

Servicios de acceso a servicios REST

Índice

1 Consulta del libros.....	2
2 Timeline público de Twitter.....	3

Ayuda

Entre las plantillas de la sesión se incluye la aplicación `RESTfulSwingClient`. Se trata de una aplicación Java de escritorio que nos permite probar servicios web RESTful de forma sencilla. Simplemente tendremos que importarla en Eclipse y ejecutar su clase `restfulswingclient.RESTfulSwingClientApp`. Se abrirá una ventana en la que podremos introducir la URL del servicio a probar y realizar una petición, proporcionando el contenido y cabeceras que especifiquemos. En la misma ventana podremos visualizar la respuesta recibida. Esta aplicación puede resultar de utilidad para entender mejor el funcionamiento de los servicios que utilizaremos en esta sesión.

1. Consulta del libros

En primer lugar vamos a implementar una aplicación que consulte el listado de libros disponible en una biblioteca online. Partiremos de la plantilla `ListadoLibros`, donde encontramos una tabla o lista que deberemos rellenar con los datos obtenidos del servicio REST de la biblioteca. Se pide:

a) Realizar desde la aplicación una llamada al servicio, solicitando que nos devuelva la representación XML del recurso lista de libros. La URL a la que debemos acceder para obtener dicho listado es:

```
http://server.jtech.ua.es:8080/jbib-rest/resources/libros
```

Recoge el resultado como una cadena, y provisionalmente muéstralo como *log*. Comprueba que hemos obtenido los datos de forma correcta en formato XML (es importante enviar en la petición la cabecera `Accept` para asegurarnos de que se devuelva este formato).

Ayuda iOS

En el controlador `UAListadoLibrosViewController`, en `viewWillAppear` configura la petición para que pida representación `application/xml`.

Ayuda Android

En la actividad `ListadoLibrosActivity`, y dentro de ella en la tarea asíncrona `CargarTweetsTask`, configura la petición para obtener representación `application/xml`.

b) Implementa el *parsing* del XML anterior, y obtén una lista de libros (objetos `Libro` o `UALibro`). Tras esto se deberá actualizar la interfaz para mostrar en ella la lista de libros obtenida.

Ayuda iOS

Implementa el método del delegado del parser `parser: didStartElement:`

namespaceURI: qualifiedName: attributes: para que lea los datos de los libros del XML y los añada a la lista de libros del controlador encapsulados en objetos UALibro. En actualizarLibrosConDatos: deberemos ejecutar el parser utilizando self como delegado.

Ayuda Android

En parseLibros utiliza XmlPullParser para obtener la lista de libros a partir del XML.

2. Timeline público de Twitter

Vamos a continuar trabajando con el cliente de Twitter que comenzamos en la sesión anterior. Hasta ahora esta aplicación está obteniendo la lista de *tweets* de memoria. Vamos ahora a modificar este comportamiento para que obtenga los *tweets* de los servicios REST publicados por Twitter. Concretamente, accederemos al servicio de *timeline* público y utilizaremos la representación JSON:

http://twitter.com/statuses/public_timeline.json

Ayuda iOS

En el controlador UAListadoTweetsViewController, en viewDidLoad elimina la llamada a inicializaTweets, que es quien carga una lista predefinida de *tweets* en memoria, y en su lugar realiza una conexión asíncrona al servicio mostrando el indicador de actividad de red cuando sea oportuno. En el método actualizarTweetsConDatos, lee el JSON obtenido en forma de NSArray. De cada elemento del *array* (se trata de diccionarios), obtén las propiedades text, user.screen_name y user.profile_image_url (realmente user es una propiedad que contiene otras propiedades, es decir, es un diccionario) y encapsúlalas en objetos UATweet.

Ayuda Android

En la actividad ClienteTwitterActivity, y dentro de ella en la tarea CargarTweetsTask, cambia la inicialización de la lista de *tweets* predefinida por una llamada al servicio de Twitter anterior, y obtén la representación JSON de la respuesta.

