



Desenvolupament d'Aplicacions Web

Mòdul 7 – Desenvolupament web en entorn servidor

EAC5

(Curs 2022–23 / 2n semestre)

Solució

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

Exercici 1:

Heu d'implementar un servei SOAP que ens permeti consultar l'estoc de videojocs de la botiga.

Ha d'implementar els mètodes:

`getAllGames();`

`getGame(String gameId);`

`getNumUnitatsVenudes(String gameId);`

`comprar(String gameId);`

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 `VideojocsRepository`, que és un `InMemoryRepository`


Pugeu dos documents en aquesta pregunta

//interficie

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

@WebService


@SOAPBinding(style = SOAPBinding.Style.DOCUMENT)

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 1 de 7
	Versió: 03		

```
public interface JocServiceEndpoint {  
  
    @WebMethod ArrayList<Joc> getAllGames();  
  
    @WebMethod Joc getGame(String gameId);  
  
    @WebMethod Joc getNumUnitatsVenudes(String gameId);  
  
    @WebMethod Joc comprar(String gameId);  
  
}
```

//classe

```
import cat.xtec.ioc.service.Joc;  
  
import javax.jws.WebService;  
  
import cat.xtec.ioc.service.JocServiceEndpoint;  
  
import java.util.ArrayList;  
  
@WebService(serviceName = "JocService", endpointInterface =  
"cat.xtec.ioc.service.JocServiceEndpoint")  
  
public class JocServiceEndpointImpl implements JocServiceEndpoint {  
  
    private final InMemoryJocRepository jocRepository = new InMemoryJocRepository();  
  
    @Override  
  
    public ArrayList<Joc> getAllGames() {  
  
        return new ArrayList<Joc>(this.jocRepository.getAllGames());  
  
    }  
  
    @Override  
  
    public Joc getGame(String gameId) {  
  
        return this.jocRepository.getGame(gameId);  
  
    }  
  
}
```

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 2 de 7
	Versió: 03		

@Override

```
public Joc getNumUnitatsVenudes(String gameId) {  
    return this.jocRepository.getNumUnitatsVenudes(gameID);  
}
```

@Override


```
public Joc comprar(String gameId) {  
    return this.jocRepository.(getNumUnitatsVenudes(1+gameID));  
}  
}
```

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="VideojocService">  
  <import namespace="http://service.ioc.xtec.cat/"  
    location="http://localhost:8080/resentioc-video/VideojocService?wsdl=1"/>  
  <binding xmlns:ns1="http://service.ioc.xtec.cat/"  
    name="VideoJocServiceEndpointImplPortBinding" type="ns1:VideoJocServiceEndpoint">  
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"  
      style="document"/>  
    <operation name="getNumUnitatsVenudes">  
      <soap:operation soapAction=""/>  
      <input>  
        <soap:body use="literal"/>  
      </input>  
    </operation>  
  </binding>  
</definitions>
```

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 3 de 7
	Versió: 03		

```

        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="getAllGames">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="getGame">
        <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="VideojocService">
    <port name="VideoJocServiceEndpointImplPort"
binding="tns:VideoJocServiceEndpointImplPortBinding">
        <soap:address
location="http://localhost:8080/resentioc-video/VideojocService"/>
    </port>
</service>

```

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 4 de 7
	Versió: 03		

</definitions>

Exercici 3:

Prepareu un servei REST en una classe que ens permeti consultar la llista de videojocs d'una botiga, cercar-los per nom, saber-ne les unitats venudes d'un en concret, fer-ne una compra (modificar el número d'exemplars venuts +1) i afegir-ne un de nou al catàleg.

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.

Podeu escriure aquí el codi o pujar-ne un fitxer


```
@Path("/videojocs")
@Stateless
public class VideojocsRestService {

    private VideojocRepository videojocRepository = new InMemoryVideojocRepository ( );

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

    @GET
    @Path("/{nom}")
    @Produces(MediaType.APPLICATION_JSON)
    public Videojoc find(@PathParam("nom") String nom) {
        return this.videojocRepository.get ( nom ) ;
    }

    @GET
    @Path("getUnitatsVenudes/{id}")
    @Produces(MediaType.APPLICATION_JSON)
    public Videojoc getUnitatsVenudes(@PathParam("nom") String nom) {
```

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 5 de 7
	Versió: 03		

```
        return this.videojocRepository.getUnitatsVenudes ( nom );
    }

    @PUT
    @Path("/{id}")
    @Consumes(MediaType.APPLICATION_JSON)
    public void edit(@PathParam("id") String id) {
        this.videojocRepository.comprar(id) ;
    }

    @POST
    @Consumes(MediaType.APPLICATION_JSON)
    public void create(Videojoc videojoc) {
        this.bookRepository.add(videojoc);
    }
}
```

Exercici 4:


Escriviu un possible client en Java per a un servei REST publicat a la URL <http://localhost:8080/consultaJocs/rest/>

que aconseguixi la llista de tots els jocs del catàleg. (/jocs)

Amb quina API treballeu?

Podeu escriure aquí la resposta o pujar un fitxer.

```
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;
```

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 6 de 7
	Versió: 03		


```
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/consultaJocs/rest/jocs").port(8080).build();
        WebTarget target = CLIENT.target(uri);
        Invocation invocation = target.request(MediaType.APPLICATION_JSON).buildGet();
        Response res = invocation.invoke();
        //List<Videojoc> returnedGamesList = res.readEntity(new GenericType<List<Videojoc>>() {});
        System.out.println("GetStatus of response: " + res.getStatus());
        System.out.println(res.readEntity(String.class));
    }
}
```

La api utilitzada es JAX-RS

	Codi: I71	Exercici d'avaluació continuada 5	Pàgina 7 de 7
	Versió: 03		