

Informe del Proyecto C de NST : Consulta de Foaf

NST-C01
Manuel Costas
Daniel Couto
Andrés Muñoz

Introducción

En este documento se presenta el informe del proyecto C de Nuevos Servicios Telemáticos basado en una Consulta de Foaf mediante una API REST. Se recogen tanto las tecnologías empleadas como las funciones a utilizar, los recursos expuestos y una explicación pormenorizada del proyecto.

Descripción de las tecnologías empleadas

En lo referente a las tecnologías empleadas, nos encontramos con varias a listar:

- Java : Java es un lenguaje de programación orientado a objetos con el que se ha realizado tanto la página web del usuario a acceder al servicio mediante un servlet como las distintas funciones de la API sobre la que se consulta.
- HTTP : Protocolo de transferencia utilizado para las transacciones en la World Wide Web, como protocolo orientado a transacciones siguiendo un esquema petición-respuesta.
- SPARQL : Lenguaje estandarizado para la consulta de grafos RDF, siendo una tecnología clave para el desarrollo de la Web Semántica.

Descripción funcional del proyecto

A continuación se explican las funcionalidades del proyecto en cuestión:

1. Al estar todo concentrado en un único archivo, se requiere software para el despliegue del servicio como Apache Tomcat(actualmente en su versión 9.0) para la API REST como tal y para la referencia de los archivos foaf de lectura, a los que se direccionará como localhost:xxxx/DATA/foafX.rdf .
2. El usuario accederá al servicio de consulta mediante la URL localhost:XXXX/RestDemo/NSTC. Dicha URL le desplegará una página muy sencilla con tres consultas a tratar. La API reside bajo la URL localhost:XXXX/RestDemo/Consultas, siendo referenciado por cada consulta cuando se solicita la petición correspondiente.
3. La primera consulta solicita un nombre como parámetro de entrada. A la hora de responder, se escribirá en un textfield de tamaño predeterminado el archivo o archivos FOAF que se correspondan con la persona de la que se introdujo el nombre, pudiendo ser uno o varios archivos FOAF de los que se utilizan para esta consulta.
4. Con respecto a la segunda consulta, se piden tres parámetros. Por este orden, son el nombre de una persona, el campo de búsqueda y el nivel de búsqueda. La respuesta a esta consulta será una lista de nombres de aquellas personas que sean amigos de la persona introducida, bajo ese nivel de búsqueda y con el mismo parámetro que tengan en el campo de búsqueda. En el caso de no introducir un campo de búsqueda, se omitirá

dicha restricción devolviendo únicamente los amigos en ese nivel de búsqueda. Del mismo modo, se entiende un nivel de búsqueda 1 como el nivel de búsqueda directa (es decir, las personas que reconoce en su archivo FOAF), nivel de búsqueda 2 como amigos de aquellos amigos de conoce, estos últimos incluidos y así sucesivamente.

5. Y como tercera consulta tenemos el nivel de cercanía de 2 personas cualesquiera, siendo una variante del número de Bacon (en honor al actor Kevin Bacon). Se trata de ver la proximidad entre dos personas en el grafo establecido con los archivos Foaf expuestos.

Adicionalmente, se tratará aquí una breve descripción de los campos de información que contienen los archivos de este proyecto:

- <foaf:name> : Nombre de la persona, formado por el primer nombre y primer apellido.
- <foaf:title> : Diferenciación de título, siendo Sr o Sra.
- <foaf:givenname> : Nombre de dicha persona.
- <foaf:family_name> : Apellidos de dicha persona.
- <foaf:nick> : Apodo o alias que pueda tener.
- <foaf:mbox_sha1sum> : Correo electrónico de la persona codificado.
- <foaf:homepage> : Página web personal (incluyendo redes sociales)
- <foaf:depiction> : En el caso de que tenga una imagen, por defecto None.
- <foaf:phone> : Teléfono de contacto de dicha persona
- <foaf:workplaceHomepage> : Página web de su lugar de trabajo.
- <foaf:workInfoHomepage> : Página web que informa de su trabajo.
- <foaf:schoolHomepage> : Enlace a su centro de estudios
- <foaf:knows> : Etiqueta en la que hay un amigo, caracterizado por los campos name, mbox_sha1sum y seeAlso.
- <foaf:seeAlso> : Referencia al archivo FOAF de la persona amiga.

Documentación de la API REST

A la hora de profundizar sobre el funcionamiento de la API REST se introducirán los parámetros de consulta y tras esto las llamadas instauradas para realizar las consultas.

Parámetros de consulta:

Clave	Tipo de parámetro	Descripción
name	String	Nombre de la persona en las consultas 1 y 2
field	String	Parámetro de coincidencia en la consulta 2
level	int	Profundidad de la búsqueda de la consulta 2
name1, name2	String, String	Nombres de las personas en la consulta 3

A continuación se plasmarán las distintas llamadas admitidas por la API REST.

1. Observar los datos del archivo de una persona en concreto

GET localhost:8080/RestDemo/Consultas/{name}

Parámetros de entrada: name

Respuesta

- Código 200: Se devuelve un documento en formato XML en la que se ve los datos expuestos anteriormente en el textfield. De no existir dicha persona, no se verá la etiqueta <identifier> que marca la existencia de una persona localizada en ese banco de archivos FOAF.

Ejemplo de respuesta

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?><identifiers><identifier><family>Sabucedo</family><givenname>
Luis</givenname><mail>5527897769681abb37eaba00d81adbf9a061b020</mail><
name>Luis
Sabucedo</name><nick>Modesto</nick><page>http://gist.det.uvigo.es/~lsabucedo/
</page><phone>tel:619678212</phone><photo>photo:None</photo><schoolhp>ht
tp://teleco.uvigo.es</schoolhp><title>Sr</title><workinfo>http://linkedin/in/luis-alva
rez-sabucedo</workinfo><workplace>http://teleco.uvigo.es</workplace></identifier
></identifiers>
```

- Código 400: Se devuelven las etiquetas sin el contenido entre la etiqueta <identifier> en el textfield de la página de acceso.

Ejemplo de respuesta

```
<?xml version="1.0" encoding="UTF-8"standalone="yes"?><identifiers></identifiers>
```

2. Localizar los amigos de una persona según un parámetro y dado un nivel de búsqueda

GET localhost:8080/RestDemo/Consultas/{name}/{field}/{level}

Parámetros de entrada: name, field, level.

Respuestas:

- Código 200: Se devolverá una secuencia de nombres en una cadena separados por una almohadilla (#)

Ejemplo de respuesta

```
#Luis Sabucedo#Daniel Couto#Andres Muñoz#Juan Santos#Manuel
Costas#Iago Tudela#Andrea Carballo#Marta Rodriguez
```

- Código 400: Se devolverá un mensaje de búsqueda no hallada diferenciando las palabras, separadas por la almohadilla(#)

Ejemplo de respuesta

```
#{name}#Consulta#No#Encontrada
```

3. Trazar la distancia entre dos personas dados sus nombres

GET localhost:8080/RestDemo/Consultas/{name1}/{name2}

Parámetros de entrada: name1, name2.

Respuesta:

- Código 200: Se devolverá un entero que muestra la proximidad entre dichas dos personas

Ejemplo de respuesta

2

- Código 400: Se devolverá un valor de 0 en el caso de que hubiese un error en la consulta.

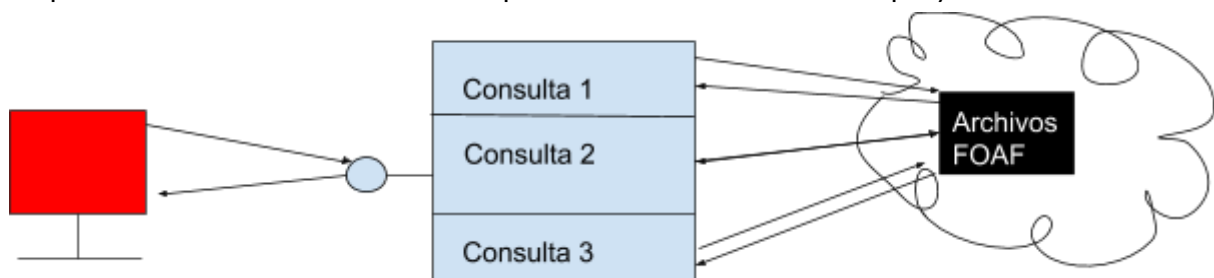
Ejemplo de respuesta

0

Cabe destacar que a la hora de hacer las consultas desde el cliente, no hay problemas de conectividad. Si bien es cierto que realizando las consultas directamente con URL en los código 400 hay excepciones no tratadas, error subsanado a la hora de acceder desde el servlet.

Descripción de los módulos/arquitectura del proyecto

En lo referente a la arquitectura del proyecto, conviene plasmarlo visualmente para después entrar en la composición interna del proyecto como tal.



El cliente desde la página web accede a la API con una Consulta en concreto, en ésta se hace una interacción con los Archivos FOAF, realizando la consulta y devolviéndole al usuario la resolución de la misma.

En cuestión de la composición interna del proyecto, se hará una breve explicación de los distintos archivos que forman parte de la API REST para describir el funcionamiento interno de la misma:

- **ApiRestMain** : En este archivo se recoge la principal funcionalidad del proyecto. Recoge el contexto de las consultas así como lo que realizan cada una de ellas. Las funciones ConsultaPrimaria, ConsultaSecundaria y ConsultaTerciaria son las llamadas de la API REST descritas en el punto anterior. Las funciones Consulta1, Consulta2 y Consulta3 describen el procedimiento de cada consulta para llevar a cabo el resultado. Las funciones findShortestPath, getMainName, getFoaf y query se refieren a funciones internas utilizadas en Consulta2 y Consulta3 para simplificar la visión del código a la hora de trabajar con él. Por último, la función readFoaf es la responsable de leer los foaf y migrarlos como archivos File para su procesamiento interno.

- FoafManager : En esta clase se procesan varias funciones utilizadas en la Consulta2. La función getFoafName se encarga de tomar el nombre de un archivo dado, así como getFoafFields para los campos de datos y getFoafFriends para los amigos del archivo de una persona dada.
- GraphSearcher : En este archivo aparecen las funciones utilizadas para la consulta 2, tanto en el caso de que se introduzca un campo como si no, diferenciando dichas dos acciones así como funciones de apoyo para la localización del nivel de búsqueda.
- PeopleManager : Recoge una función de lectura de los archivos foaf una vez han sido procesados por readFoaf.
- Identifier y Person : Estas dos clases Java describen cómo funcionan las Consultas 1 y 2. Identifier define los campos a devolver en el archivo de respuesta de la Consulta 1 mientras que Person procesa en base a los parámetros de entrada de la Consulta2 un objeto de resolución que después se procesará para enviar con el formato explicado anteriormente
- NSTC : Este archivo contiene el servlet a ejecutar para acceder como usuario al servicio ofrecido por la API REST. Su ejecución muestra la página web en la que se realizan las distintas consultas.

Principales problemas encontrados

Durante el trabajo de implementar este proyecto, han existido distintos problemas como:

- Cuestiones del lenguaje de programación: Una API REST suele programarse con JavaScript y mediante formato JSON. La mayoría de ejemplos o guías encontrados partían de esas dos premisas, por lo que la adaptación a lenguaje Java resultaba un añadido. No obstante, la familiaridad con el lenguaje hizo que este problema se subsanara.
- Desconocimiento general de la metodología: Pese a impartir este curso académico una introducción a API REST, a la hora de trabajar con ella resultaba novedoso por términos de que resulta extraño no haber practicado o cursado anteriormente este tipo de tecnologías de servicios Web. Bien es cierto que una vez visto en profundidad la implementación de este tipo de servicios, fue una situación más llevadera.

Aspectos más significativos o que han resultado más complejos para el desarrollo del proyecto

Principalmente, la mayor complejidad a la hora de desarrollar el proyecto se ha basado en unificar todas las partes de cada uno de los integrantes. Esto se basa en que, pese a que los archivos se obtienen de la misma manera, en cada consulta se procesan de una manera distinta. Si se hubiera hablado con anterioridad sobre estos puntos, gran parte del código pudo haber sido reutilizado, reduciendo la cantidad de líneas de código.

En otro nivel, otra complejidad se basaba en los cambios a realizar en el momento de devolver los parámetros tras las clases de grupo C con el profesor.

Resultados obtenidos y partes obviadas en el desarrollo final

En el desarrollo final, teníamos una API REST que ejecutaba de manera exitosa las consultas propuestas. En la parte final se concentraba el adaptar el servlet a las propuestas de implementación diseñadas y en distribuir los archivos FOAF para tener el diagrama expuesto anteriormente.

A la hora de tener los archivos distribuidos, se ha optado por crear un directorio en Apache Tomcat en el cual los archivos se procesan mediante URL, ya expuesto anteriormente. Esto conviene recordarlo para modificar el archivo server.xml de Tomcat para adaptar dicha carpeta y hacerla accesible para la API REST.

Breve exposición de las pruebas del proyecto

Para finalizar el informe, se hará una breve explicación de las pruebas del proyecto a mostrar en la presentación:

- Consulta 1 : Se hará la solicitud de varios usuarios, viendo cómo aparece su archivo FOAF en el textfield de la página de consulta. Adicionalmente, se verá qué pasa cuando no se introduce un valor y cuando la persona localizada tiene más de un archivo FOAF, lo que devolverá ambos FOAF separados por las etiquetas <identifier> .
- Consulta 2 : Se realizarán búsquedas con y sin el parámetro de campo de búsqueda y con varios niveles de búsqueda para ver las evoluciones y los cambios entre consultas.
- Consulta 3 : Se realizará la búsqueda del índice de proximidad de varias personas. Posteriormente, se modificará un archivo FOAF para ver cómo se ven alteradas esas mismas consultas.