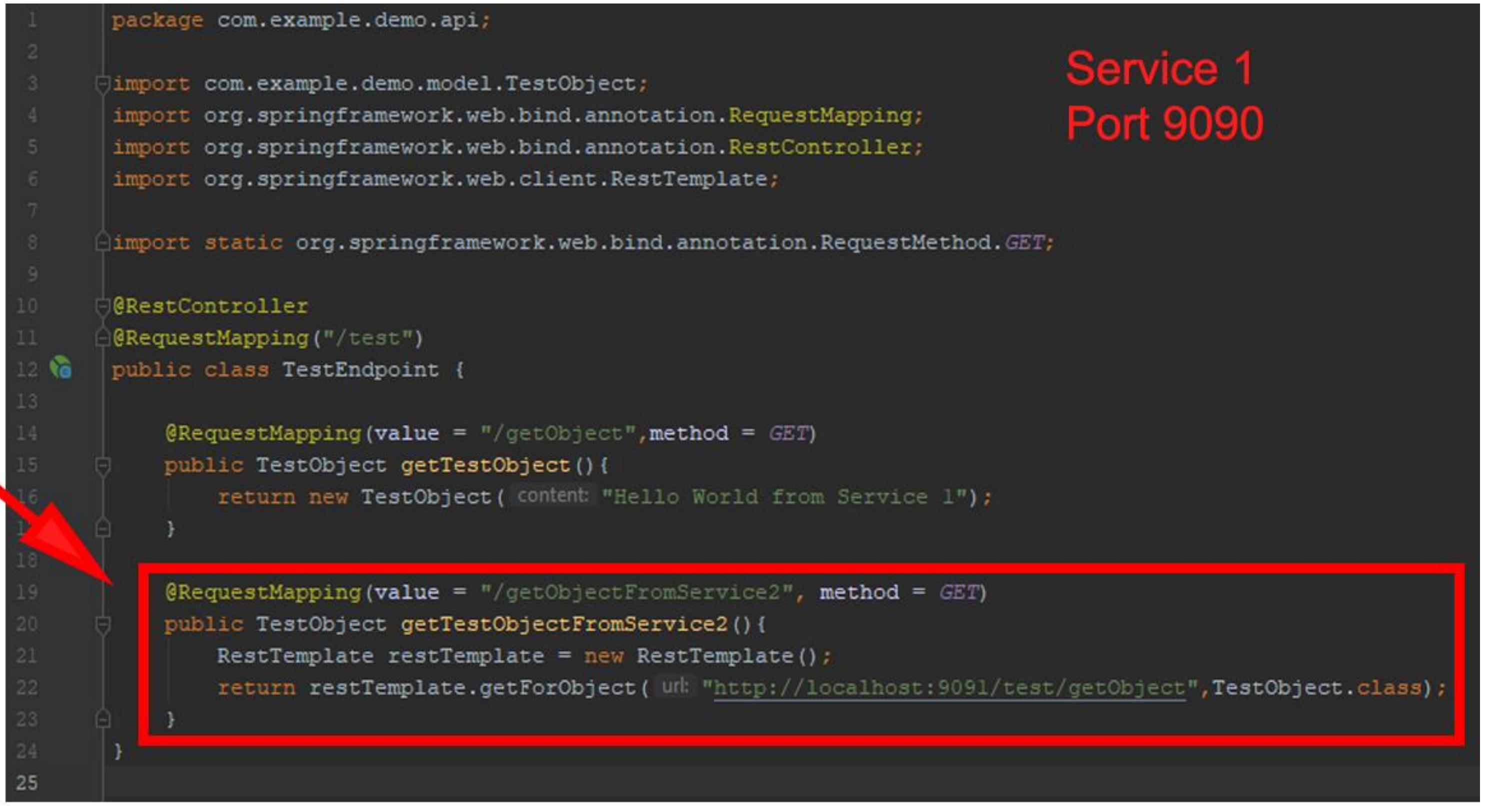
```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject(){
        return new TestObject( content: "Hello World from Service 1");
    }

    @RequestMapping(value = "/getObjectFromService2", method = GET)
    public TestObject getTestObjectFromService2(){
        RestTemplate restTemplate = new RestTemplate();
        return restTemplate.getForObject( url: "http://localhost:9091/test/getObject",TestObject.class);
    }
}
```

Benötigte Dependency: Spring Boot Starter Web

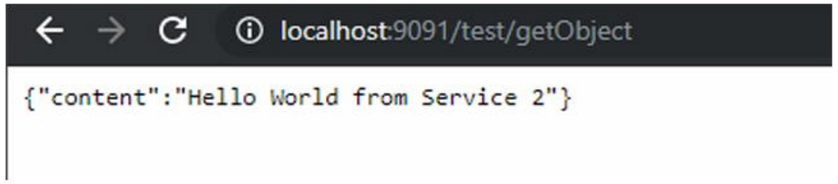


```
localhost:9090/test/getObjectFromService2
```

```
{"content":"Hello World from Service 2"}
```

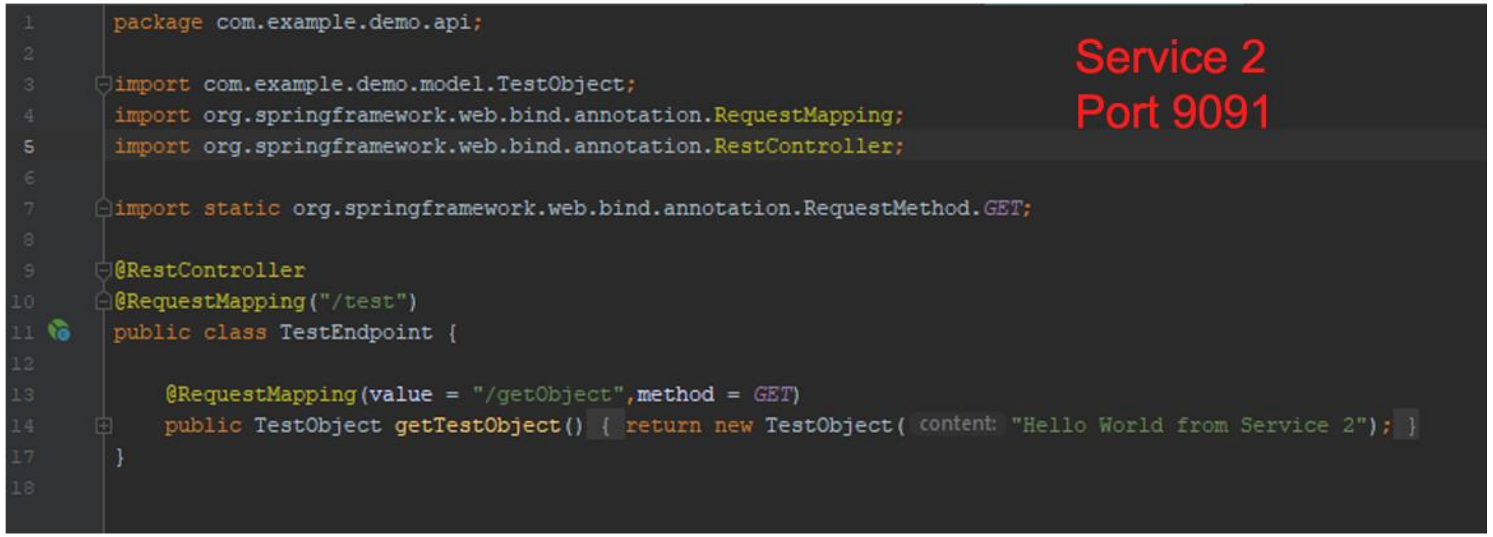


```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject(){
        return new TestObject( content: "Hello World from Service 1");
    }

    @RequestMapping(value = "/getObjectFromService2", method = GET)
    public TestObject getTestObjectFromService2(){
        RestTemplate restTemplate = new RestTemplate();
        return restTemplate.getForObject( url: "http://localhost:9091/test/getObject",TestObject.class);
    }
}
```

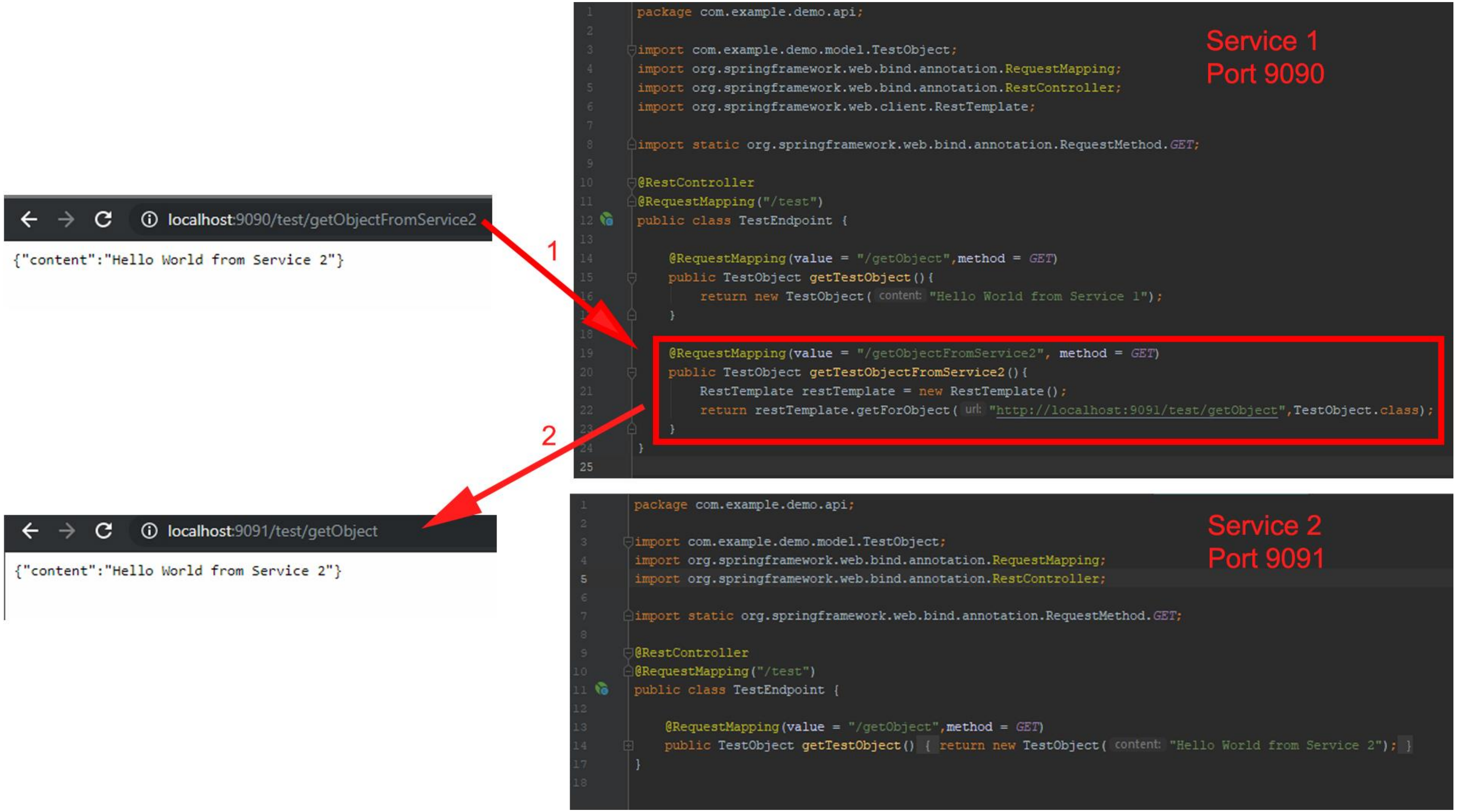


```
localhost:9091/test/getObject
```

```
{"content":"Hello World from Service 2"}
```



```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject() { return new TestObject( content: "Hello World from Service 2"); }
}
```



```
localhost:9090/test/getObjectFromService2
```

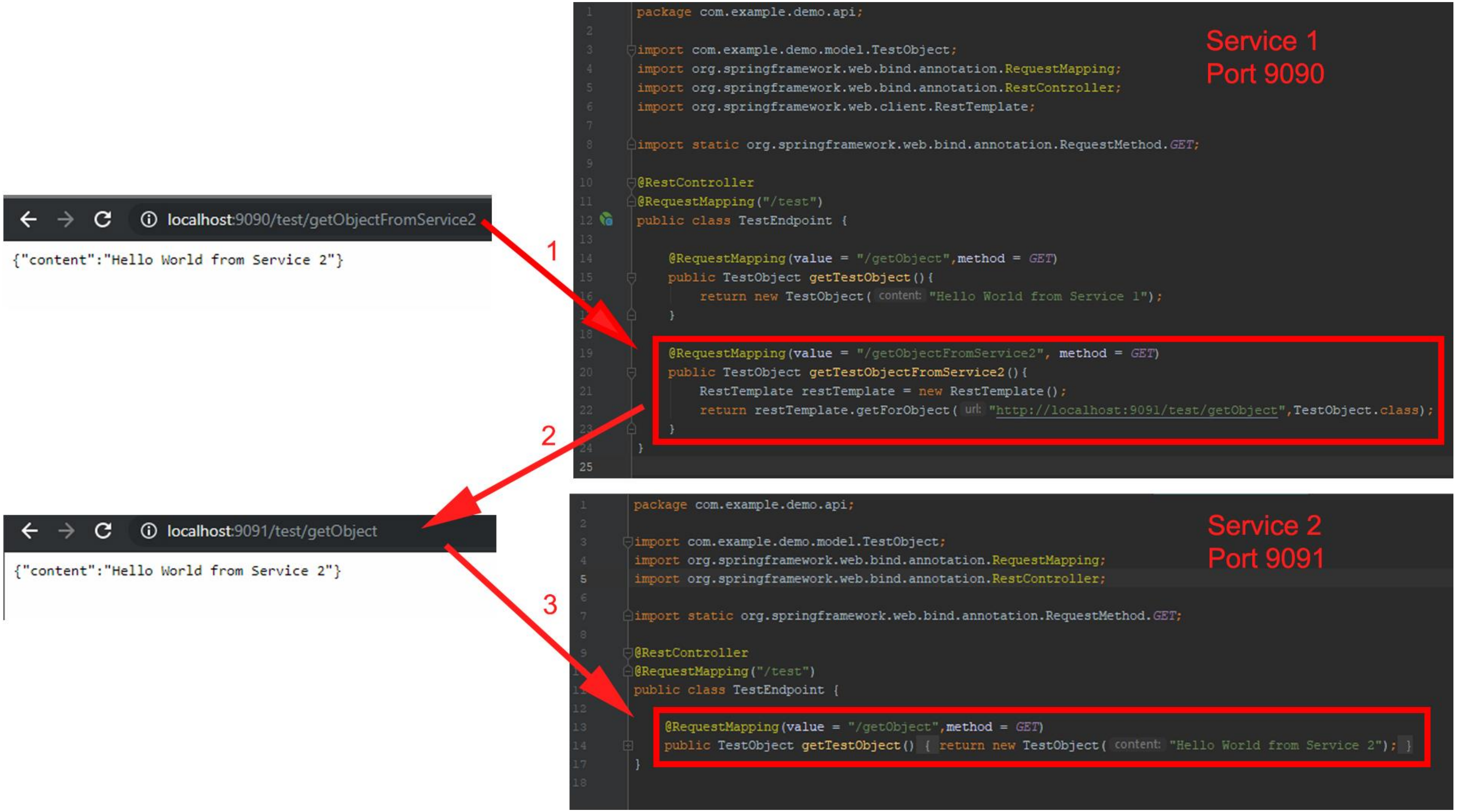
```
{"content":"Hello World from Service 2"}
```

```
localhost:9091/test/getObject
```

```
{"content":"Hello World from Service 2"}
```

```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject(){
        return new TestObject( content: "Hello World from Service 1");
    }

    @RequestMapping(value = "/getObjectFromService2", method = GET)
    public TestObject getTestObjectFromService2(){
        RestTemplate restTemplate = new RestTemplate();
        return restTemplate.getForObject( url: "http://localhost:9091/test/getObject",TestObject.class);
    }
}
```

```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject() { return new TestObject( content: "Hello World from Service 2"); }
}
```



localhost:9090/test/getObjectFromService2

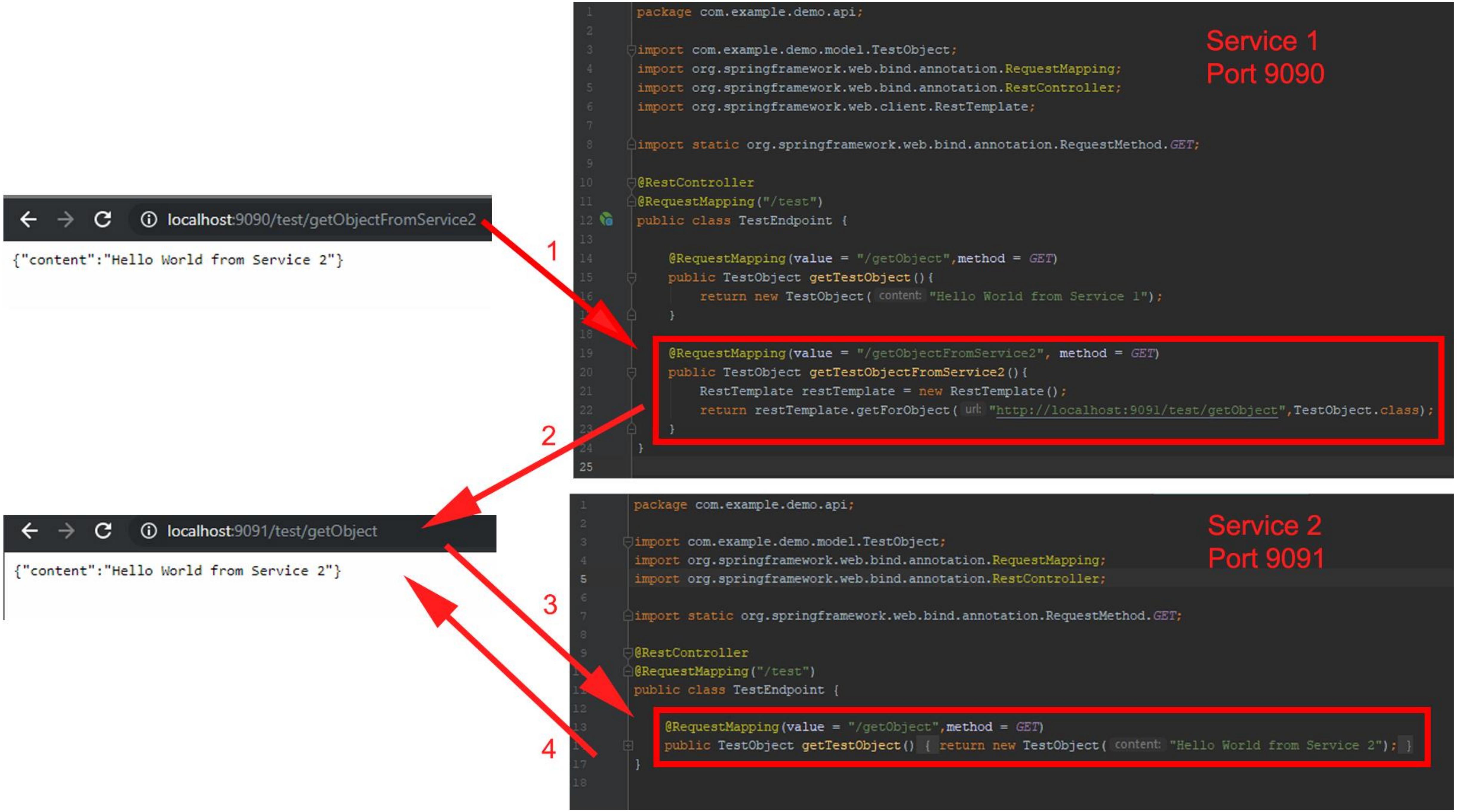
```
{"content":"Hello World from Service 2"}
```

```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject(){
        return new TestObject( content: "Hello World from Service 1");
    }

    @RequestMapping(value = "/getObjectFromService2", method = GET)
    public TestObject getTestObjectFromService2(){
        RestTemplate restTemplate = new RestTemplate();
        return restTemplate.getForObject( url: "http://localhost:9091/test/getObject",TestObject.class);
    }
}
```

localhost:9091/test/getObject

```
{"content":"Hello World from Service 2"}
```

```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject() { return new TestObject( content: "Hello World from Service 2"); }
}
```

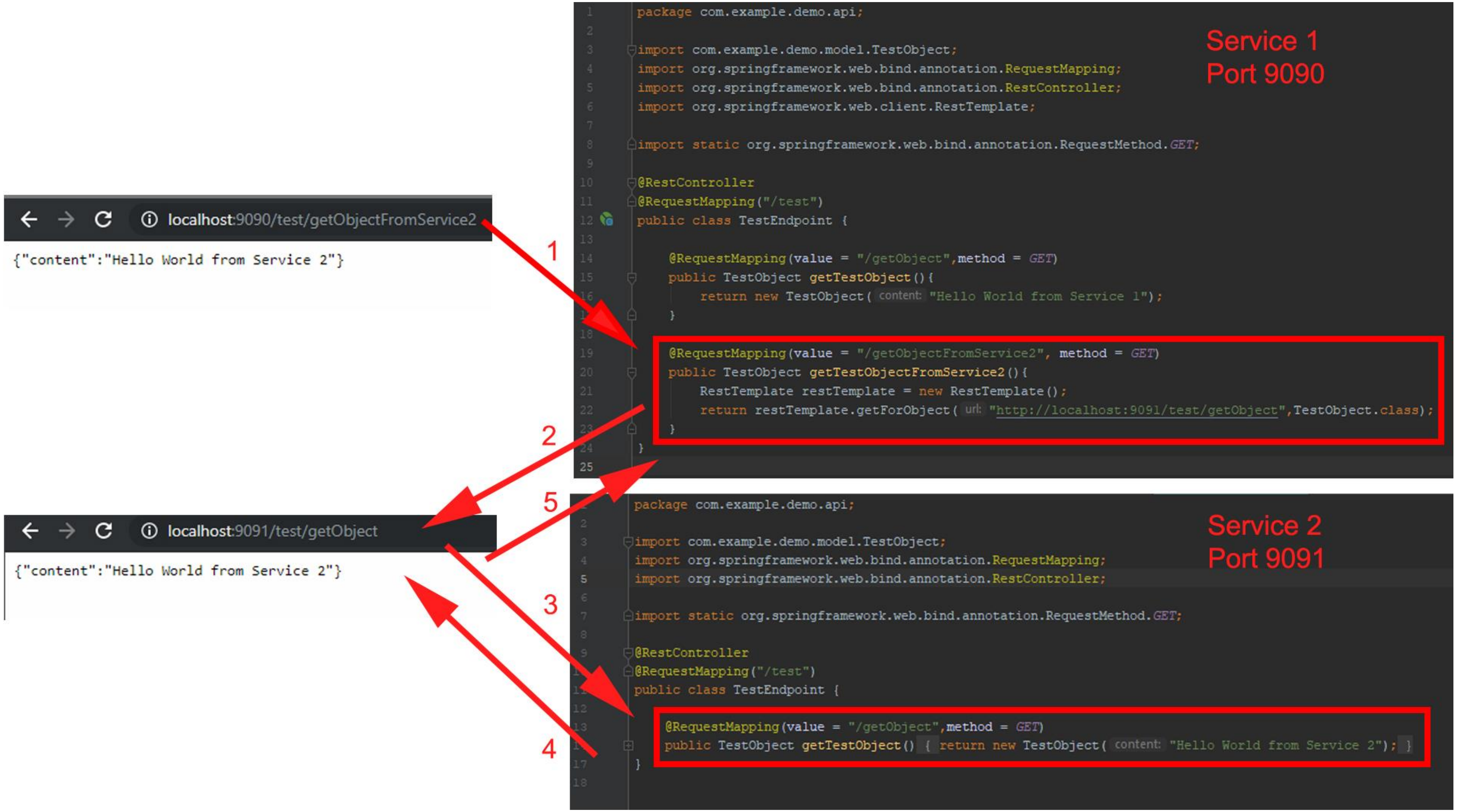


```
localhost:9090/test/getObjectFromService2
{"content":"Hello World from Service 2"}
```

```
localhost:9091/test/getObject
{"content":"Hello World from Service 2"}
```

```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject(){
        return new TestObject( content: "Hello World from Service 1");
    }

    @RequestMapping(value = "/getObjectFromService2", method = GET)
    public TestObject getTestObjectFromService2(){
        RestTemplate restTemplate = new RestTemplate();
        return restTemplate.getForObject( url: "http://localhost:9091/test/getObject",TestObject.class);
    }
}
```

```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject() { return new TestObject( content: "Hello World from Service 2"); }
}
```



localhost:9090/test/getObjectFromService2

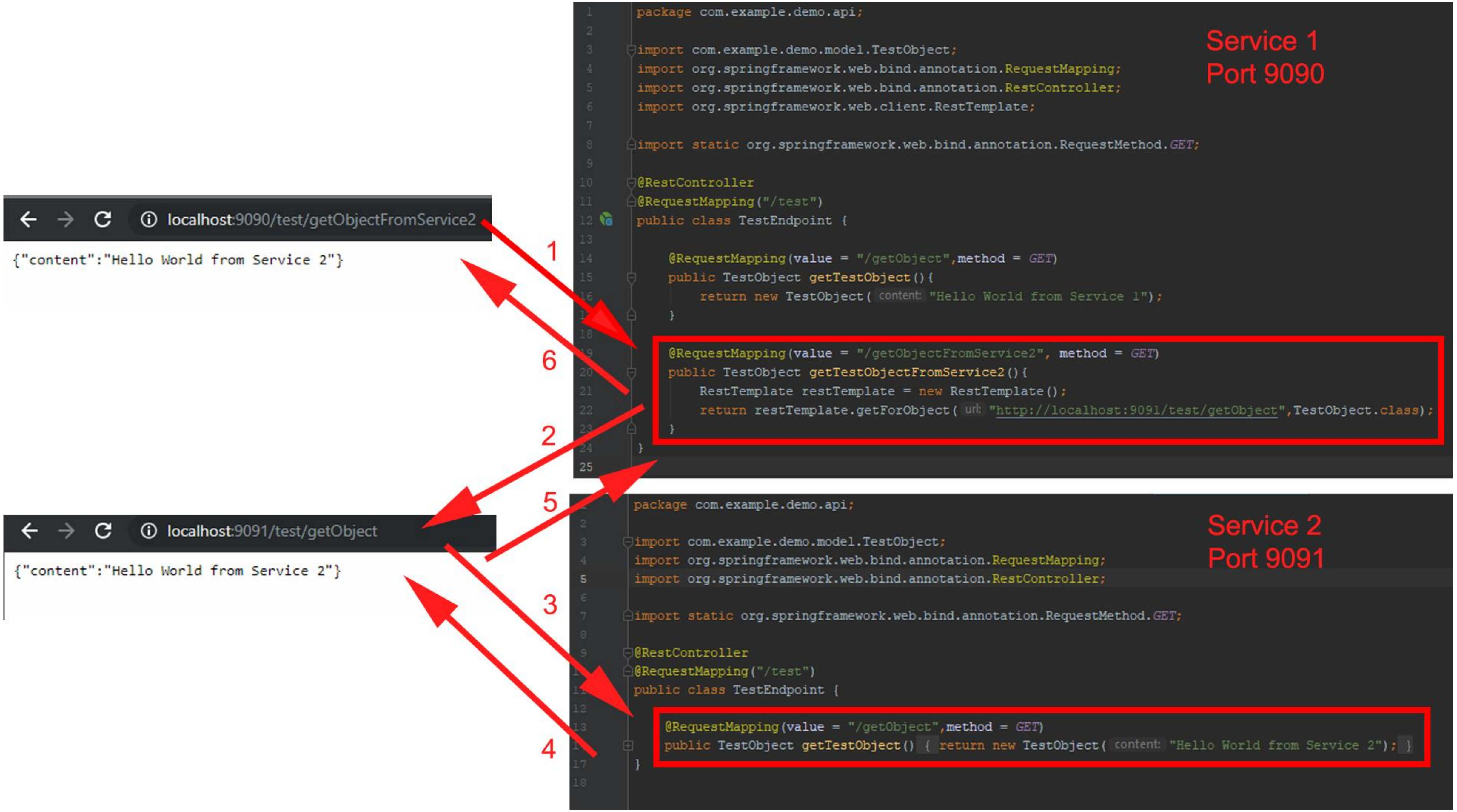
```
{"content":"Hello World from Service 2"}
```

localhost:9091/test/getObject

```
{"content":"Hello World from Service 2"}
```

```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject(){
        return new TestObject( content: "Hello World from Service 1");
    }

    @RequestMapping(value = "/getObjectFromService2", method = GET)
    public TestObject getTestObjectFromService2(){
        RestTemplate restTemplate = new RestTemplate();
        return restTemplate.getForObject( url: "http://localhost:9091/test/getObject",TestObject.class);
    }
}
```

```java
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject() { return new TestObject( content: "Hello World from Service 2"); }
}
```



localhost:9090/test/getObjectFromService2

```
{"content":"Hello World from Service 2"}
```

localhost:9091/test/getObject

```
{"content":"Hello World from Service 2"}
```

```
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject(){
        return new TestObject( content: "Hello World from Service 1");
    }

    @RequestMapping(value = "/getObjectFromService2", method = GET)
    public TestObject getTestObjectFromService2(){
        RestTemplate restTemplate = new RestTemplate();
        return restTemplate.getForObject( url: "http://localhost:9091/test/getObject",TestObject.class);
    }
}
```

```
package com.example.demo.api;

import com.example.demo.model.TestObject;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import static org.springframework.web.bind.annotation.RequestMethod.GET;

@RestController
@RequestMapping("/test")
public class TestEndpoint {

    @RequestMapping(value = "/getObject",method = GET)
    public TestObject getTestObject() { return new TestObject( content: "Hello World from Service 2"); }
}
```

## Typische Technische Herausforderungen bei der Entwicklung von Microservices

- Konfigurationsmanagment ✓ erweiterbar durch:  Spring Cloud Config
- Monitoring & Logging ✓ - Spring Boot Actuator & Spring Boot Starter Logging
- Sicherheit ✓ - Spring Boot Starter Security
- Service-Discovery ✓ - Spring Cloud Starter Eureka Client
- Deployment & Testing ✓ - fat jar Erstellung & Spring Boot Starter Test
- Load-Balancing ✓ - Spring Cloud Starter Netflix Ribbon
- Kommunikation zwischen den Services ✓ - Spring Boot Starter Web
  erweiterbar durch: Feign, Hystrix

Fragen oder Anmerkungen?

Präsentation und Code: https://bit.ly/2WsoaIA

## Quellen:

- https://www.innoq.com/de/articles/2016/10/microservices-eine-bestandsaufnahme/
- https://de.wikipedia.org/wiki/Spring_(Framework)
- https://docs.spring.io/spring/docs/5.1.6.RELEASE/spring-framework-reference/overview.html#overview
- https://start.spring.io/
- https://docs.spring.io/spring-boot/docs/2.1.4.RELEASE/reference/pdf/spring-boot-reference.pdf
- https://howtodoinjava.com/spring-boot-tutorials/
- https://jaxenter.de/eine-fruehlingshafte-loesung-fuer-microservices-spring-boot-42028
- https://jaxenter.de/spring-boot-2279
- https://www.javaguides.net/2019/01/standard-project-structure-for-spring-boot-projects.html
- https://docs.spring.io/spring-boot/docs/current/gradle-plugin/reference/html/
- https://spring.io/guides/gs/serving-web-content/
- https://docs.spring.io/spring/docs/5.1.6.RELEASE/spring-framework-reference/overview.html#overview
- https://www.youtube.com/watch?v=WPKv8NA-ZhE