

## Desenvolupament d'Aplicacions Web

### Mòdul 7 – Desenvolupament web en entorn servidor

## EAC5

(Curs 2021–22 / 1r semestre)

#### Solució

**En aquest document apareixen les respostes de les preguntes de resposta oberta. Les respostes de les tipus text ja apareixen al qüestionari. L'ordre pot ser diferent en cada cas**

#### Exercici 1:

Heu d'implementar un servei SOAP que ens permeti consultar la cartellera d'un teatre.

Ha d'implementar els mètodes:

```
getAllEspectacles();  
getEspectacle(String espectacleID);  
getEntradesVenudes(String espectacleID);  
comprar(String espectacleID, int numEntrades);
```


Cal crear els fitxers d'interfície i la classe que l'implementi. En la classe, podeu assumir que teniu ja creada una classe repositori anomenada `CartelleraRepository`, que és un `InMemoryRepository`

//interfície

```
import cat.xtec.ioc.domain.Espectacle;  
import java.util.ArrayList;  
import javax.jws.WebMethod;  
import javax.jws.WebService;  
import javax.jws.soap.SOAPBinding;
```

```
@WebService  
@SOAPBinding(style = SOAPBinding.Style.DOCUMENT)
```

```
public interface CartelleraServiceEndpoint {  
    @WebMethod ArrayList<Espectacle> getAllEspectacles();  
    @WebMethod Espectacle getEspectacle(String espectacleID);  
    @WebMethod Espectacle getEntradesVenudes(String espectacleID);  
    @WebMethod Espectacle comprar(String espectacleID, int numEntrades);  
}
```

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 1 de 6
	Versió: 03	DAW_M07B2_EAC5_Solució_2122S1	Lliurament: 09/11/2021

//classe

```
import cat.xtec.ioc.service.Espectacle;
import javax.jws.WebService;
import cat.xtec.ioc.service.CartelleraServiceEndpoint ;
import java.util.ArrayList;

@WebService(serviceName = "CartelleraService", endpointInterface =
"cat.xtec.ioc.service.CartelleraServiceEndpoint ")

public class CartelleraServiceEndpointImpl implements CartelleraServiceEndpoint {
    private final InMemoryCartelleraRepository cartelleraRepository = new InMemoryCartelleraRepository();

    @Override
    public ArrayList<Espectacle> getAllEspectacles() {
        return new ArrayList<Espectacle>(this.cartelleraRepository.getAllEspectacles());
    }

    @Override
    public Espectacle getEspectacle(String spectacleID) {
        return this.cartelleraRepository.getEspectacle(spectacleID);
    }

    @Override
    public Espectacle getEntradesVenudes(String spectacleID) {
        return this.cartelleraRepository.getEntradesVenudes(spectacleID);
    }


    @Override
    public Espectacle comprar(String gameID, int numEntrades) {
        this.cartelleraRepository.comprar(gameID, numEntrades);
        return getEspectacle(spectacleID);
    }
}
```

## Exercici 2:

Del webservice anterior, caldrà que pugeu en aquesta tasca el WSDL de contracte del servei.

Copieu aquí o pugeu el fitxer

```
<?xml version='1.0' encoding='UTF-8'?>
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://impl.service.ioc.xtec.cat/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://impl.service.ioc.xtec.cat/" name="CartelleraService">
<import namespace="http://service.ioc.xtec.cat/" location="http://localhost:8080/ioc/CartelleraService?wsdl=1"/>
<binding xmlns:ns1="http://service.ioc.xtec.cat/" name="CartelleraJocServiceEndpointImplPortBinding"
type="ns1:CartelleraServiceEndpoint">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="getEntradesVenudes">
        <soap:operation soapAction=""/>
    </operation>
</binding>
</definitions>
```

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 2 de 6
	Versió: 03	DAW_M07B2_EAC5_Solució_2122S1	Lliurament: 09/11/2021


```

        <input>
            <soap:body use="literal"/>
        </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
<operation name="getAllEspectacles">
    <soap:operation soapAction=""/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
<operation name="getEspectacle">
    <soap:operation soapAction=""/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
<operation name="comprar">
    <soap:operation soapAction=""/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
</binding>
<service name="CartelleraService">
    <port name="CartelleraServiceEndpointImplPort"
binding="tns:CartelleraServiceEndpointImplPortBinding">
        <soap:address location="http://localhost:8080/ioc/CartelleraService"/>
    </port>
</service>
</definitions>

```

## Exercici 3:

Prepareu un servei REST en una classe que ens permeti consultar la llista d'espectacles d'un teatre, cercar-los per nom, saber-ne les entrades venudes d'un en concret, fer-ne una compra (modificar el número d'entrades venudes + 1) i afegir-ne un espectacle nou a la cartellera.

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 3 de 6
	Versió: 03	DAW_M07B2_EAC5_Solució_2122S1	Lliurament: 09/11/2021

Tot el que necessiteu (ex: repositoris InMemory...) ho podeu donar per implementat i no cal que ho mostreu, però ho haureu d'usar amb criteri.

```
@Path("/cartellera")
@Stateless
public class CartelleraRestService {
    private CartelleraRepository cartelleraRepository = new InMemoryCartelleraRepository ( );

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public List<Espectacle> getAll ( ) {
        return this.cartelleraRepository.getAll( );
    }

    @GET
    @Path("/{nom}")
    @Produces(MediaType.APPLICATION_JSON)
    public Espectacle findByName(@PathParam("nom") String nom) {
        return this.cartelleraRepository.get ( nom ) ;
    }

    @GET
    @Path("getEntradesVenudes/{nom}")
    @Produces(MediaType.APPLICATION_JSON)
    public Espectacle getEntradesVenudes(@PathParam("nom") String nom) {
        return this.cartelleraRepository.getEntradesVenudes ( nom ) ;
    }

    @PUT
    @Path("/{nom}")
    @Consumes(MediaType.APPLICATION_JSON)
    public void comprarEntrada(@PathParam("nom") String nom) {
        this.cartelleraRepository.comprarEntrada(nom) ;
    }

    @POST
    @Consumes(MediaType.APPLICATION_JSON)
    public void afegirEspectacle(Espectacle espectacle) {
        this.cartelleraRepository.afegirEspectacle(espectacle);
    }
}
```

## Exercici 4:

Assumint que la classe **Espectacle** té els següents atributs:

**String** nomEspectacle;

**String** nomTeatre;

**int** capacitat;

**int** entradesVenudes;

aconseguir la sortida JSON que ens retornarà el servei si executem GET a les URL's **"/cartellera"** i **"/cartellera/{nom}"** del servei REST implementat

GET /cartellera (tindrà més o menys segons el número d'espectacles que hagueu afegit):

```
[{"nomEspectacle": "Mar i Cel", "nomTeatre": "Victoria", "capacitat": "2000", "entradesVenudes": "200"},
{"nomEspectacle": "El llibertí", "nomTeatre": "Poliorama", "capacitat": "1000", "entradesVenudes": "530"},
{"nomEspectacle": "El Joc del Calamar", "nomTeatre": "Cafè teatre Llantiol", "capacitat": "440",
"entradesVenudes": "780"}]
```

```
GET /cartellera/{nom}: {"nomEspectacle": nom, "nomTeatre": "Liceu", "capacitat": "10000",
"entradesVenudes": "100"}
```

.....

## Exercici 5:

Escriure un possible client en Java per a un servei REST publicat a la URL

**http://localhost:8080/consultaCartellera/rest/**

que aconseguixi la llista de tots els espectacles de la cartellera. (**/cartellera**)

Amb quina API treballem?

```
import java.net.URI;
import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Invocation;
import javax.ws.rs.client.WebTarget;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.UriBuilder;
```

```
public class Cliente {
```

```
    private static final Client CLIENT = ClientBuilder.newClient();
```

```
    public static void main(String[] args) {
```

```
        URI uri = UriBuilder.fromUri("http://localhost:8080/consultaCartellera/rest/
cartellera").port(8080).build();
```

```
        WebTarget target = CLIENT.target(uri);
```

```
        Invocation invocation = target.request(MediaType.APPLICATION_JSON).buildGet();
```

```
        Response res = invocation.invoke();
```

```
        List<Espectacle> returnedList = res.readEntity(new GenericType<List<Espectacle>>() {});
```


```
        System.out.println("GetStatus of response: " + res.getStatus());
```

```
        // TODO
```

```
        fer alguna cosa amb la llista, mostrar-la, per exemple
```

```
    }
}
```

La api utilitzada es JAX-RS

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 5 de 6
	Versió: 03	DAW_M07B2_EAC5_Solució_2122S1	Lliurament: 09/11/2021

## Exercici 6:

**Escriuiu com afegirieu un nou Espectacle al sistema utilitzant el servei REST i el verb POST**

Es pot utilitzar una eina que permeti fer peticions HTTP, com per exemple CURL.

Caldria executar el següent:

```
curl -H "Content-Type: application/json" -X POST -d '{"nomEspectacle":"Cegada de amor","nomTeatre":" Teatre Tívoli","capacitat":"1500", "entradesVenudes":"458"}' http://localhost:8080/consultaCartellera/rest/cartellera
```