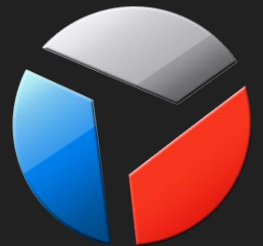


Payara (Micro) + Oracle JET

Diego Silva Límaco
[@apuntesdejava](#)



Dirigido a

- Backend developer
- Frontend developer
- Java Web Developer

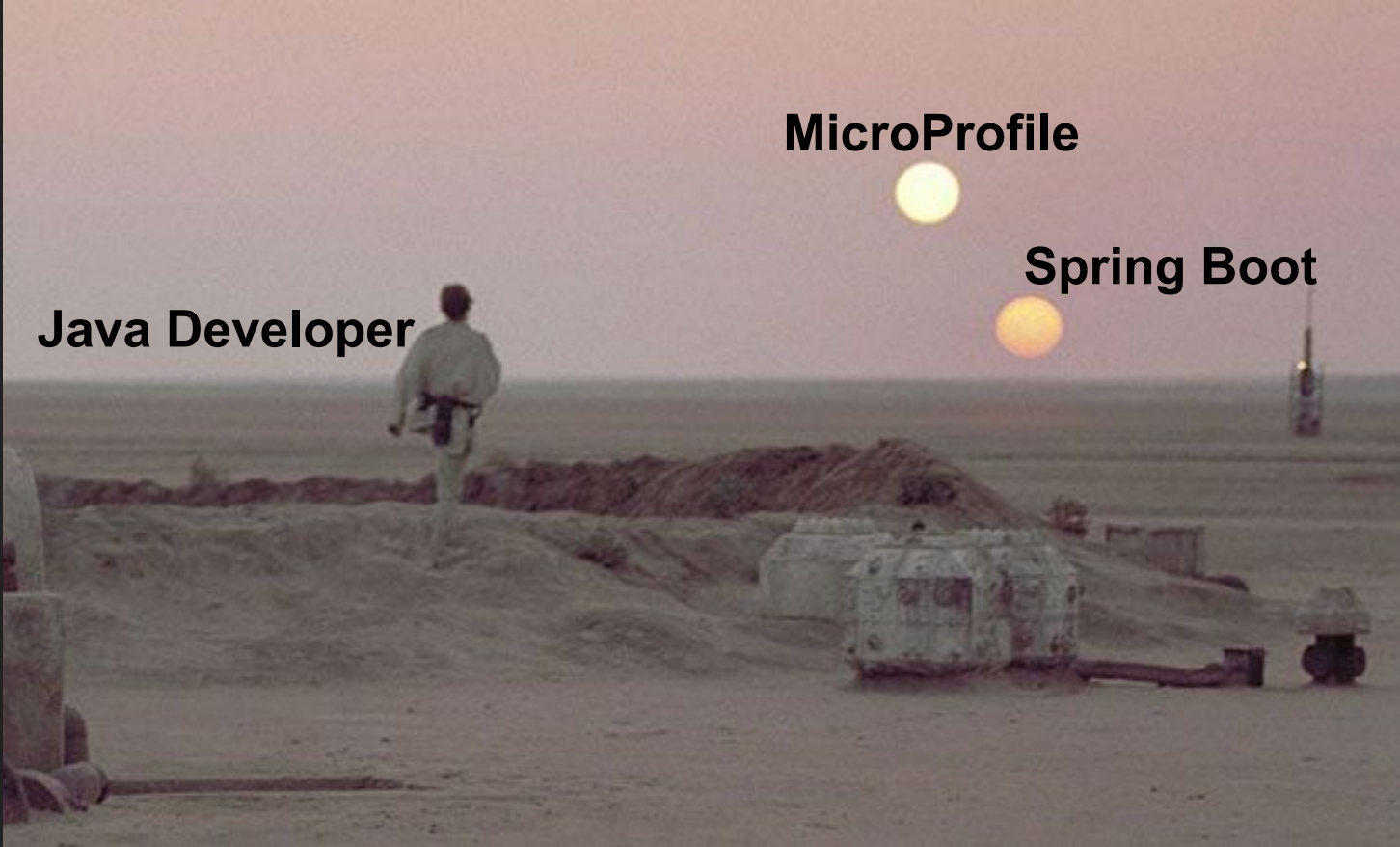
Agenda

- Conocer la creación de servicios REST con Java (Jakarta EE)
- Despliegue de la aplicación en Payara Micro
- Creación de una aplicación Web con Oracle JET
- Consumir los servicios REST de Payara Micro con Oracle JET



Payara Micro

- Versión Microservicios de Payara Server
- Preparado para entornos Cloud
- Compatible con Eclipse MicroProfile 1.2
- Pesa menos de 70MB
- Ejecuta .WAR desde la línea de comandos.



Java Developer

MicroProfile

Spring Boot

Proyecto Microservicio - Dependencias

```
<dependency>
  <groupId>javax</groupId>
  <artifactId>javaee-api</artifactId>
  <version>8.0</version>
  <scope>provided</scope>
</dependency>
```

```
<dependency>
  <groupId>org.eclipse.microprofile</groupId>
  <artifactId>microprofile</artifactId>
  <version>1.3</version>
  <type>pom</type>
  <scope>provided</scope>
</dependency>
```

```
<dependency>
  <groupId>fish.payara.extras</groupId>
  <artifactId>payara-micro</artifactId>
  <version>5.182</version>
  <scope>provided</scope>
</dependency>
```

Proyecto Microservicio - Clase Base REST

```
package com.apuntesdejava.microprofile;

import javax.ws.rs.ApplicationPath;
import javax.ws.rs.core.Application;

@ApplicationPath("/")
public class AppServiceConfig extends Application{

}
```

Proyecto Microservicio - Ejemplo Endpoint

```
@Path("hello")
public class HelloEndpoint {

    @GET
    @Path("my.name")
    @Produces(MediaType.TEXT_PLAIN)
    public Response sayHi(@QueryParam("name") String name) {
        String message = "Hola_" + name;
        return Response.ok(message).build();
    }
}
```


Proyecto Microservicio - JPAProducer

```
@ApplicationScoped
public class JPAProducer {           I
    @Produces
    @PersistenceContext
    private EntityManager em;
}
```

Proyecto Microservicio - Repository

```
@ApplicationScoped
public class PersonRepository {

    @Inject
    private EntityManager em;

    public List<Person> findAll() {
        return em.createQuery("select c from Person c", Person.class)
            .getResultList();
    }

    public Person findById(Long id) {
        return em.find(Person.class, id);
    }
}
```

Proyecto Microservicio - PersonEndpoint

```
@Path("person")
@Produces(MediaType.APPLICATION_JSON)
@RequestScoped
public class PersonEndpoint {

    @Inject
    private PersonRepository personRepository;

    @GET
    public Response findAll() {
        List<Person> list = personRepository.findAll();
        return Response.ok(list).build();
    }
}
```

Proyecto Microservicio - persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1"
  xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
    http://xmlns.jcp.org/xml/ns/persistence/persistence\_2\_1.xsd">
  <persistence-unit name="examplePU" transaction-type="JTA">
    <jta-data-source>jdbc/travel</jta-data-source>
    <shared-cache-mode>ALL</shared-cache-mode>
    <properties>
      <property name="javax.persistence.schema-generation.database.action"
        value="create"/>
    </properties>
  </persistence-unit>
</persistence>
```

Proyecto Microservicio - DataSource

```
create-jdbc-connection-pool --ping=true --pooling=true\  
  --restype=javax.sql.XADataSource\  
  --datasourceclassname=${ENV=JDBC_DRIVER}\  
  --property  
serverTimeZone=America/Lima:useSSL=false:characterEncoding=UTF8:Password=${ENV=JDBC  
_PASSWORD}:User=${ENV=JDBC_USER}:DatabaseName=${ENV=JDBC_DATABASE}:ServerName=${ENV  
=JDBC_HOST} travelPool  
  
create-jdbc-resource --connectionpoolid travelPool jdbc/travel
```

Proyecto Microservicio - execute

```
set JDBC_HOST=localhost
set JDBC_USER=travel
set JDBC_DATABASE=travel
set JDBC_PASSWORD=travel
set JDBC_DRIVER=com.mysql.cj.jdbc.MySQLXDataSource
```

```
java -classpath
target\mysql-connector-java.jar;target\payara-micro.jar
fish.payara.micro.PayaraMicro\
  --deploy target\microservice-example.war\
  --postbootcommandfile post-boot-commands.txt
```

Payara Micro - Demo



Oracle JET

- JavaScript Extension Toolkit
- Dirigido a desarrolladores de JavaScript de nivel intermedio a avanzado Frontend.
- Colección de bibliotecas de JavaScript de código abierto y de Oracle
- Simplicidad y eficiencia en la creación de aplicaciones.



ORACLE
CORDOVA™

Oracle JET - Instalación

```
npm install -g @oracle/ojet-cli
```

Oracle JET - Creando aplicación

```
ojet create <app name>  
--template=navdrawer || navbar || basic || blank
```

Convirtiendo a híbrido

```
npm install -g cordova  
  
ojet add hybrid  
ojet add platform android
```

Oracle JET - ¿Qué se puede hacer?

Controls

Buttons

Enabled Disabled

Conveyor Belt

Carbon Helium Lit >

File Picker

Drop files here or click to upload

Film Strip

Car Bus >

Image



Menus

Search
Calendar
Clock

Messages

ORACLE

- ERROR
Database replication failed
Error code 6501
- CONFIRMATION
New Hire Request
Request for office space completed

Progress

0% 100%

Progress List

item-p

item-d

item-e

Tags

h1 header
h2 header
h3 header
h4 header

Toolbar



Train

Step One Step Two St

Oracle JET - Preparando listado (Knockout JS)

```
self.persons = ko.observableArray();
```

```
//...
```

```
$.getJSON("http://localhost:8080/microservice-example/person", (data) => {  
    self.persons(data);  
});
```

Oracle JET - Preparando listado (oj.ArrayDataProvider)

```
self.dataprovider = new oj.ArrayDataProvider(  
    self.persons, {  
        keyAttributes: 'personId'  
    }  
);
```

Oracle JET - Preparando listado (HTML)

```
<oj-table data='[[dataprovder]]'  
  style='width: 100%; height:100%;'  
  columns=' [{"headerText":"ID","field":"personId"},  
            {"headerText":"Nombre", "field":"name"},  
            {"headerText":"Act.", "field":"lastUpdated"}]'  
>  
  
</oj-table>
```

Oracle JET - Demo

Conclusiones

Payara Micro:

1. ~~Java~~-Jakarta EE
2. Tiny war
3. Ejecutable en Heroku, Docker, Standalone

Oracle JET

1. JavaScript puro (no lenguajes nuevos)
2. OpenSource
3. Web Responsive
4. Híbrido
5. Buenas prácticas UI

Muchas gracias



/apuntesdejava



/apuntesdejava



@apuntesdejava



/in/diegosilval