

Proyecto SW2

Gestión de una agenda con XML, ahora con servicios RESTful

Enunciado

En este proyecto se busca poner en práctica lo aprendido en clase de REST. Para ello se va a diseñar un pequeño programa en Java que gestione una agenda utilizando un servicio web REST. Recordad que el enunciado mencionará Agenda y Persona, pero cada uno debe realizarlo sobre las clases que tiene asignadas.

En este caso además se pide que haya autenticación de usuarios y que cada uno tenga su propia Agenda almacenada en una base de datos.

Además, se pedirá que se sigan lo más fielmente posible los principios de la filosofía REST.

Estructura XML

Debéis definir una estructura XML de la Agenda (según corresponda) con XSD. Ejemplo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="emailAddress">
    <xs:restriction base="xs:token">
      <xs:maxLength value="254"/>
      <xs:pattern value="[_\ -a-zA-Z0-9\.\.+] +@[a-zA-Z0-9] (\.\.?[_\ -a-zA-Z0-9]*[a-zA-Z0-9]) *"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="Persona">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="email" type="emailAddress"/>
      <xs:element name="telephone" type="xs:string"/>
      <xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="Agenda">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Persona" type="Persona" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Persona" type="Persona"/>
</xs:schema>
```

Servicios Básicos

1. Permitir crear una nueva Agenda (con autenticación)
2. Permitir leer dicha Agenda (con autenticación)
3. Permitir Crear Personas en dicha Agenda (C) (con autenticación)
4. Permitir Leer las Personas en dicha Agenda (R) (con autenticación)
5. Permitir Actualizar Personas en dicha Agenda (U) (con autenticación)
6. Permitir Borrar Personas en dicha Agenda (D) (con autenticación)
7. Permitir exportar la Agenda del servicio web o una Persona (con autenticación)
8. Permitir importar la Agenda al servicio web o una Persona (con autenticación)
9. (Opcional) Desarrollar algún servicio más que se considere oportuno/interesante

Servicios sin autenticación

1. Ofrecer un servicio que al enviarle una Agenda y/o Persona la devuelva en HTML, convertido con XSLT (sin autenticación)
2. (Opcional) Ofrecer un servicio que permita validar una Persona o Agenda con la XSD (sin autenticación)
3. (Opcional) Desarrollar algún servicio más sin autenticación que se considere oportuno/interesante

Servicios gestión autenticación

1. Ofrecer los servicios necesarios para gestionar la autenticación del usuario.

Servicios avanzados (opcionales)

1. Permitir obtener una Persona en concreto de un usuario a partir del nombre.
2. Permitir que un mismo usuario tenga varias Agendas, por lo tanto, desarrollar las operaciones CRUD sobre Agenda también.
3. Permitir que existieran Agendas compartidas entre varios usuarios
4. Cualquier otro servicio que se considere necesario/útil.

Se debe intentar seguir la filosofía REST al completo al realizar estos servicios, incluyendo HATEOAS.

Cliente Java

1. Desarrollar un cliente Java que interactúe con los servicios creados.

Cliente Web

1. Desarrollar una aplicación web que consuma los servicios creados.

Interfaz

La interfaz de los clientes debe permitir acceder a toda la funcionalidad antes descrita:

1. Opción para crear una nueva agenda
2. Opción para obtener la Agenda contenida en el servicio web y mostrarla
3. Opción para obtener una Persona del servicio y mostrarla por pantalla
4. Opción para crear una Persona y añadirla al servicio web de la agenda
5. Opción para borrar una Persona del servicio web
6. Opción para actualizar una Persona del servicio web
7. Opciones para exportar/importar Agenda y Persona
8. Opción para enviar una Agenda para su validación
9. Opción para enviar una Persona para su validación
10. Autenticarse
11. Etc...

Estructura y diseño

1. Seguid siempre que sea posible las convenciones de nomenclatura de Java y sed coherentes en todo el proyecto
2. Separad claramente las distintas responsabilidades dentro de vuestro proyecto. Usad clases y paquetes para estructurar el código de forma adecuada, usando herencia y interfaces cuando sea conveniente.
3. La interacción entre los distintos componentes del sistema debe realizarse **mediante objetos** (como Persona o Agenda).
4. No deben aparecer rutas absolutas para los ficheros.
5. Debéis entregar tres proyectos distintos, uno que genera los servicios y los dos clientes.

Final

1. Revisar que se responde a lo que se pide.
2. Revisar que el código que enviáis funciona y que no hay errores obvios.
3. El código debe funcionar “out-of-the box”
4. Revisar que el proyecto que vais a enviar funciona sin la necesidad de ningún archivo adicional o que está incluido en el proyecto. Añadid la documentación necesaria para hacerlo funcionar. El proyecto debe ser autocontenido y será evaluado por otro grupo.

Roadmap recomendado

1. Completar todo lo pedido en el enunciado RESTProject. Básicamente las operaciones CRUD sobre Agenda, en la primera versión puedes utilizar la anotación `@Singleton` para probar y un plugin para hacer peticiones HTTP.
2. Integrar la base de datos. Debe ser con un pool, exactamente igual que para Sistemas Web I. Quitar la anotación `@Singleton` la Agenda debe almacenarse en la base de datos.
3. Adaptar la práctica para que cada usuario tenga su propia Agenda y solo acceda a ella. Este paso se puede hacer a la vez que el paso anterior.
4. Integrar completamente la autenticación de usuarios y el resto de servicios necesarios para ello. Esto requiere un estudio de las opciones posibles.
5. Comprobar que los servicios desarrollados cumplen con la filosofía REST, intentar mejorarlos e incluir HATEOAS.
6. Desarrollar una primera versión del cliente Java.
7. Completar el resto de servicios necesarios.
8. Desarrollar una primera versión del cliente Java Web.
9. (Opcional) Desarrollar alguno de los apartados opcionales
10. Revisar una vez más los servicios, que la gestión de usuarios está correctamente realizada, que un usuario no puede acceder a la Agenda de otro, etc.
11. Revisar los clientes y completarlos si fuera necesario.
12. Completar la documentación y revisar una vez más que todo funciona
13. Entregar la práctica de la forma indicada.
14. Realizar la defensa en la forma y fecha que se acuerde.
15. REST and enjoy.