Microservicios – Arquitectura y Desarrollo

LAB Balanceo de Carga con Ribbon (Visualizando el Puerto)

Objetivos

- Mostrar al participante el procedimiento para el balanceo de carga de Microservicios usando Netflix Ribbon
- Visualizar que instancia del servicio está procesando la petición (Request)

Procedimiento

1. En la entidad Producto agrega un campo llamado port de tipo transient en el proyecto **servicio-productos**

```
package com.cognos.app.productos.model.entity;

import java.io.Serializable;
import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.Transient;

@Entity
@Table(name="productos")
public class Producto implements Serializable{

	private static final long serialVersionUID = 3526278020978785736L;

	@Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
	private Long id;
	private String nombre;
	private Double precio;
	@Column(name = "create_at")
	@Temporal(TemporalType.DATE)
	private Date createAt;
	@Transient
```

```
	private Integer port;

	public Integer getPort() {
		return port;
	}
	public void setPort(Integer port) {
		this.port = port;
	}
	public Long getId() {
		return id;
	}
	public void setId(Long id) {
		this.id = id;
	}
	public String getNombre() {
		return nombre;
	}
	public void setNombre(String nombre) {
		this.nombre = nombre;
	}
	public Double getPrecio() {
		return precio;
	}
	public void setPrecio(Double precio) {
		this.precio = precio;
	}
	public Date getCreateAt() {
		return createAt;
	}
	public void setCreateAt(Date createAt) {
		this.createAt = createAt;
	}


}
```

2. En el proyecto **servicio-items** agrega el campo port a la clase

```
package com.cognos.app.items.model;

import java.util.Date;

public class Producto {
   private Long id;
   private String nombre;
   private Double precio;
   private Date createAt;

   private Integer port;

public Integer getPort() {
	return port;
```

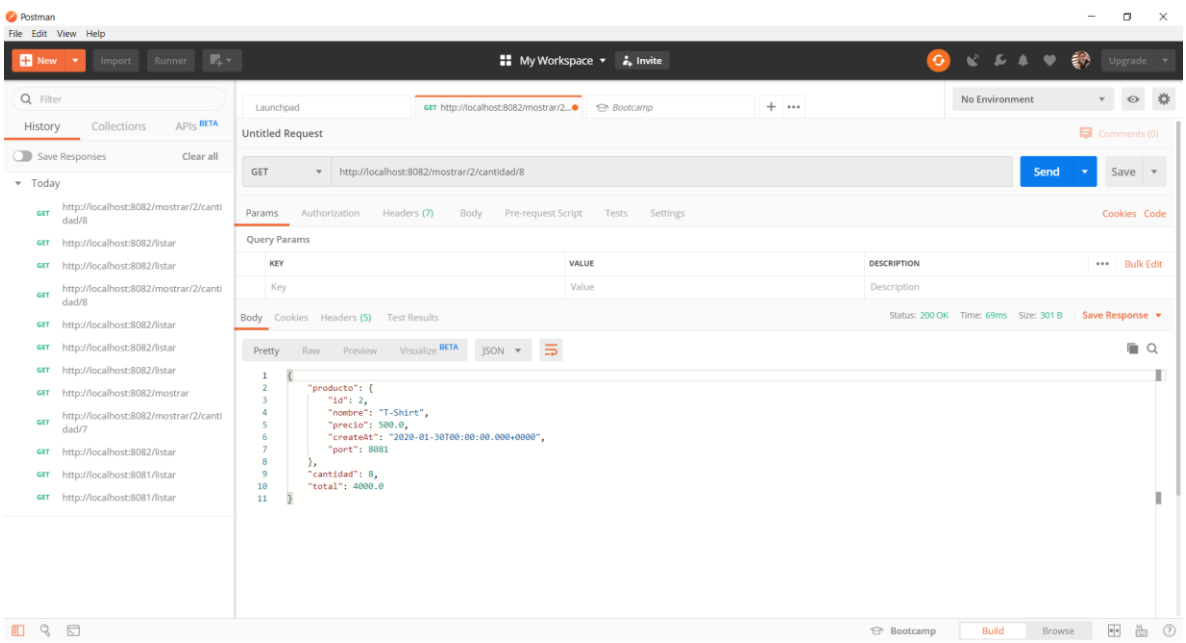
```
}
public void setPort(Integer port) {
	this.port = port;
}
public Long getId() {
	return id;
}
public void setId(Long id) {
	this.id = id;
}
public String getNombre() {
	return nombre;
}
public void setNombre(String nombre) {
	this.nombre = nombre;
}
public Double getPrecio() {
	return precio;
}
public void setPrecio(Double precio) {
	this.precio = precio;
}
public Date getCreateAt() {
	return createAt;
}
public void setCreateAt(Date createAt) {
	this.createAt = createAt;
}


}
```

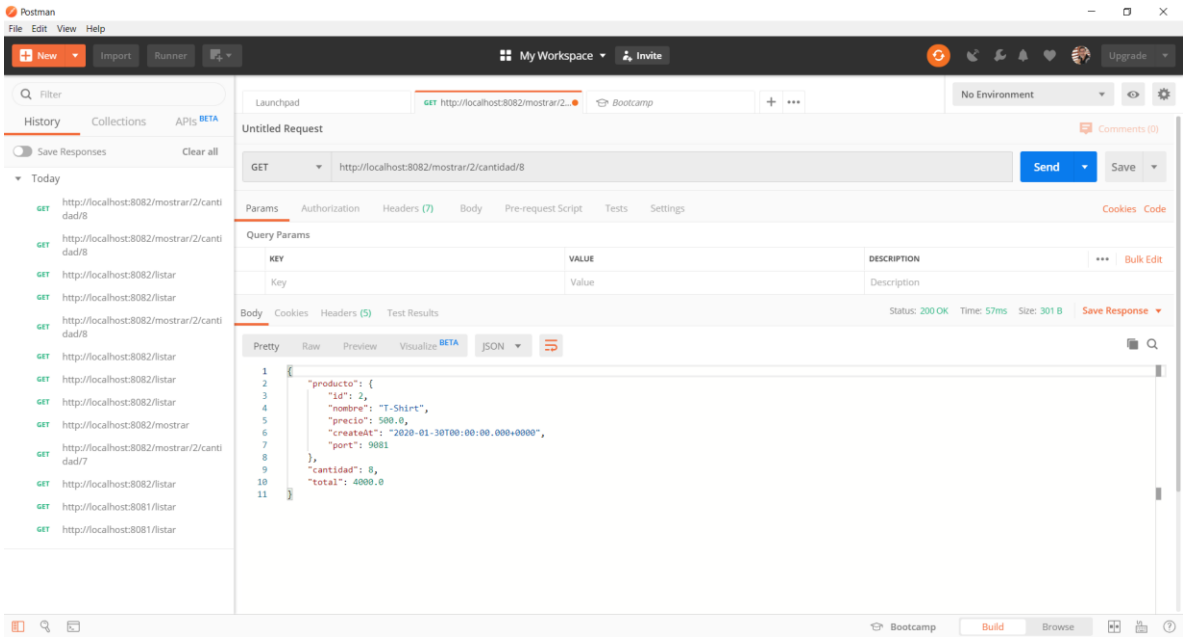
3. Modifica la clase **ProductoController** en el proyecto **servicio-productos**

```
package com.cognos.app.productos.controllers;

import java.util.List;
import java.util.stream.Collectors;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.env.Environment;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RestController;

import com.cognos.app.productos.model.entity.Producto;
import com.cognos.app.productos.model.service.ProductoService;

@RestController
public class ProductoController {

	@Autowired
```

```
        private Environment env;


        @Autowired
        private ProductoService productoService;

        @GetMapping("/listar")
        public List<Producto> listar(){
                return productoService.findAll().stream().map(p -> {

        p.setPort(Integer.parseInt(env.getProperty("local.server.port")));
                        return p;
                }).collect(Collectors.toList());
        }

        @GetMapping("/mostrar/{id}")
        public Producto detalle(@PathVariable Long id) {
                Producto producto = productoService.findById(id);

        producto.setPort(Integer.parseInt(env.getProperty("local.server.port")));
                return producto;
        }
}
```

4. Reinicia las instancias del servicio servicio-productos, luego inicia el servicio **servicio-items** y prueba desde postman





Nota: Realiza sucesivas peticiones y visualiza el cambio del puerto en cada petición, esto es porque el algoritmo por defecto de Ribbon es “round robin”

(Opcional) ***Usando Balanceo de Carga con RestTemplate***

5. Edita la clase **ProductoController**

```
package com.cognos.app.productos.controllers;

import java.util.List;
import java.util.stream.Collectors;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.core.env.Environment;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RestController;

import com.cognos.app.productos.model.entity.Producto;
import com.cognos.app.productos.model.service.ProductoService;

@RestController
public class ProductoController {

	@Autowired
	private Environment env;
```

```
	@Value("${server.port}")
	private Integer port;

	@Autowired
	private ProductoService productoService;

	@GetMapping("/listar")
	public List<Producto> listar(){
		return productoService.findAll().stream().map(p -> {

	//p.setPort(Integer.parseInt(env.getProperty("local.server.port")));
			p.setPort(port);
			return p;
		}).collect(Collectors.toList());
	}

	@GetMapping("/mostrar/{id}")
	public Producto detalle(@PathVariable Long id) {
		Producto producto = productoService.findById(id);

	//producto.setPort(Integer.parseInt(env.getProperty("local.server.port"))
);
		producto.setPort(port);
		return producto;
	}
}
```

6. Modifica la clase **AppConfig** en el proyecto **servicio-items**

```
package com.cognos.app.items;

import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.client.RestTemplate;

@Configuration
public class AppConfig {
	@Bean("clienteRest")
	@LoadBalanced
	public RestTemplate registrarRestTemplate() {
		return new RestTemplate();
	}
}
```

7. Modifica la clase **ItemServiceImpl** en el proyecto de **servicio-items**

```
package com.cognos.app.items.model.service;

import java.util.Arrays;
import java.util.HashMap;
```

```java
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

import com.cognos.app.items.model.Item;
import com.cognos.app.items.model.Producto;

@Service
public class ItemServiceImpl implements ItemService {

	@Autowired
	private RestTemplate clienteRest;
	@Override
	public List<Item> findAll() {
		String url="http://servicio-productos/listar";
		List<Producto> productos =
Arrays.asList(clienteRest.getForObject(url, Producto[].class));
		List<Item> items =  productos.stream().map(p -> new
Item(p,1)).collect(Collectors.toList());
		return items;
	}

	@Override
	public Item findbyId(Long id, Integer cantidad) {
		Map<String, String> pathVariables = new HashMap<String,String>();
		pathVariables.put("id",id.toString());
		String url = "http://servicio-productos/mostrar/{id}";
		Producto producto = clienteRest.getForObject(url,
Producto.class,pathVariables);
		return new Item(producto, cantidad);
	}

}
```

8. Modifica la clase `ItemServiceImpl`

```java
package com.cognos.app.items.model.service;

import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

import com.cognos.app.items.model.Item;
import com.cognos.app.items.model.Producto;
```

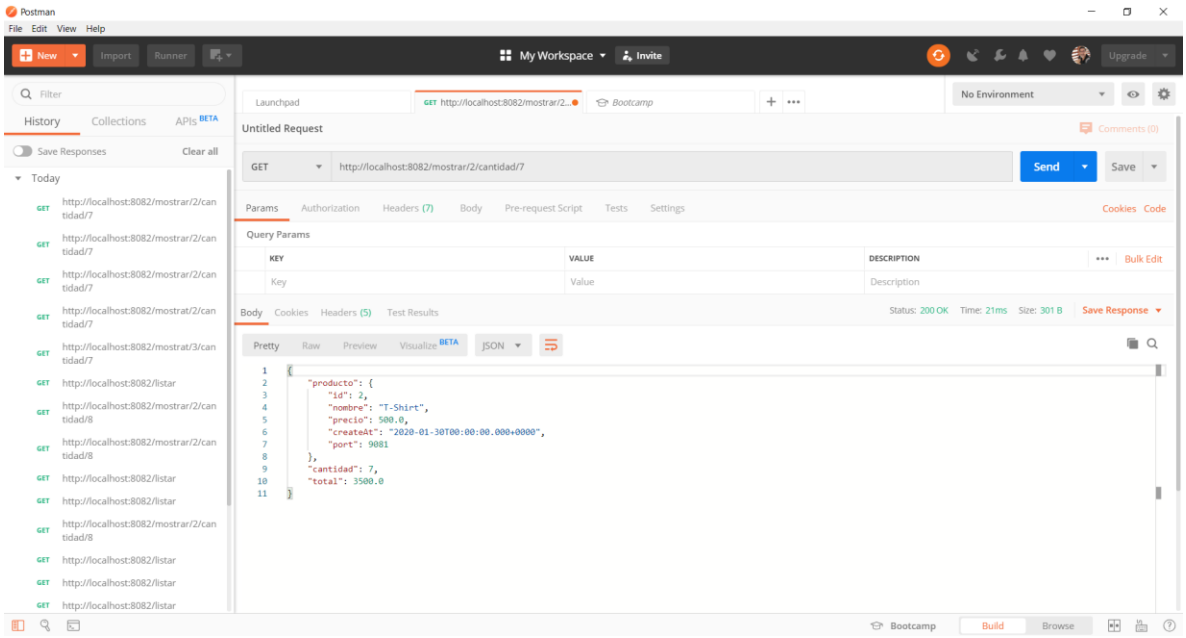
```
@Service("serviceRestTemplate")
public class ItemServiceImpl implements ItemService {

	@Autowired
	private RestTemplate clienteRest;
	@Override
	public List<Item> findAll() {
		String url="http://servicio-productos/listar";
		List<Producto> productos =
Arrays.asList(clienteRest.getForObject(url, Producto[].class));
		List<Item> items =  productos.stream().map(p -> new
Item(p,1)).collect(Collectors.toList());
		return items;
	}

	@Override
	public Item findbyId(Long id, Integer cantidad) {
		Map<String, String> pathVariables = new HashMap<String,String>();
		pathVariables.put("id",id.toString());
		String url = "http://servicio-productos/mostrar/{id}";
		Producto producto = clienteRest.getForObject(url,
Producto.class,pathVariables);
		return new Item(producto, cantidad);
	}

}
```

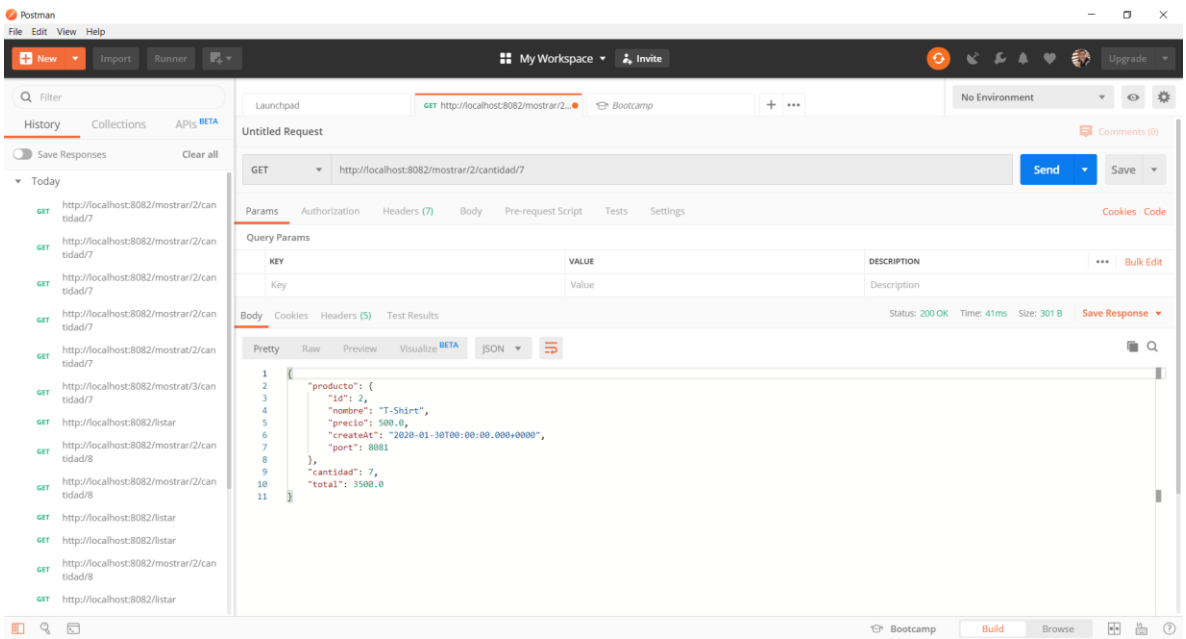
9. Edita clase **ItemController** del proyecto **servicio-items**

```
package com.cognos.app.items.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RestController;

import com.cognos.app.items.model.Item;
import com.cognos.app.items.model.service.ItemService;

@RestController
public class ItemController {
    @Autowired
    @Qualifier("serviceRestTemplate")
      private ItemService itemService;

    @GetMapping("/listar")
    public List<Item> listar(){
      return itemService.findAll();
    }
```

```
    @GetMapping("/mostrar/{id}/cantidad/{cantidad}")
    public Item detalle(@PathVariable Long id,@PathVariable Integer cantidad) {
       return itemService.findbyId(id, cantidad);
    }
}
```

10. Reiniciar las instancias de todos los servicios y probar con postman





Nota Importante:

En la clase ProductoClienteRest.java, remover la url de la anotación **@FeignClient**

```
package com.cognos.app.items.cliente;
import java.util.List;
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;

import com.cognos.app.items.model.Producto;

@FeignClient(name = "servicio-productos",url = "http://localhost:8081")
public interface ProductoClienteRest {

        @GetMapping("/listar")
        public List<Producto> listar();

        @GetMapping("/mostrar/{id}")
    public Producto mostrar(@PathVariable Long id);
}
```

Debe quedar así:

```
package com.cognos.app.items.cliente;
import java.util.List;
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import com.cognos.app.items.model.Producto;

@FeignClient(name = "servicio-productos"
public interface ProductoClienteRest {

        @GetMapping("/listar")
        public List<Producto> listar();

        @GetMapping("/mostrar/{id}")
         public Producto mostrar(@PathVariable Long id);
}
```