

Començat el	diumenge, 31 octubre 2021, 10:42
Estat	Acabat
Completat el	dimarts, 9 novembre 2021, 11:54
Temps emprat	9 dies 1 hora
Qualificació	8,00 sobre 10,00 (80%)

CFGS Desenvolupament d'Aplicacions Web

Mòdul 7 B2 – Desenvolupament web en entorn servidor

UF4 – Serveis web. Pàgines dinàmiques interactives. Webs Híbrids

Unitat 5 – Serveis web amb Java Enterprise Edition 7

EAC5

(Curs 2021–22 / 1r semestre)

Presentació i resultats d'aprenentatge

Aquest exercici d'avaluació continuada (EAC) es correspon amb els continguts treballats a la unitat 5 “**Serveis web amb Java Enterprise Edition 7**”

Els **resultats d'aprenentatge** que es plantegen són:

- Desenvolupa serveis web analitzant el seu funcionament i implantant l'estructura dels seus components.

Criteris d'avaluació

La puntuació màxima assignada a cada activitat s'indica a l'enunciat.

Els criteris que es tindran en compte per avaluar el treball de l'alumnat són els següents:

- Identifica les característiques pròpies i l'àmbit d'aplicació dels serveis web.
- Reconeix els avantatges d'utilitzar serveis web per proporcionar accés a funcionalitats incorporades a la lògica de negoci d'una aplicació.
- Determina les tecnologies i els protocols implicats en la publicació i utilització de serveis web.
- Programa un servei web.
- Crea el document de descripció del servei web.
- Verifica el funcionament del servei web.

- Consumeix el servei web.

Forma de lliurament

Us demanem que, a part de contestar aquest qüestionari, copieu totes les preguntes i respostes en un document *.odt.

En aquest document hi heu d'afegir totes les preguntes i respostes de TOTS els qüestionaris de l'EAC en curs i, finalment, entregar-ho a la tasca EACx corresponent.

(si copieu les respostes al document sempre tindreu una còpia en cas que s'acabi la sessió del navegador també. ;D)

1

Puntuació 3,00 sobre 3,00

Completa

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

Ha d'implementar els mètodes:

```
getAllEspectacles();
getEspectacle(String spectacleID);
getEntradesVenudes(String spectacleID);
comprar(String spectacleID, 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

//Service End Point Implements:

```
package cat.xtec.ioc.service.impl;
import cat.xtec.ioc.domain.Espectacle;
import cat.xtec.ioc.domain.repository.CartelleraRepository;
import cat.xtec.ioc.domain.repository.impl.InMemoryRepository;
import cat.xtec.ioc.service.EspectacleServiceEndpoint;
import java.util.ArrayList;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@WebService(serviceName = "EspectacleServiceEndpointImpl",
            endpointInterface = "cat.xtec.ioc.service.EspectacleServiceEndpoint")
public class EspectacleServiceEndpointImpl implements EspectacleServiceEndpoint {

    private final CartelleraRepository cartelleraRepository = new InMemoryRepository();

    public ArrayList<Espectacle> getAllEspectacles() {
return new ArrayList<Espectacle>(this.cartelleraRepository.getAllEspectacles());    }
    public Espectacle getEspectacle(String spectacleID) {

        return cartelleraRepository.getEspectacle(spectacleID);
    }
    public ArrayList<Espectacle> comprar(String spectacleID, int numEntrades) {
        return cartelleraRepository.comprar(spectacleID, numEntrades);
    }
    public int numEntradesVenudesEspectacle(String string) {
return cartelleraRepository.numEntradesVenudesEspectacle(string);    }
```

```
return cat.getEntradesVenudesEspectacle(String);  
}  
}
```

//Service End Point

```
package cat.xtec.ioc.service;  
import cat.xtec.ioc.domain.Espectacle;  
import java.util.ArrayList;  
import javax.ws.WebMethod;  
import javax.ws.WebService;  
import javax.ws.soap.SOAPBinding;  
  
@WebService  
@SOAPBinding(style = SOAPBinding.Style.DOCUMENT)  
public interface EspectacleServiceEndpoint {  
  
    @WebMethod ArrayList<Espectacle> getAllEspectacles();  
    @WebMethod Espectacle getEspectacle(String spectacleID);  
    @WebMethod int numEntradesVenudesEspectacle(String nomEspectacle);  
    @WebMethod ArrayList<Espectacle> comprar(String spectacleID, int numEntrades);  
}
```

//Domini Espectacle:

```
package cat.xtec.ioc.domain;  
  
public class Espectacle {  
  
    private String EspectacleID;  
    private String nomEspectacle;  
    private String nomTeatre;  
    private int capacitat;  
    private int entradesVenudes;  
    public Espectacle() {  
    }  
    public Espectacle(String idEspectacle,String nomEspectacle, String nomTeatre, int capacitat, int  
entradesVenudes) {  
        this.EspectacleID=idEspectacle;  
        this.nomEspectacle = nomEspectacle;  
        this.nomTeatre = nomTeatre;  
        this.capacitat = capacitat;  
        this.entradesVenudes = entradesVenudes;  
    }  
    public String getEspectacleID() {  
        return EspectacleID;  
    }  
    public void setEspectacleID(String EspectacleID) {  
        this.EspectacleID = EspectacleID;  
    }  
    public String getNomEspectacle() {  
        return nomEspectacle;  
    }  
    public void setNomEspectacle(String nomEspectacle) {  
        this.nomEspectacle = nomEspectacle;  
    }  
    public String getNomTeatre() {  
        return nomTeatre;  
    }  
    public void setNomTeatre(String nomTeatre) {  
        this.nomTeatre = nomTeatre;  
    }  
    public int getCapacitat() {  
        return capacitat;  
    }  
    public void setCapacitat(int capacitat) {  
        this.capacitat = capacitat;  
    }  
    public int getEntradesVenudes() {  
        return entradesVenudes;  
    }  
}
```

```

    }

    public void setEntradesVenudes(int entradesVenudes) {
        this.entradesVenudes = entradesVenudes;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
        final Espectacle other = (Espectacle) obj;
        if ((this.EspectacleID == null) ? (other.EspectacleID != null) :
!this.EspectacleID.equals(other.EspectacleID)) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "Espectacle{" + "EspectacleID=" + EspectacleID + ", nomEspectacle=" + nomEspectacle + ",
nomTeatre=" + nomTeatre + ", capacitat=" + capacitat + ", entradesVenudes=" + entradesVenudes + '}';
    }
}

```

//In memory Implements:

```

package cat.xtec.ioc.domain.repository.impl;
import cat.xtec.ioc.domain.Espectacle;
import java.util.ArrayList;
import java.util.Collection;
import java.util.HashMap;
import java.util.Map;
import org.springframework.stereotype.Repository;
import cat.xtec.ioc.domain.repository.CartelleraRepository;
import java.util.List;

@Repository
public class InMemoryRepository implements CartelleraRepository {
    private final Map<String, Espectacle> espectacles = new HashMap<String, Espectacle>();

    public InMemoryRepository() {

        Espectacle espectacle1 = new Espectacle("111", "Gala ballarins catalans al món","Teatre_Girona", 400,200);
        Espectacle espectacle2 = new Espectacle("222", "Vals de Mefisto", "Teatre_Barcelona",200,100);
        Espectacle espectacle3 = new Espectacle("333","El barber de Sevilla de
Rossini","Teatre_Granollers",100,50);
        Espectacle espectacle4 = new Espectacle("444","El barber de Sevilla de Rossini","Teatre_Girona",400,200);

        espectacles.put("111", espectacle1);
        espectacles.put("222", espectacle2);
        espectacles.put("333", espectacle3);
        espectacles.put("444", espectacle4);

    }

    public Collection<Espectacle> getAllShows() {
        return espectacles.values();
    }

    public Collection<Espectacle> getAllEspectacles() {
        return espectacles.values();
    }

    public Espectacle getEspectacle(String espectacleID) {
        return espectacles.get(espectacleID);
    }

    public int numEntradesVenudesEspectacle(String idEspectacle) {

```

```
int result=0;
for (Espectacle spectacle : espectacles.values()) {
    if(spectacle.getEspectacleID().equals(idEspectacle)) {
        result=spectacle.getEntradesVenudes();
    }
}
return result;
}

public ArrayList<Espectacle> comprar(String spectacleID, int numEntrades) {
    ArrayList<Espectacle> result = new ArrayList<Espectacle>();
    for (Espectacle spectacle : espectacles.values()) {
        if(spectacle.getEspectacleID().equals(spectacleID)) {
            for (int i = 0; i < numEntrades; i++){
                spectacle.setEntradesVenudes(numEntrades);
                result.add(spectacle);
            }
        }
    }
    return result;
}
}
```

Comentari:

2

Puntuació 1,00 sobre 1,00

Completa

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

Copieu aquí o pugeu el fitxer

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.4.3 (RELEASE-2.4.3-ad3323c; 2019-01-21T17:46:06+0000) JAXWS-RI/2.3.2 JAXWS-API/2.3.2 JAXB-RI/2.3.2 JAXB-API/2.3.2 git-revision#unknown. -->
<!-- Generated by JAX-WS RI (https://github.com/eclipse-ee4j/metro-jax-ws). RI's version is Metro/2.4.3 (RELEASE-2.4.3-ad3323c; 2019-01-21T17:46:06+0000) JAXWS-RI/2.3.2 JAXWS-API/2.3.2 JAXB-RI/2.3.2 JAXB-API/2.3.2 git-revision#unknown. -->
<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="EspectacleServiceEndpointImpl">
<import namespace="http://service.ioc.xtec.cat/" location="http://localhost:8080/cat.xtec.ioc.espectacle/EspectacleServiceEndpointImpl?wsdl=1"/>
<binding xmlns:ns1="http://service.ioc.xtec.cat/" name="EspectacleServiceEndpointImplPortBinding" type="ns1:EspectacleServiceEndpoint">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<operation name="getAllEspectacles">
<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>
<operation name="getEspectacle">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="numEntradesVenudesEspectacle">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="EspectacleServiceEndpointImpl">
<port name="EspectacleServiceEndpointImplPort" binding="tns:EspectacleServiceEndpointImplPortBinding">
<soap:address location="http://localhost:8080/cat.xtec.ioc.espectacle/EspectacleServiceEndpointImpl"/>
</port>
</service>
</definitions>
```

Comentari:

3

Puntuació 1,00 sobre 1,00

Correcte

Indiqueu si és certa o falsa la següent afirmació:

La comunicació amb un webservice es realitza mitjançant missatges SOAP, que son XML amb una estructura determinada.

Trieu-ne una:

- ☒ Vertader ✓
- ☐ Fals

La resposta correcta és 'Vertader'.

Assumint que la classe Espectacle té els següents atributs:

```
String nomEspectacle;  
String nomTeatre;  
int capacitat;  
int entradesVenudes;
```

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

```
//Domini: Espectacle
```

```
package cat.xtec.ioc.domain;  
  
public class Espectacle {  
  
    private String nomEspectacle;  
    private String nomTeatre;  
    private int capacitat;  
    private int entradesVenudes;  
  
    public Espectacle(String nomEspectacle, String nomTeatre, int capacitat, int entradesVenudes) {  
        this.nomEspectacle = nomEspectacle;  
        this.nomTeatre = nomTeatre;  
        this.capacitat = capacitat;  
        this.entradesVenudes = entradesVenudes;  
    }  
    public String getNomEspectacle() {  
        return nomEspectacle;  
    }  
    public void setNomEspectacle(String nomEspectacle) {  
        this.nomEspectacle = nomEspectacle;  
    }  
    public String getNomTeatre() {  
        return nomTeatre;  
    }  
    public void setNomTeatre(String nomTeatre) {  
        this.nomTeatre = nomTeatre;  
    }  
    public int getCapacitat() {  
        return capacitat;  
    }  
    public void setCapacitat(int capacitat) {  
        this.capacitat = capacitat;  
    }  
    public int getEntradesVenudes() {  
        return entradesVenudes;  
    }  
    public void setEntradesVenudes(int entradesVenudes) {  
        this.entradesVenudes = entradesVenudes;  
    }  
}
```

```
//Repositori: EspectacleRepository  
package cat.xtec.ioc.domain.repository;  
  
import cat.xtec.ioc.domain.Espectacle;  
import java.util.List;
```

```

public interface EspectacleRepository {

    List<Espectacle> getAll();
    //void add(Espectacle espectacle);
    //void update(Espectacle espectacle);
    //void delete(String titul);
    Espectacle get(String titul);
    List<Espectacle> findByNom(String nom);

}

//Repositori.imp:InMemoryEspectacleRepository
package cat.xtec.ioc.domain.repository.impl;

import cat.xtec.ioc.domain.Espectacle;
import cat.xtec.ioc.domain.repository.EspectacleRepository;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class InMemoryEspectacleRepository implements EspectacleRepository {

    private final Map<String, Espectacle> shows = new HashMap<String, Espectacle>();

    public InMemoryEspectacleRepository() {
        Espectacle primer = new Espectacle("Espectacle_Catalunya","Teatre Catalunya",300, 200);
        Espectacle segon = new Espectacle("Espectacle_Girona","Teatre Girona",200, 100);
        Espectacle tercer = new Espectacle("Espectacle_Granollers","Teatre Granollers",100, 80);

        shows.put("Espectacle_Catalunya", primer);
        shows.put("Espectacle_Girona", segon);
        shows.put("Espectacle_Granollers", tercer);
    }

    public List<Espectacle> getAll() {
        return new ArrayList<Espectacle>(shows.values());
    }

    public Espectacle get(String nom) {
        return this.shows.get(nom);
    }

}

//service:ApplicationConfig
package cat.xtec.ioc.service;

import javax.ws.rs.core.Application;

@javax.ws.rs.ApplicationPath("rest")
public class ApplicationConfig extends Application {

}

//service:EspectacleRestService

package cat.xtec.ioc.service;

import cat.xtec.ioc.domain.Book;
import cat.xtec.ioc.domain.Espectacle;
import cat.xtec.ioc.domain.repository.EspectacleRepository;
import cat.xtec.ioc.domain.repository.impl.InMemoryBookRepository;
import cat.xtec.ioc.domain.repository.impl.InMemoryEspectacleRepository;
import java.util.List;
import javax.inject.Singleton;
import javax.ws.rs.GET;
import javax.ws.rs.NotFoundException;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;

```



```
import javax.ws.rs.core.Context;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.UriInfo;

@Path("/cartellera")
@Singleton
public class EspectacleRestService {

    @Context    private UriInfo uriInfo;

    private EspectacleRepository spectacleRepository = new InMemoryEspectacleRepository();

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

    @GET
    @Path("/{nom}")
    @Produces(MediaType.APPLICATION_JSON)
    public Response find(@PathParam("nom") String nom) {
        Espectacle spectacle = this.spectacleRepository.get(nom);
        if(spectacle == null) {
            throw new NotFoundException();
        }
        return Response.ok(spectacle).build();
    }
}
```

Comentari:

Es tractava de mostrar el JSON, no el codi.

Revisa la solució

5

Puntuació 1,00 sobre 1,00

Completa

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 en un número que li passem) i afegir-ne un de nou a la cartellera.

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.

```
//Domini Espectacle:
package cat.xtec.ioc.domain;

public class Espectacle {

    private String idEspectacle;
    private String nomEspectacle;
    private String nomTeatre;
    private int capacitat;
```

```
private int capacitat;
private int entradesVenudes;

    public Espectacle(String idEspectacle,String nomEspectacle, String nomTeatre, int capacitat, int
entradesVenudes) {
        this.idEspectacle=idEspectacle;
        this.nomEspectacle = nomEspectacle;
        this.nomTeatre = nomTeatre;
        this.capacitat = capacitat;
        this.entradesVenudes = entradesVenudes;
    }
    public String getIdEspectacle() {
        return idEspectacle;
    }
    public void setIdEspectacle(String idEspectacle) {
        this.idEspectacle = idEspectacle;
    }
    public String getNomEspectacle() {
        return nomEspectacle;
    }
    public void setNomEspectacle(String nomEspectacle) {
        this.nomEspectacle = nomEspectacle;
    }
    public String getNomTeatre() {
        return nomTeatre;
    }
    public void setNomTeatre(String nomTeatre) {
        this.nomTeatre = nomTeatre;
    }
    public int getCapacitat() {
        return capacitat;
    }
    public void setCapacitat(int capacitat) {
        this.capacitat = capacitat;
    }
    public int getEntradesVenudes() {
        return entradesVenudes;
    }
    public void setEntradesVenudes(int entradesVenudes) {
        this.entradesVenudes = entradesVenudes;
    }
}
```

```
//Damain repositori: EspectacleRepository
package cat.xtec.ioc.domain.repository;
import cat.xtec.ioc.domain.Espectacle;
import java.util.List;
public interface EspectacleRepository {

    List<Espectacle> getAll();
    Espectacle get(String titul);
    List<Espectacle> findByNomTeatre(String nom);
    List<Espectacle> findByNomEspectacle(String nom);
    int  numEntradesVenudesEspectacle(String nomEspectalce);
    List<Espectacle> comprarEspectacle(String nomEspectacle, int num);
    void add(Espectacle espectacle);

}
```

```
//Application Config
```

```
package cat.xtec.ioc.service;
import javax.ws.rs.core.Application;
@javax.ws.rs.ApplicationPath("rest")
public class ApplicationConfig extends Application {
```

}
//EspectacleRestService

```
package cat.xtec.ioc.service;
import cat.xtec.ioc.domain.Espectacle;
import cat.xtec.ioc.domain.repository.EspectacleRepository;
```

```
import cat.xtec.ioc.domain.repository.EspectacleRepository;
import cat.xtec.ioc.domain.repository.impl.InMemoryEspectacleRepository;
import java.net.URI;
import java.util.List;
import javax.inject.Singleton;
import javax.ws.rs.BadRequestException;
import javax.ws.rs.Consumes;
import javax.ws.rs.GET;
import javax.ws.rs.NotFoundException;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.core.Context;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.UriInfo;
@Path("/cartellera")
@Singleton
public class EspectacleRestService {
    @Context private UriInfo uriInfo;

    private EspectacleRepository spectacleRepository = new InMemoryEspectacleRepository();

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

    @GET
    @Path("/{nom}")
    @Produces(MediaType.APPLICATION_JSON)
    public Response find(@PathParam("nom") String nom) {
        Espectacle spectacle = this.spectacleRepository.get(nom);
        if(spectacle == null) {
            throw new NotFoundException();
        }
        return Response.ok(spectacle).build();
    }

    //llista d'espectacles d'un teatre
    @GET
    @Path("findByNomTeatre/{nomTeatre}")
    @Produces(MediaType.APPLICATION_JSON)
    public List<Espectacle> findByNomTeatre(@PathParam("nomTeatre") String nomTeatre) {
        return this.spectacleRepository.findByNomTeatre(nomTeatre);
    }

    //cercar-los per nom espectacle
    @GET
    @Path("findByNomEspectacle/{nomEspectacle}")
    @Produces(MediaType.APPLICATION_JSON)
    public List<Espectacle> findByNomEspectacle(@PathParam("nomEspectacle") String nomEspectacle) {
        return this.spectacleRepository.findByNomEspectacle(nomEspectacle);
    }

    //comprarEspectacle, retorna un numero de espectacles (numero d'entrades comprades)
    //a partir del nom d'un espectacle
    @GET
    @Path("comprarEspectacle/{nomEspectacle}/{numEntrades}")
    //@Path("comprarEspectacle/{numEntrades}")
    @Produces(MediaType.APPLICATION_JSON)
    public List<Espectacle> comprarEspectacle(@PathParam("nomEspectacle") String nomEspectacle,
    @PathParam("numEntrades") int numEntrades) {
        return this.spectacleRepository.comprarEspectacle(nomEspectacle,numEntrades);
    }

    //numEntradesVenudesEspectacle. En aquest cas pel l'id, també podria se pel nom
    @GET
    @Path("numEntradesVenudesEspectacle/{idEspectacle}")
    @Produces(MediaType.APPLICATION_JSON)
    public int numEntradesVenudesEspectacle(@PathParam("idEspectacle") String idEspectacle) {
```

```
        public int numEntradesVenudesEspectacle(@PathParam("idEspectacle") String idEspectacle) {
            return this.espectacleRepository.numEntradesVenudesEspectacle(idEspectacle);
        }

        @POST
        @Consumes(MediaType.APPLICATION_JSON)
        public Response create(Espectacle spectacle) {
            if(spectacle == null) {
                throw new BadRequestException();
            }
            this.espectacleRepository.add(spectacle);

            URI bookUri = uriInfo.getAbsolutePathBuilder().path(spectacle.getIdEspectacle()).build();
            return Response.ok(bookUri).build();
        }
    }
}
```

// Tot i que l'enunciat no el demana

```
// InMemoryEspectacleRepository
```

```
package cat.xtec.ioc.domain.repository.impl;
import cat.xtec.ioc.domain.Espectacle;
import cat.xtec.ioc.domain.repository.EspectacleRepository;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
public class InMemoryEspectacleRepository implements EspectacleRepository {

    private final Map<String, Espectacle> shows = new HashMap();

    public InMemoryEspectacleRepository() {
        Espectacle primer = new Espectacle("111","Espectacle_Catalunya","Teatre_Catalunya",300, 200);
        Espectacle segon = new Espectacle("222","Espectacle_Girona","Teatre_Girona",200, 100);
        Espectacle tercer = new Espectacle("333","Espectacle_Nacional","Teatre_Granollers",100, 80);
        Espectacle quart = new Espectacle("444","Espectacle_Nacional","Teatre_Catalunya",300, 200);
        Espectacle cinque = new Espectacle("555","Espectacle_Nacional","Teatre_Girona",300, 200);

        shows.put("111", primer);
        shows.put("222", segon);
        shows.put("333", tercer);
        shows.put("444", quart);
        shows.put("555", cinque);
    }

    public List<Espectacle> getAll() {
        return new ArrayList<Espectacle>(shows.values());
    }

    public Espectacle get(String nom) {
        return this.shows.get(nom);
    }

    public List<Espectacle> findByNomTeatre(String nomTeatre) {
        List<Espectacle> result = new ArrayList<Espectacle>();
        for (Espectacle spectacle : shows.values()) {
            if(spectacle.getNomTeatre().equals(nomTeatre)) {
                result.add(spectacle);
            }
        }
        return result;
    }

    public List<Espectacle> findByNomEspectacle(String nom) {
        List<Espectacle> result = new ArrayList<Espectacle>();
        for (Espectacle spectacle : shows.values()) {
            if(spectacle.getNomEspectacle().equals(nom)) {
                result.add(spectacle);
            }
        }
        return result;
    }

    public List<Espectacle> comprarEspectacle(String nomEspectacle, int numEntrades) {
        List<Espectacle> result = new ArrayList<Espectacle>();
    }
}
```

```
List<Espectacle> result = new ArrayList<Espectacle>();
for (Espectacle espectacle : shows.values()) {
    if(espectacle.getNomEspectacle().equals(nomEspectacle)) {
        for (int i = 0; i < numEntrades; i++){
            espectacle.setEntradesVenudes(numEntrades);
            result.add(espectacle);
        }
    }
}
return result;
}

public int numEntradesVenudesEspectacle(String idEspectacle) {
    int result=0;
    for (Espectacle espectacle : shows.values()) {
        if(espectacle.getIdEspectacle().equals(idEspectacle)) {
            result=espectacle.getEntradesVenudes();
        }
    }
    return result;
}

public void add(Espectacle espectacle) {
    this.shows.put(espectacle.getIdEspectacle(), espectacle);
}
}
```

Comentari:

6

Puntuació 1,00 sobre 1,00

Correcte

JAX-RS proveeix de les següents anotacions:

Trieu-ne una o més:

- ☐ a. @Product
- ☐ b. @Stateless
- ☐ c. @Service
- ☒ d. @Path
- ☐ @Path
- ☐ @Path
- ☒ @Path ✓
- ☒ e. @GET ✓

La resposta és correcta.

Les respostes correctes són: @Path

@Path

@Path

@Path, @GET

Escriviu 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 trebal·leu?

// Es mostra la llista de tots els espectacles de la cartellera. (/cartellera)

i de pas es comprova que funciona

//L'API amb la qual es treballa: JAX-RS

```
package cat.xtec.ioc.test;
```

```
import cat.xtec.ioc.domain.Espectacle;
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.net.URI;
import java.util.List;
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.GenericType;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.UriBuilder;
import static org.junit.Assert.assertTrue;
import org.junit.Test;
public class EspectacleRestServiceTest {
    private static final URI uri =
UriBuilder.fromUri("http://localhost:8080/cat.xtec.EAC5/rest/cartellera/").port(8080).build();
    private static final Client client = ClientBuilder.newClient();

    @Test
    public void shouldReturnAllEspectacles() throws JsonProcessingException {
URI uri = UriBuilder.fromUri("http://localhost:8080/cat.xtec.EAC5/rest/cartellera/").port(8080).build();
        WebTarget target = client.target(uri);
        Invocation invocation = target.request(MediaType.APPLICATION_JSON).buildGet();
        Espectacle primer = new Espectacle("111","Espectacle_Catalunya","Teatre_Catalunya",300, 200);
        Espectacle segon = new Espectacle("222","Espectacle_Girona","Teatre_Girona",200, 100);

        // Act
        Response res = invocation.invoke();
        List<Espectacle> returnedEpectacles = res.readEntity(new GenericType<List<Espectacle>>() {});

        // Assert
        assertTrue(returnedEpectacles.contains(primer));
        assertTrue(returnedEpectacles.contains(segon));
        //Shows
        //Es mostra tot el json de la pagina
        ObjectMapper mapper = new ObjectMapper();
        System.out.print(mapper.writerWithDefaultPrettyPrinter().writeValueAsString(returnedEpectacles));

        //Tots els espectacles:
        for(Espectacle spectacle: returnedEpectacles ){
            System.out.println(spectacle.getNomEspectacle());
        }
    }
}
```



```
package cat.xtec.ioc.domain;

public class Espectacle {

private String idEspectacle;
private String nomEspectacle;
private String nomTeatre;
private int capacitat;
private int entradesVenudes;
public Espectacle(){
}

    public Espectacle(String idEspectacle,String nomEspectacle, String nomTeatre, int capacitat, int entradesVenudes) {
        this.idEspectacle=idEspectacle;
        this.nomEspectacle = nomEspectacle;
        this.nomTeatre = nomTeatre;
        this.capacitat = capacitat;
        this.entradesVenudes = entradesVenudes;
    }
    public String getIdEspectacle() {
        return idEspectacle;
    }

    public void setIdEspectacle(String idEspectacle) {
        this.idEspectacle = idEspectacle;
    }
    public String getNomEspectacle() {
        return nomEspectacle;
    }
    public void setNomEspectacle(String nomEspectacle) {
        this.nomEspectacle = nomEspectacle;
    }
    public String getNomTeatre() {
        return nomTeatre;
    }
    public void setNomTeatre(String nomTeatre) {
        this.nomTeatre = nomTeatre;
    }
    public int getCapacitat() {
        return capacitat;
    }
    public void setCapacitat(int capacitat) {
        this.capacitat = capacitat;
    }
    public int getEntradesVenudes() {
        return entradesVenudes;
    }
    public void setEntradesVenudes(int entradesVenudes) {
        this.entradesVenudes = entradesVenudes;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
        final Espectacle other = (Espectacle) obj;
        if ((this.idEspectacle == null) ? (other.idEspectacle != null) : !this.idEspectacle.equals(other.idEspectacle)) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "Espectacle[" + "idEspectacle=" + idEspectacle + ", nomEspectacle=" + nomEspectacle + ", nomTeatre=" + nomTeatre + "
    }
```

```
return Espectacle{ + idEspectacle= + idEspectacle + , nomEspectacle= + nomEspectacle + , nomTeatre= + nomTeatre + ,
capacitat=" + capacitat + ", entradesVenudes=" + entradesVenudes + '}'};
    }
}
```

//Repository:

```
package cat.xtec.ioc.domain.repository;
import cat.xtec.ioc.domain.Espectacle;
import java.util.List;
public interface EspectacleRepository {

    List<Espectacle> getAll();

}
```

//Memori repository:

```
package cat.xtec.ioc.domain.repository.impl;
import cat.xtec.ioc.domain.Espectacle;
import cat.xtec.ioc.domain.repository.EspectacleRepository;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
public class InMemoryEspectacleRepository implements EspectacleRepository {

    private final Map<String, Espectacle> shows = new HashMap();

    public InMemoryEspectacleRepository() {
        Espectacle primer = new Espectacle("111","Espectacle_Catalunya","Teatre_Catalunya",300, 200);
        Espectacle segon = new Espectacle("222","Espectacle_Girona","Teatre_Girona",200, 100);
        Espectacle tercer = new Espectacle("333","Espectacle_Nacional","Teatre_Granollers",100, 80);
        Espectacle quart = new Espectacle("444","Espectacle_Nacional","Teatre_Catalunya",300, 200);
        Espectacle cinque = new Espectacle("555","Espectacle_Nacional","Teatre_Girona",300, 200);

        shows.put("111", primer);
        shows.put("222", segon);
        shows.put("333", tercer);
        shows.put("444", quart);
        shows.put("555", cinque);
    }

    public List<Espectacle> getAll() {
        return new ArrayList<Espectacle>(shows.values());
    }
}
```

//Aplication Config:

```
package cat.xtec.ioc.service;
```

```
import javax.ws.rs.core.Application;
@javax.ws.rs.ApplicationPath("rest")
public class ApplicationConfig extends Application {
}
```

//Rest Service:

```
package cat.xtec.ioc.service;
import cat.xtec.ioc.domain.Espectacle;
import cat.xtec.ioc.domain.repository.EspectacleRepository;
import cat.xtec.ioc.domain.repository.impl.InMemoryEspectacleRepository;
import java.net.URI;
import java.util.List;
import javax.inject.Singleton;
import javax.ws.rs.BadRequestException;
import javax.ws.rs.Consumes;
import javax.ws.rs.GET;
import javax.ws.rs.NotFoundException;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
```

```
import javax.ws.rs.Produces;
import javax.ws.rs.core.Context;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.UriInfo;
@Path("/cartellera")
@Singleton
public class EspectacleRestService {
    @Context private UriInfo uriInfo;

    private EspectacleRepository spectacleRepository = new InMemoryEspectacleRepository();

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

Comentari:

8

Puntuació 0,00 sobre 1,00

Completa

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

//S'afegirieu un nou Espectacle al sistema utilitzant el servei REST i el verb POST

// i de pass es comprova que funciona un test

```
package cat.xtec.ioc.test;

import cat.xtec.ioc.domain.Espectacle;
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.net.URI;
import java.util.List;
import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.client.Invocation;
import javax.ws.rs.client.WebTarget;
import javax.ws.rs.core.GenericType;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.UriBuilder;
import static org.junit.Assert.assertTrue;
import static org.junit.Assert.assertEquals;
import org.junit.Test;
public class EspectacleRestServiceTest {
    private static final URI uri =
UriBuilder.fromUri("http://localhost:8080/consultaCartellera/rest/").port(8080).build();
    private static final Client client = ClientBuilder.newClient();

    @Test
```

```
    public void createEspectacleShouldReturnTheURLToGetTheEspectacle() {
        // Arrange
        Espectacle spectacle = new Espectacle("666","Espectacle_Sise","Teatre_Sise",6000, 666);
        URI uri= UriBuilder.fromUri("http://localhost:8080/consultaCartellera/rest/").port(8080).build();
        WebTarget target = client.target(uri);
```

```
        newTarget(target).contentType(MediaType.APPLICATION_JSON).buildPost(Entity.entity(espectacle,
        MediaType.APPLICATION_JSON));

        // Act
        Response res = invocation.invoke();
        URI returnedUri = res.readEntity(URI.class);

        // Assert
        URI expectedUri =
        UriBuilder.fromUri("http://localhost:8080/consultaCartellera/rest/").port(8080).path("666").build();
        assertEquals(expectedUri, returnedUri);
    }}
}
```

//domain:

```
package cat.xtec.ioc.domain;
public class Espectacle {

private String idEspectacle;
private String nomEspectacle;
private String nomTeatre;
private int capacitat;
private int entradesVenudes;
public Espectacle(){
}

    public Espectacle(String idEspectacle,String nomEspectacle, String nomTeatre, int capacitat, int
    entradesVenudes) {
        this.idEspectacle=idEspectacle;
        this.nomEspectacle = nomEspectacle;
        this.nomTeatre = nomTeatre;
        this.capacitat = capacitat;
        this.entradesVenudes = entradesVenudes;
    }
    public String getIdEspectacle() {
        return idEspectacle;
    }

    public void setIdEspectacle(String idEspectacle) {
        this.idEspectacle = idEspectacle;
    }
    public String getNomEspectacle() {
        return nomEspectacle;
    }
    public void setNomEspectacle(String nomEspectacle) {
        this.nomEspectacle = nomEspectacle;
    }
    public String getNomTeatre() {
        return nomTeatre;
    }
    public void setNomTeatre(String nomTeatre) {
        this.nomTeatre = nomTeatre;
    }
    public int getCapacitat() {
        return capacitat;
    }
    public void setCapacitat(int capacitat) {
        this.capacitat = capacitat;
    }
    public int getEntradesVenudes() {
        return entradesVenudes;
    }
    public void setEntradesVenudes(int entradesVenudes) {
        this.entradesVenudes = entradesVenudes;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null) {
```

```
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Espectacle other = (Espectacle) obj;
    if ((this.idEspectacle == null) ? (other.idEspectacle != null) :
!this.idEspectacle.equals(other.idEspectacle)) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Espectacle{" + "idEspectacle=" + idEspectacle + ", nomEspectacle=" + nomEspectacle + ",
nomTeatre=" + nomTeatre + ", capacitat=" + capacitat + ", entradesVenudes=" + entradesVenudes + '}';
}

}
```

//Repositori:

```
package cat.xtec.ioc.domain.repository;
import cat.xtec.ioc.domain.Espectacle;
import java.util.List;
public interface EspectacleRepository {

    List<Espectacle> getAll();
    void add(Espectacle espectacle);

}
```

```
//In memory repository:
package cat.xtec.ioc.domain.repository.impl;
import cat.xtec.ioc.domain.Espectacle;
import cat.xtec.ioc.domain.repository.EspectacleRepository;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
public class InMemoryEspectacleRepository implements EspectacleRepository {

    private final Map<String, Espectacle> shows = new HashMap();

    public InMemoryEspectacleRepository() {
        Espectacle primer = new Espectacle("111","Espectacle_Catalunya","Teatre_Catalunya",300, 200);
        Espectacle segon =  new Espectacle("222","Espectacle_Girona","Teatre_Girona",200, 100);
        Espectacle tercer = new Espectacle("333","Espectacle_Nacional","Teatre_Granollers",100, 80);
        Espectacle quart = new Espectacle("444","Espectacle_Nacional","Teatre_Catalunya",300, 200);
        Espectacle cinque = new Espectacle("555","Espectacle_Nacional","Teatre_Girona",300, 200);

        shows.put("111", primer);
        shows.put("222", segon);
        shows.put("333", tercer);
        shows.put("444", quart);
        shows.put("555", cinque);
    }

    public List<Espectacle> getAll() {
        return new ArrayList<Espectacle>(shows.values());
    }

    public void add(Espectacle espectacle) {
        this.shows.put(espectacle.getIdEspectacle(), espectacle);
    }

}
```

//Application Config:

```
package cat.xtec.ioc.service;
import javax.ws.rs.core.Application;
@javax.ws.rs.ApplicationPath("rest")
public class ApplicationConfig extends Application {
}
```

//Espectacle rest service:

```
package cat.xtec.ioc.service;
import cat.xtec.ioc.domain.Espectacle;
import cat.xtec.ioc.domain.repository.EspectacleRepository;
import cat.xtec.ioc.domain.repository.impl.InMemoryEspectacleRepository;
import java.net.URI;
import java.util.List;
import javax.inject.Singleton;
import javax.ws.rs.BadRequestException;
import javax.ws.rs.Consumes;
import javax.ws.rs.GET;
import javax.ws.rs.NotFoundException;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.core.Context;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.UriInfo;
@Path("/cartellera")
@Singleton
public class EspectacleRestService {
    @Context private UriInfo uriInfo;

    private EspectacleRepository espectacleRepository = new InMemoryEspectacleRepository();

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public List<Espectacle> getAll() {
        return this.espectacleRepository.getAll();
    }
    @POST
    @Consumes(MediaType.APPLICATION_JSON)
    public Response create(Espectacle espectacle) {
        if(espectacle == null) {
            throw new BadRequestException();
        }
        this.espectacleRepository.add(espectacle);

        URI bookUri = uriInfo.getAbsolutePathBuilder().path(espectacle.getIdEspectacle()).build();
        return Response.ok(bookUri).build();
    }
}
```

Comentari:
S'esperava el com es faria, consumint el servei, no com està implementat el servei

