REST WEB SERVICES EN JAVA EE:

Start Page | pom.xml [sga-jee-web] | PersonaServiceRS.java | web.xml | Persona.java

Source | General | Servlets | Filters | Pages | References | Security | History

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-
    version="4.0">
     <session-config>
         <session-timeout>
             30
         </session-timeout>
     </session-config>
     <welcome-file-list>
         <welcome-file>faces/index.xhtml</welcome-file>
     </welcome-file-list>
     <servlet>
         <servlet-name>JerseyWebApplication</servlet-name>
         <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
         <init-param>
             <param-name>jersey.config.server.provider.packages</param-name>
             <param-value>sv.com.gm.sga.servicio</param-value>
         </init-param>
         <load-on-startup>1</load-on-startup>
     </servlet>
     <servlet-mapping>
         <servlet-name>JerseyWebApplication</servlet-name>
         <url-pattern>/webservice/*</url-pattern>
     </servlet-mapping>
</web-app>
```

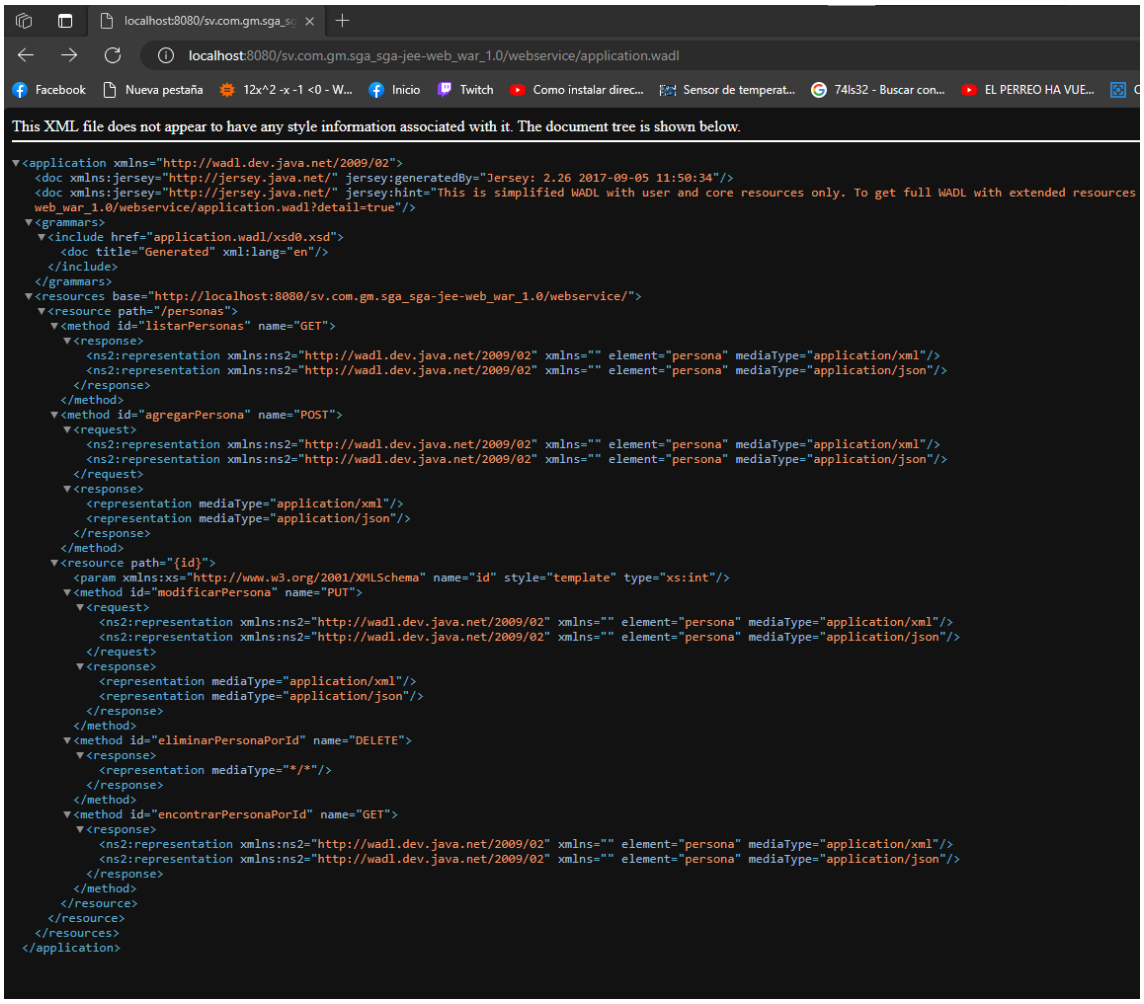
Start Page | pom.xml [sga-jee-web] | PersonaServiceRS.java | web.xml | Persona.java

Source | History

```java
package sv.com.gm.sga.domain;

import java.io.Serializable;
import java.util.List;
import javax.persistence.*;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.*;

@Entity
@NamedQueries({
    @NamedQuery(name = "Persona.findAll", query = "SELECT p FROM Persona p"),
    @NamedQuery(name = "Persona.findByIdPersona", query = "SELECT p FROM Persona p WHERE p.idPersona = :idPersona"),
    @NamedQuery(name = "Persona.findByNombre", query = "SELECT p FROM Persona p WHERE p.nombre = :nombre"),
    @NamedQuery(name = "Persona.findByApellido", query = "SELECT p FROM Persona p WHERE p.apellido = :apellido"),
    @NamedQuery(name = "Persona.findByEmail", query = "SELECT p FROM Persona p WHERE p.email = :email"),
    @NamedQuery(name = "Persona.findByTelefono", query = "SELECT p FROM Persona p WHERE p.telefono = :telefono")})
@XmlAccessorType(XmlAccessType.FIELD)
@XmlRootElement
public class Persona implements Serializable {
```

```java
package sv.com.gm.sga.servicio;
import java.util.List;
import javax.ejb.Stateless;
import javax.inject.Inject;
import javax.ws.rs.*;
import javax.ws.rs.core.*;
import javax.ws.rs.core.Response.Status;
import sv.com.gm.sga.domain.Persona;

@Path("/personas")
@Stateless
public class PersonaServiceRS {

    @Inject
    private PersonaService personaService;

    @GET
    @Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    public List<Persona> listarPersonas() {
        return personaService.listarPersonas();
    }

    @GET
    @Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    @Path("{id}")//hace referencia a /personas/{id}
    public Persona encontrarPersonaPorId(@PathParam("id") int id) {
        return personaService.encontrarPersonaPorId(new Persona(idPersona: id));
    }

    @POST
    @Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    @Consumes({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    public Response agregarPersona(Persona persona) {
        try {
            personaService.registrarPersona(persona);
            return Response.ok().entity(entity: persona).build();
        } catch (Exception e) {
            e.printStackTrace(s: System.out);
            return Response.status(status: Status.INTERNAL_SERVER_ERROR).build();
        }
    }

    @PUT
    @Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
```

```java
    @PUT
    @Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    @Consumes({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    @Path("{id}")
    public Response modificarPersona(@PathParam("id") int id, Persona personaModificada) {
        try {
            Persona persona = personaService.encontrarPersonaPorId(new Persona(idPersona: id));
            if (persona != null) {
                personaService.modificarPersona(persona:personaModificada);
                return Response.ok().entity(entity: personaModificada).build();
            } else {
                return Response.status(status: Status.NOT_FOUND).build();
            }
        } catch (Exception e) {
            e.printStackTrace(s: System.out);
            return Response.status(status: Status.INTERNAL_SERVER_ERROR).build();
        }
    }

    @DELETE
    @Path("{id}")
    public Response eliminarPersonaPorId(@PathParam("id") int id) {
        try {
            personaService.eliminarPersona(new Persona(idPersona: id));
            return Response.ok().build();
        } catch (Exception e) {
            e.printStackTrace(s: System.out);
            return Response.status(status: 404).build();
        }
    }

}
```



CLIENTE

```java
package domain;

import javax.xml.bind.annotation.XmlRootElement;


@XmlRootElement
public class Persona {

    private int idPersona;
    private String nombre;
    private String apellido;
    private String email;
    private String telefono;

    public Persona(){}

    public Persona(int idPersona){
        this.idPersona = idPersona;
    }

    public int getIdPersona() {
        return idPersona;
    }

    public void setIdPersona(int idPersona) {
        this.idPersona = idPersona;
    }

    public String getNombre() {
        return nombre;
    }
```

```java
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getApellido() {
        return apellido;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }

    @Override
    public String toString() {
        return "Persona{" + "idPersona=" + idPersona + ", nombre=" + nombre + ", apellido=" + apellido + ", email=" + email + ", telefono=" + telefono + '}';
    }

}
```

```java
    private static Client cliente;
    private static WebTarget webTarget;
    private static Persona persona;
    private static List<Persona> personas;
    private static Invocation.Builder invocationBuilder;
    private static Response response;

    public static void main(String[] args) {
        cliente = ClientBuilder.newClient();

        //Leer una persona (metodo get)
        webTarget = cliente.target(uri:URL_BASE).path(path:"/personas");
        //Proporcionamos un idPersona valido
        persona = webTarget.path(path:"/1").request(acceptedResponseTypes:MediaType.APPLICATION_XML).get(responseType:Persona.class);
        System.out.println("Persona recuperada:" + persona);

        //Leer todas las personas (metodo get con readEntity de tipo List<>
        personas = webTarget.request(acceptedResponseTypes:MediaType.APPLICATION_XML).get(responseType:Response.class).readEntity(new GenericType<List<Persona>>(){});
        System.out.println(x:"\nPersonas recuperadas");
        //imprimirPersonas(personas);
```

```
Find:  Previous  Next  Select  aA  ".*

test.TestPersonaServiceRS  main

Output

Java DB Database Process  GlassFish Server 5.0  Run (TestPersonaServiceRS)

Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ ClienteSgaRestWs ---
Persona recuperada:Persona{idPersona=1, nombre=Juan, apellido=Higuain, email=jperez@mail.com, telefono=22577777}

Personas recuperadas
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.145 s
Finished at: 2023-11-03T22:30:45-06:00
------------------------------------------------------------------------
```

```java
//Leer todas las personas (metodo get con readEntity de tipo List<>
personas = webTarget.request(acceptedResponseTypes:MediaType.APPLICATION_XML).get(responseType:Response.class).readEntity(new GenericType<List<Persona>>(){});
System.out.println(x:"\nPersonas recuperadas");
imprimirPersonas(personas);
```

```
Persona:Persona{idPersona=1, nombre=Juan, apellido=Higuain, email=jperez@mail.com, telefono=22577777}
Persona:Persona{idPersona=2, nombre=Karla, apellido=Gomez, email=kgomez@mail.com, telefono=25252525}
Persona:Persona{idPersona=3, nombre=Maria, apellido=Gutierrez, email=mgutierrez@mail.com, telefono=88991199}
Persona:Persona{idPersona=4, nombre=Pedro, apellido=Luna, email=pedro@gmail.com, telefono=13132153}
Persona:Persona{idPersona=6, nombre=Hugo, apellido=Hernandez, email=hhernandez@mail.com, telefono=55778822}
Persona:Persona{idPersona=7, nombre=pepe, apellido=problemas, email=pepe@hotmail.com, telefono=79447751}
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.446 s
Finished at: 2023-11-03T22:34:06-06:00
------------------------------------------------------------------------
```

```
invocationBuilder = webTarget.request(acceptedResponseTypes: MediaType.APPLICATION_XML);
response = invocationBuilder.post(entity: Entity.entity(entity: nuevaPersona, mediaType: MediaType.APPLICATION_XML));
System.out.println(x: "");
System.out.println(x: response.getStatus());
//Recuperamos la personas recien agregada para despues modificarla y al final eliminarla
Persona personaRecuperada = response.readEntity(entityType: Persona.class);
System.out.println("Persona agregada:" + personaRecuperada);
```

```
Personas recuperadas

200
Persona agregada:Persona{idPersona=8, nombre=Carlos, apellido=Miranda, email=cmiranda3@mail.com, telefono=99887700}
```

```
200
Persona agregada:Persona{idPersona=9, nombre=Carlos, apellido=Miranda, email=cmiranda3@mail.com, telefono=99887700}

response:200
Persona modifica:Persona{idPersona=9, nombre=Carlos, apellido=Ramirez, email=cmiranda3@mail.com, telefono=99887700}
```

```
//Modificar la persona (metodo put)
//persona recuperada anteriormente
Persona personaModificar = personaRecuperada;
personaModificar.setApellido(apellido: "Ramirez");
String pathId = "/" + personaModificar.getIdPersona();
invocationBuilder = webTarget.path(path: pathId).request(acceptedResponseTypes: MediaType.APPLICATION_XML);
response = invocationBuilder.put(entity: Entity.entity(entity: personaModificar, mediaType: MediaType.APPLICATION_XML));

System.out.println(x: "");
System.out.println("response:" + response.getStatus());
System.out.println("Persona modifica:" + response.readEntity(entityType: Persona.class));
```

```
//eliminar una persona
//persona recuperada anteriormente
Persona personaEliminar = personaRecuperada;
String pathEliminarId = "/" + personaEliminar.getIdPersona();
invocationBuilder = webTarget.path(path: pathEliminarId).request(acceptedResponseTypes: MediaType.APPLICATION_XML);
response = invocationBuilder.delete();
System.out.println(x: "");
System.out.println("response:" + response.getStatus());
System.out.println("Persona Eliminada" + personaEliminar);
```

```
response:200
Persona EliminadaPersona{idPersona=10, nombre=Carlos, apellido=Ramirez, email=cmiranda3@mail.com, telefono=99887700}
------------------------------------------------------------------------
```