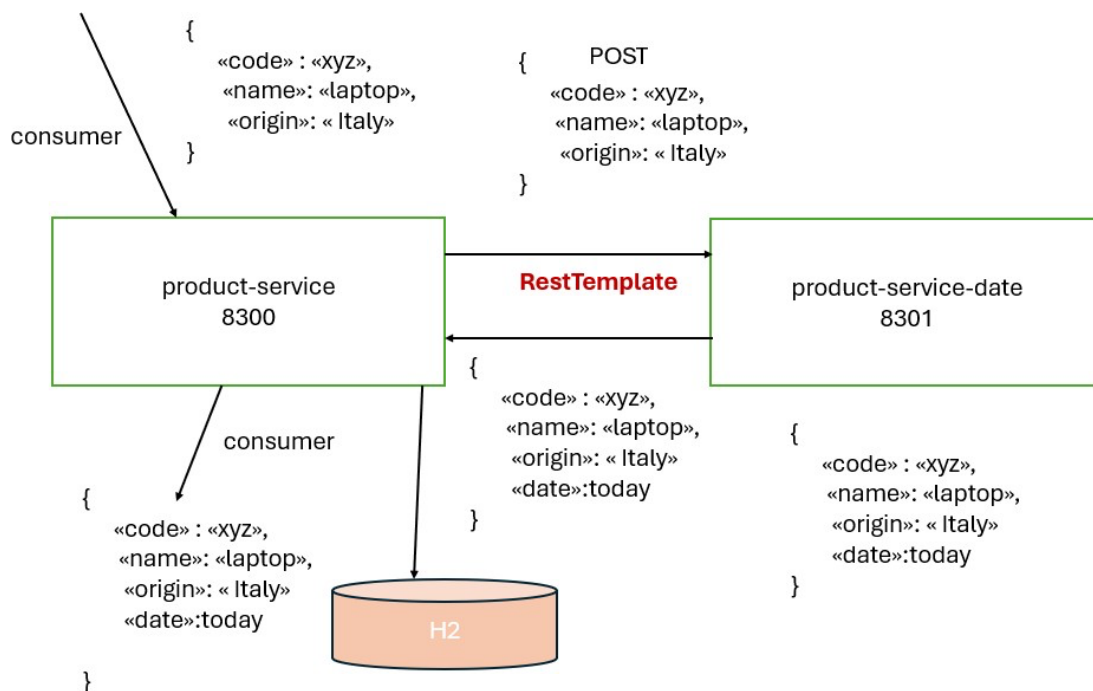


## COMUNICAZIONE TRAMITE MICROSERVIZI VIA RestTemplate

### ARCHITETTURA REALIZZATA



### REALIZZAZIONE PRIMO MICROSERVIZIO (product-service) - MAVEN COME package manager

1. Starters: web, spring data, h2
2. configurazione application.properties

```
spring.application.name=product-service
server.port=8300
spring.h2.console.enabled=true
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
```

### 3. Implementazione Entity Product

```
@Entity
public class Product implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    private String code;

    private String name;

    private String origin;

    public String getCode() {
```

```

        return code;
    }

    public void setCode(String code) {
        this.code = code;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getOrigin() {
        return origin;
    }

    public void setOrigin(String origin) {
        this.origin = origin;
    }

    protected Product() {
    }

    public Product(String code, String name, String origin) {

        this.code = code;
        this.name = name;
        this.origin = origin;
    }
}

```

#### 4. Implementazione Interfaccia Repository Spring Data JPA

```

public interface ProductRepository extends JpaRepository<Product, String> {}

```

#### 5. Implementazione Service Layer

```

package com.sistemi.informativi.service;

import com.sistemi.informativi.entity.Product;

public interface ProductService {

    public Product saveProduct(Product product);
}

```

```

package com.sistemi.informativi.service;

import org.springframework.stereotype.Service;

import com.sistemi.informativi.entity.Product;
import com.sistemi.informativi.repository.ProductRepository;

@Service
public class ProductServiceImpl implements ProductService {

    private ProductRepository productRepository;

    public ProductServiceImpl(ProductRepository productRepository) {

        this.productRepository = productRepository;
    }

    // BUSINESS LOGIC TO DO

    @Override
    public Product saveProduct(Product product) {

        return productRepository.save(product);
    }
}

```

## 6. Implementazione RestController

```

package com.sistemi.informativi.controller;

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

import com.sistemi.informativi.entity.Product;
import com.sistemi.informativi.service.ProductService;

@RestController
@RequestMapping("/rest/api/products")
public class ProductController {

    /*
     * Una volta chiesto a Spring Boot di istanziare il RestTemplate possiamo
     * chiedere la DI di RestTemplate
     */

    private ProductService productService;

    private RestTemplate restTemplate;

```

```

public ProductController(ProductService productService, RestTemplate restTemplate) {

this.productService = productService;
this.restTemplate = restTemplate;
}

/*
 * Questo Microservizio deve ricevere un JSON che rappresenta un prodotto
 * ESEMPIO: { "code":"xyz", "name:"laptop", "origin":"Italy" }
 *
 * Dopo aver ricevuto il JSON fa in modo che il suo contenuto venga salvato sul
 * database H2 Successivamente invia il JSON del prodotto ad un secondo
 * microservizio che chiameremo product-service-date e che starà in ascolto
 * sulla porta 3301 con url rest/api/products/date
 *
 *
 */

@PostMapping
public Object saveAndSendProduct(@RequestBody Product product) {

// URL DEL MICROSERVIZIO CON CUI COMUNICHIAMO AL QUALE INVIAMO
// IL JSON DEL
// PRODOTTO
String apiURL = "http://localhost:8301/rest/api/products/date";

// IL PRODOTTO VIENE INSERITO SU H2
productService.saveProduct(product);

/*
 * Tramite il metodo postForObject consumiamo una operazione esposta sotto forma
 * di post dal secondo microservizio
 */
return restTemplate.postForObject(apiURL, product, Object.class);

/*
 * VIENE RESTITUITO AL CONSUMER UN JSON ARRICCHITO CON DATA
 * ODIERNA ESEMPIO {
 * "code":"xyz", "name:"laptop", "origin":"Italy", "date" :today }
 */

}

}

```

## REALIZZAZIONE SECONDO MICROSERVIZIO (product-service-date) - MAVEN COME package manager

1. Starters:web
2. configurazione application.properties  
spring.application.name=product-service-date  
server.port=8301

### 3. Implementazione JavaBean Product

```
public class Product {  
  
    private String code;  
  
    private String name;  
  
    private String origin;  
  
    private Date date;  
  
    public String getCode() {  
        return code;  
    }  
  
    public void setCode(String code) {  
        this.code = code;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getOrigin() {  
        return origin;  
    }  
  
    public void setOrigin(String origin) {  
        this.origin = origin;  
    }  
  
    public Date getDate() {  
        return date;  
    }  
  
    public void setDate(Date date) {  
        this.date = date;  
    }  
  
    protected Product() {  
  
    }  
}
```

```

    public Product(String code, String name, String origin, Date date) {

        this.code=code;
        this.name = name;
        this.origin = origin;
        this.date = date;
    }

}

```

#### 4. Implementazione @RestController

```

@RestController
@RequestMapping("/rest/api/products/dates")
public class ProductDateController {

    @PostMapping
    public @ResponseBody Object addDateToProduct(@RequestBody Product product){

        Product receivedProduct = product;
        receivedProduct.setDate(new Date(System.currentTimeMillis()));

        return receivedProduct;
    }

}

```

#### TEST CON POSTMAN

**POST http://localhost/8300/rest/api /products**

**con JSON**

```

{
  "code": "xyz",
  "name": "laptop",
  "origin": "Italy"
}

```