

Servicios de Internet (2023/24)

Practica 2

Convocatoria de Enero

1. Objetivo

El objetivo de esta práctica es profundizar en algunas de las tecnologías presentadas en las clases de teoría, y adquirir la habilidad para trabajar con algunas aplicaciones y las APIs que implementan. Para ello, en la práctica se desarrollará un servicio de librería para consulta de libros.

2. Formato de la información

Para almacenar la información se ha creado el lenguaje *LML*, una aplicación XML cuyas reglas se describen textualmente en el Apéndice A.

Los documentos estarán en URLs de servidores web accesibles sin comprobaciones de seguridad. La URL de un fichero de datos se publicará próximamente en MOOVI.

3. Tareas de la práctica

La práctica consiste en implementar un servicio de consultas sobre la información de los ficheros *LML*.

A través de interfaz HTML, se generarán pantallas navegables desde el browser. Esta incluirá los elementos necesarios para que el usuario prosiga la consulta enviando, en cada fase, un parámetro de las siguientes que guiará la evolución de la consulta. En el documento “*Screens.pdf*” se presentan ejemplos de las pantallas esperadas en el servicio. En estas pantallas se implementará la siguiente sucesión de consultas a través de fases (indicada mediante el parámetro *fase*):

- ☐ **0:** Pide la pantalla inicial en la que se presentará un mensaje de bienvenida. Será el valor por defecto si no hay parámetro *fase*. Se mostrará el nombre del fichero procesado, así como la IP del cliente y del servidor y el navegador desde el que se hace la petición. Estos datos serán generados en el servidor, es decir, no pueden ser averiguados por el cliente usando JavaScript.
- ☐ **1:** Se muestra la lista de países conocidos en los cuales se ha publicado algún libro.
- ☐ **2:** Se muestra la lista de autores con libros publicados en un país (recibido en el parámetro *pais*).
- ☐ **3:** Se muestra la lista de libros de un cierto autor (se recibe su id en el parámetro *autor*). Para cada libro se mostrará su título y código ISBN. Se mostrarán en color rojo aquellos libros que no estén disponibles según el atributo correspondiente.

Cada página debe informar de las selecciones previas que han conducido a ella, y debe incluir un botón para retroceder en la consulta. Cada página irá firmada al pie con el nombre del autor.

Esta consulta también deberá implementarse a través de una interfaz REST. La respuesta será en formato JSON y será acorde a la API descrita en el documento correspondiente que se publicará en las próximas semanas en MOOVI.

4. Entorno de desarrollo y ejecución del servicio

El servicio de esta práctica será proporcionado por un servidor Apache TOMCAT (versión 10), que ejecutará **un único** *servlet* desarrollado por el alumno para cada una de las interfaces propuestas. Cada alumno deberá configurar y ejecutar un TOMCAT propio en su cuenta. Ese TOMCAT será el empleado en el examen práctico.

En la configuración del TOMCAT se definirá un contexto llamado *sintX* (donde *X* es el número de la cuenta del alumno), que será el ámbito en el que se desarrollarán todos los servicios del alumno. Ese contexto estará enlazado con la carpeta “*public_html/webapps*” de la cuenta del alumno.

El acceso inicial al servicio basado en HTML se realizará accediendo a la siguiente URL (suponiendo que el *Firefox* se está ejecutando en la misma máquina que el TOMCAT), y donde *port* es el resultado de sumar 7000 y *X*:

http://localhost:port/sintX/P2Lib

Ese acceso al servicio provocará la ejecución directa de un *servlet* llamado **SintXP2** (su fichero fuente será *SintXP2.java*), que será el encargado de crear y enviar las respuestas (*X* es el número de la cuenta).

Cada selección del usuario provocará **siempre** el envío de datos al *servlet*, que identificará la consulta, buscará el resultado, construirá la respuesta, y la devolverá al solicitante.

El acceso a la API que da soporte a los métodos REST se realizará accediendo a la URL que estará colgada de la dirección:

http://localhost:port/sintX/P2Lib/v1/

Los detalles de esta API están recogidos en el documento API_REST_P2.pdf en la plataforma MOOVI

Aunque el servicio será desarrollado y probado por cada alumno en su TOMCAT, la corrección se realizará con otro TOMCAT, gestionado por los profesores, y que tendrá configurado el contexto del alumno de la forma arriba indicada. Por ello, **es extremadamente importante** que se respeten los nombres mencionados.

5. Tecnologías a emplear por el servlet: DOM y JAXP

El *servlet* debe analizar los ficheros LML mediante un *parser* DOM de la librería JAXP de Java. **Sólo se puede crear un parser una única vez (por tanto, fuera de cualquier bucle)**. El resultado de la lectura de cada fichero será un árbol DOM en donde el *servlet* buscará la información necesaria para resolver la consulta. **Los ficheros se leerán una sola vez**, y el tiempo que dure este proceso debe ser razonable¹. El parser se limitará a leer los documentos, sin extraer información de los mismos para clasificarla en estructuras de datos propias de cada práctica².

Toda la gestión del modelo de datos de la base de información (incluido el proceso de búsqueda de ficheros) estará ubicada en un fichero Java llamado *DataModel.java*.

Para la realización de la práctica no se podrá emplear sin permiso ninguna librería o tecnología que no esté instalada en los ordenadores del laboratorio. Un *servlet* nunca debe realizar la llamada 'System.exit()'.

6. Implementación del servicio

La práctica se realizará de forma estructurada, sangrando apropiadamente el código para facilitar su lectura. Las variables tendrán nombres significativos.

Cada alumno escribirá una serie de métodos que **deben recibir los parámetros indicados, buscar la información y devolver (ya ordenados por identificador en sentido creciente, es decir, primero los menores) los datos de la respuesta a cada fase de la consulta**, independientemente del tipo de consulta que sea. Los métodos para la consulta son:

- ☐ *ArrayList<Author> getAuthors(String countryId)*
- ☐ *ArrayList<Book> getBooks (String authorId)*
- ☐ *ArrayList<Country> getCountries ()*
- ☐ *Author getAuthor(String authorId)*
- ☐ *Country getCountry(String countryId)*
- ☐ *Book getBook(String bookId)*

Los nombres y parámetros son los indicados, todos obligatorios, y no se permite ninguno adicional.

Los tipos de datos *Author*, *Book* y *Country* serán clases definidas en la práctica para agrupar toda la necesaria que se usará posteriormente para crear la respuesta. Deberán contener, al menos, una variable miembro llamada *identificador* que actuará como clave primaria.

El servicio debe funcionar independientemente de su ubicación, es decir, cambiando el directorio *webapps* a otro ordenador y ruta. Para ello, **todos los URLs deben ser relativos, no puede haber en los ficheros fuente ningún URL o nombre de fichero con ruta absoluta, ni mención explícita a las carpetas anteriores a webapps**.

Los ficheros de la práctica 2 serán exclusivos de la misma, no compartidos con ninguna otra práctica. Para ello, para la ejecución de la práctica, los ficheros que se lean estarán ubicados dentro del directorio *webapps/p2*, mientras que las clases a ejecutar estarán en el directorio *webapps/WEB-INF/classes/p2* (es decir, formarán el paquete Java *p2*).

7. Entrega de la práctica

Para notificar la conclusión de la práctica, es obligatorio entregar la práctica (**entrega no preliminar**) en la correspondiente actividad de MOOVI, subiendo un fichero "readme.txt" que incluya la información relevante para corregir la práctica (al menos, el identificador `sintX` del usuario).

Tras la entrega de la práctica se podrán corregir todos los errores identificados, momento en el que se obtendrán los puntos de la práctica. Sin embargo, tal como se describe en la guía docente de la asignatura, los incumplimientos reiterados de la especificación podrán dar lugar a una penalización en la nota final de la asignatura.

Una copia de todos los ficheros fuente (*java*, *Makefile*...) debe quedar en el directorio *public_html/p2* de la cuenta del alumno, junto con el fichero "readme.txt". En ese directorio debe estar todo el código desarrollado, y los ficheros deben estar limpios, sin que haya en su interior código anterior comentado que ya no se use. Este será el código que se revisará por parte de los profesores, para corregir o para ayudar en la solución de algún problema, por lo que es **muy importante** que siempre esté actualizado. El nombre de todos los ficheros con código en esa carpeta deberá acabar con el número de cuenta de cada alumno.

La situación de los ficheros fuente y del contexto del *TOMCAT* que incluye los *servlets* debe ser la que se indica en este documento. También los nombres de los ficheros deben ser los especificados, incluyendo mayúsculas y minúsculas.

Para la corrección de la práctica, los ficheros y carpetas de la cuenta deben tener los siguientes permisos:

- ☐ La raíz de la cuenta tendrá permisos 750.
- ☐ La carpeta *public_html* y sus descendientes tendrán permisos 755.
- ☐ Los ficheros descendientes de *public_html* tendrán permisos 644.

Aclaraciones

- ☐ Si no se especifica lo contrario, el orden de los elementos será según identificador de los elementos a devolver y de menor a mayor, es decir, en orden creciente.
- ☐ Los identificadores serán siempre únicos.

Los ficheros HTML generados no podrán contener ninguna información sobre la presentación. Todos estos aspectos (color, tamaño, etc) deberán ser provisto a través de una CSS externa al fichero HTML.

- ☐ Todos aquellos que estén subespecificados o que puedan generar confusión, serán aclarados en el foro de la práctica 2 de Moovi. Esta fuente será la única válida para fijar los detalles de la práctica.

Apéndice A: Reglas del lenguaje LML

- ☐ El elemento raíz se llamará *libreria*.
- ☐ El contenido de esta raíz serán los elementos
 - **libro**. Contendrá el propio nombre del libro como elemento y, adicionalmente, los siguientes atributos:
 - Cada una de estas entradas contendrá los atributos:
 - **identificador**. Será una serie de caracteres numéricos de longitud hasta 10
 - **disponible**. Un valor que tomará el valor "sí" o "no". Este atributo no será obligatorio.
 - **autor**. Será el identificador del autor y se corresponderá (o no) con el de algún autor indicado en la parte correspondiente.
 - **ISBN**. Será el código ISBN del propio libro.
 - **pais**. Contendrá el nombre del país y además, como atributo (con el nombre identificador) el identificador que será una serie de caracteres numéricos de longitud hasta 10.
 - **autor**. Cada una de estas entradas contendrá el propio nombre del autor y los atributos:
 - **identificador**. Será una serie de caracteres numéricos de longitud hasta 10.
 - **nacimiento**. Un valor que indicará el año de nacimiento del autor.