

Aplicacions Distribuïdes (AD)

Práctica 4: Desarrollo de servicios web con REST

Creación de servicios web basados en REST

En esta práctica vamos a crear servicios Web RESTful con *Netbeans*.

Lo primero es crear una aplicación web que contenga el servicio REST. Tiene que ser una nueva aplicación web. Podéis llamar a dicha aplicación *RestAD*.

Una vez creada, hay que crear el servicio REST. Utilizaremos la opción *RESTful Web Services from Patterns*. Crearemos un servicio de tipo recurso genérico, que responderá a métodos HTTP siguiendo las directrices de REST. El tipo MIME soportado será *text/html*. Otros tipos MIME posibles son *application/html*, *application/json* y *text/plain*.

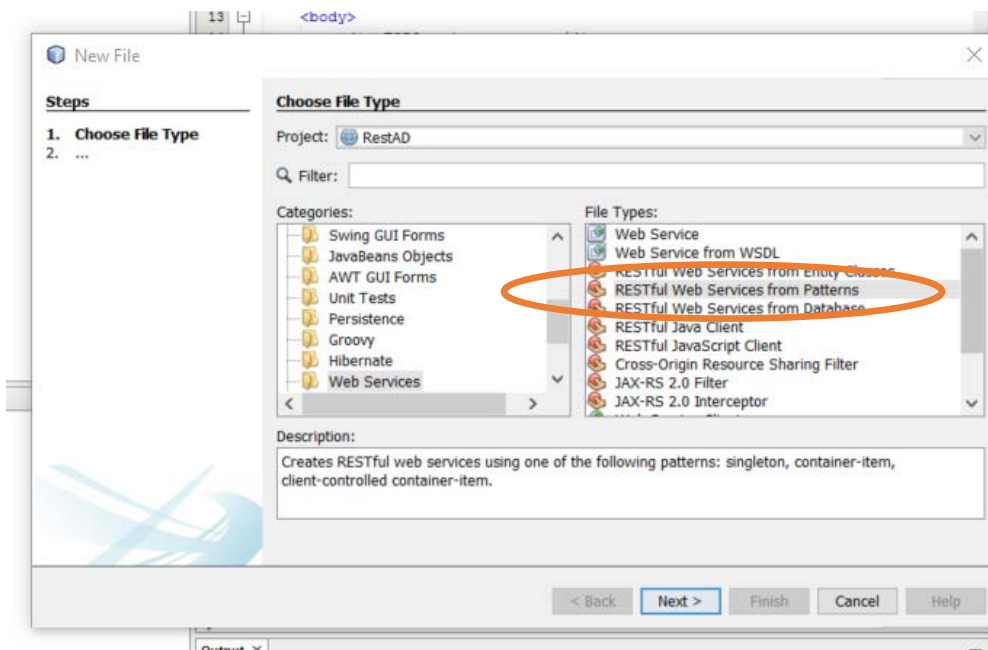


Figura 1. Creación servicio RESTful

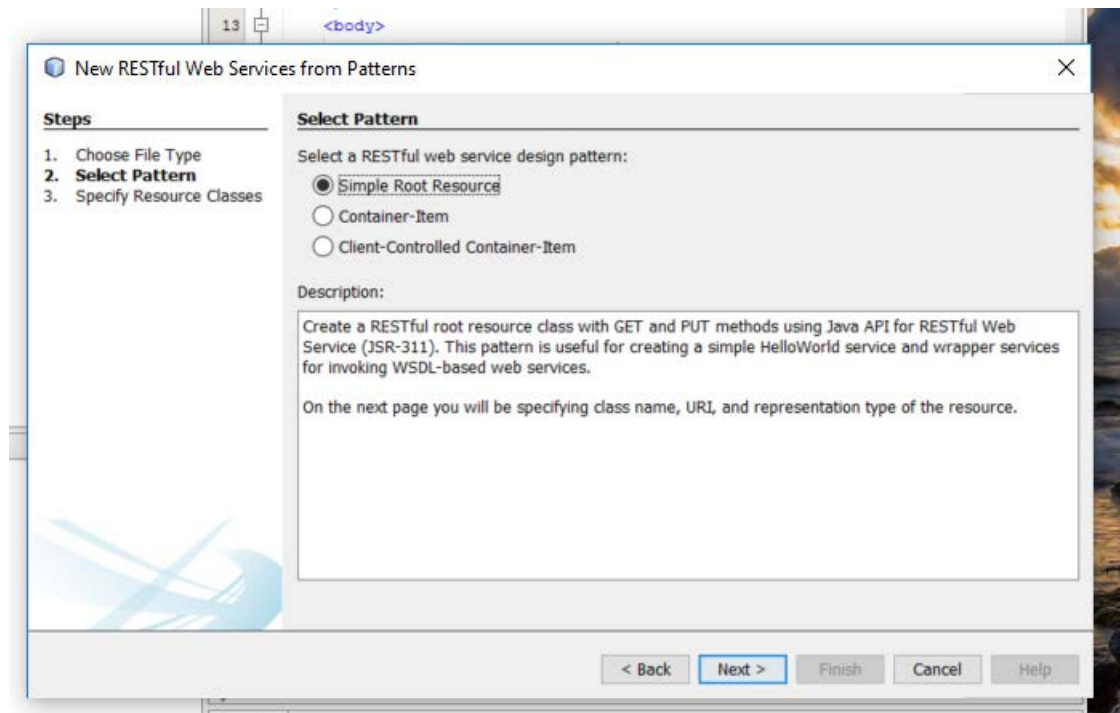


Figura 2. Selección de opción Simple Root Resource.

Definid Resource Package como `restad` y seleccionad `text/html` como tipo MIME en la pantalla que se muestra en la Figura 3.

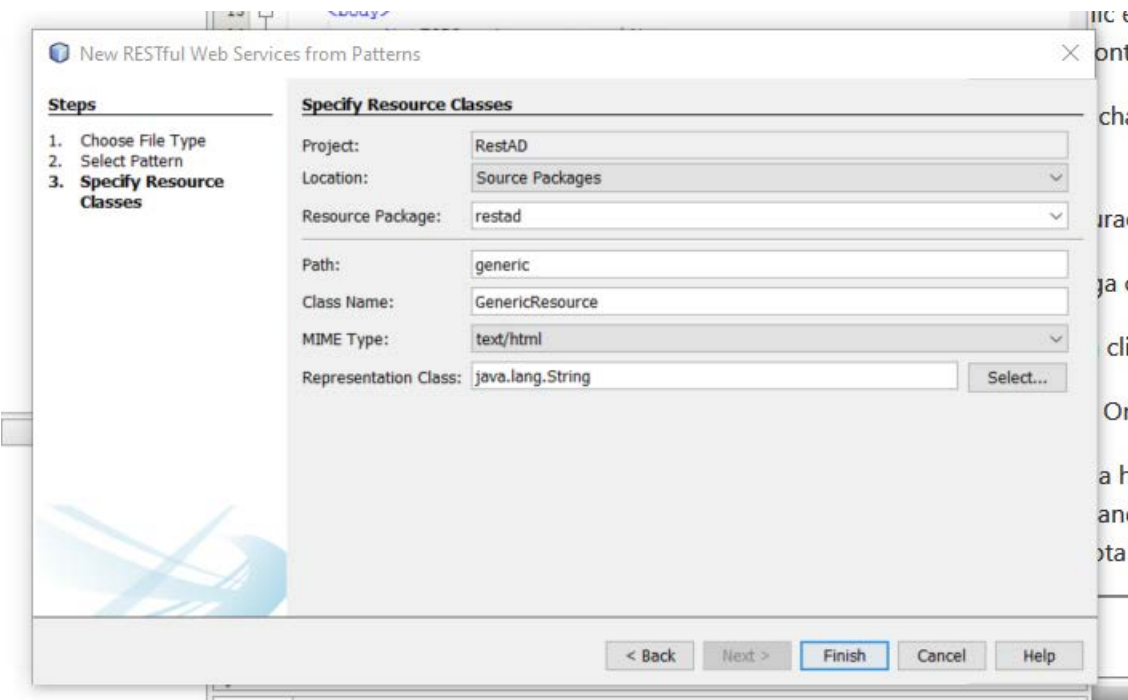


Figure 3. Creación del servicio como GenericResource.

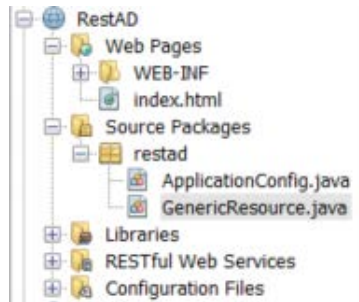
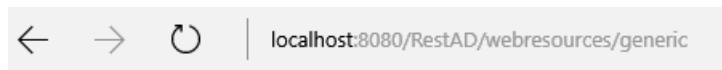


Figura 4. Ficheros generados.

Revisa el código Java de los ficheros que se han generado. Busca los métodos HTTP implementados en el fichero `GenericResource.java`.

Modifica el método `getHtml()`. Añade el código `return "<html><head/><body><h1>Hello World!</h1></body></html>";` y ejecuta la aplicación web para que se ejecute el servicio. Para acceder al método, utiliza la siguiente URL en un navegador:

`http://localhost:<puerto>/RestAD/webresources/generic/`



Hello world!

Figure 5. Ejecutando el servicio REST.

Responde a las siguientes cuestiones:

- ¿Por qué podemos acceder al servicio desde el navegador?
- ¿Qué método HTTP se está utilizando?
- ¿Con qué tipo MIME?
- ¿Dónde y cómo se indica la URL, el método HTTP y el tipo MIME del servicio REST?

Implementando el servicio

Se tiene que implementar un servicio web que responda a los métodos GET y POST. La funcionalidad cliente se puede implementar también en la misma aplicación.

El trabajo a realizar en esta práctica son las mismas operaciones que en el servicio web SOAP de la práctica 3. Las cabeceras de las operaciones REST deben ser como se muestra a continuación:

```
/**
 * POST method to register a new image
 * @param title
 * @param description
 * @param keywords
 * @param author
 * @param crea_date
 * @return
 */
@Path("register")
@POST
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
@Produces(MediaType.TEXT_HTML)
public String registerImage (@FormParam("title") String title,
                             @FormParam("description") String description,
                             @FormParam("keywords") String keywords,
                             @FormParam("author") String author,
                             @FormParam("creation") String crea_date)

/**
 * POST method to register a new image
 * @param title
 * @param description
 * @param keywords
 * @param author
 * @param crea_date
 * @return
 */
@Path("modify")
@POST
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
@Produces(MediaType.TEXT_HTML)
public String modifyImage (@FormParam("title") String title,
                           @FormParam("description") String description,
                           @FormParam("keywords") String keywords,
                           @FormParam("author") String author,
                           @FormParam("creation") String crea_date)

/**
 * GET method to list images
 * @return
 */
@Path("list")
@GET
@Produces(MediaType.TEXT_HTML)
public String listImages ()
```

```

/**
 * GET method to search images by id
 * @param id
 * @return
 */
@Path("searchID/{id}")
@GET
@Produces(MediaType.TEXT_HTML)
public String searchByID (@PathParam("id") int id)

/**
 * GET method to search images by title
 * @param title
 * @return
 */
@Path("searchTitle/{title}")
@GET
@Produces(MediaType.TEXT_HTML)
public String searchByTitle (@PathParam("title") String title)

/**
 * GET method to search images by creation date
 * @param creaDate
 * @return
 */
@Path("searchCreationDate/{date}")
@GET
@Produces(MediaType.TEXT_HTML)
public String searchByCreationDate (@PathParam("date") String date)

/**
 * GET method to search images by author
 * @param author
 * @return
 */
@Path("searchAuthor/{author}")
@GET
@Produces(MediaType.TEXT_HTML)
public String searchByAuthor (@PathParam("author") String author)

/**
 * GET method to search images by keyword
 * @param keywords
 * @return
 */
@Path("searchKeywords/{keywords}")
@GET
@Produces(MediaType.TEXT_HTML)
public String searchByKeywords (@PathParam("keywords") String
                                keywords)

```

Para probar el servicio, crea tantos formularios HTML como necesites en la propia aplicación que implementa el servicio web REST. Ten en cuenta que algunas de las operaciones no necesitan un formulario para poderlas llamar.

Trabajo adicional

Para mejorar la nota de esta práctica, se puede implementar una operación de búsqueda combinada, por ejemplo, por título y autor o por título y descripción.

Entrega

Se debe entregar un documento que responda a las preguntas iniciales, el código de la práctica y un informe donde se describa brevemente el trabajo realizado y las ampliaciones realizadas.

Consultad la fecha de entrega en el Racó.