

TRABAJO PRÁCTICO ESPECIAL

DISEÑO Y PROCESAMIENTO DE DOCUMENTOS XML

INFORME - GRUPO 3

Integrantes del grupo:

N° de Legajo	Nombre	Correo electrónico
64331	Sebastián Caules	scaules@itba.edu.ar
64045	Alexis Herrera Vegas	aherreravegas@itba.edu.ar
64612	Andrés Cortese	acortese@itba.edu.ar
64756	Jerónimo Esquivel	tesquivel@itba.edu.ar

Introducción

El trabajo práctico especial consiste en la recolección de una serie de datos guardados en la Web, que serán filtrados y organizados de manera estipulada y presentados en formato PDF. Los datos a extraer y procesar están relacionados a la competencia automovilística NASCAR y se encuentran en la API SportRadar. El objetivo del trabajo práctico es completar estas tareas mediante los conocimientos adquiridos durante la materia sobre las herramientas de consulta y transformación de documentos XML, además del correcto trabajo en equipo.

Desarrollo

En primer lugar, se accede a la API SportRadar mediante el uso de una Free API Key. La API SportRadar cuenta con un gran número de datos sobre diversos deportes y competencias, que se pueden acceder libremente y de forma gratuita. Para este trabajo, se utilizará la información guardada sobre NASCAR.

Mediante el comando `curl` se realizan los llamados a la API y obtenemos la información y datos necesarios. En el llamado a la API, se precisa el tipo y el año de la competencia de la cuál se quiere recoger la información. Las especificaciones de tipo y año se deben ingresar respectivamente como parámetros al ejecutar el archivo de bash `tpe.sh`, donde se encuentra la totalidad de instrucciones hasta la creación del PDF final. Además, previamente se debe setear la variable de entorno `$SPORTRADAR_API` con la clave para acceder correctamente a la API. La forma correcta de ejecutar este archivo es ingresando la siguiente línea en la terminal:

```
$> bash tpe.sh type year
```

De esta forma, se obtienen dos documentos en formato XML: `drivers_standings.xml`, que contiene la clasificación de corredores en ese año y competencia (o división), y `drivers_list.xml`, que contiene la información detallada de los corredores. Para la tabla presentada en forma PDF, se desea tomar parte de la información que se encuentra en estos documentos, entonces se debe crear un archivo XQUERY para filtrar la información y crear un nuevo documento XML solo con los datos requeridos. Antes, para facilitar esta última acción de filtrado, se utiliza un archivo XSL llamado `remove_namespace.xsl` que elimina los namespaces al inicio de cada documento obtenido de la API.

Entonces, con el XQUERY ya mencionado, llamado `extract_nascar_data.xq`, se arma el documento `nascar_data.xml` que contiene todos los datos ya filtrados.

El siguiente paso consta en la transformación del archivo XML en un documento PDF para presentar correctamente la información. Para esto se utiliza el archivo XSLT `generate_fo.xsl` que genera un archivo XML XSL-FO `nascar_page.fo` y donde se especifica el formato de la tabla final, cumpliendo con los requisitos pedidos en el enunciado del TP.

Finalmente, se utiliza el comando `fop` proveniente del proyecto Apache FOP instalado previamente para transformar el XML XSL-FO saliente de la consulta XSL en el documento PDF final, que posee las características pedidas.

El manejo de errores se encuentra en el archivo BASH `tpe.sh`. Si se encuentran errores, el archivo `nascar_data.xml` no se completa con los datos a presentar sino que enumera los errores encontrados. Luego se sigue el mismo proceso y se presenta un PDF con un listado de los errores. Los posibles errores que se pueden encontrar se deben al mal pasaje de parámetros al invocar a `tpe.sh`. Cabe aclarar que el tipo debe ser uno de entre `sc`, `xf`, `cw`, `go` y `mc`; y el año debe estar entre 2013 y 2024. Además, no hay que olvidar de setear la variable de entorno con la clave de la API, ya que esto también produce un error.

Las dificultades que encontramos durante el desarrollo del trabajo práctico fueron varias. Al hacer instantáneamente dos llamados a la API, solía ocasionar el error de tipo “Too Many Requests”, que solucionamos con un `sleep` de dos segundos. También, a la hora de realizar el XQUERY, tuvimos problemas para abrir los archivos XML iniciales, lo que nos requirió investigar una solución y que solucionamos con las siguientes líneas de código:

```
let $drivers_list := doc("drivers_list.xml")
let $drivers_standings := doc("drivers_standings.xml")
```

Por último, en cuanto a la distribución de las tareas dentro del equipo, hubo cooperación de todos los integrantes en todas las tareas, así que ningún fragmento del trabajo práctico fue realizado enteramente por una única persona. Si intentamos asignar roles, podemos decir que Sebastián Caules, Alexis Herrera Vegas y Andrés Cortese se ocuparon primordialmente del funcionamiento de la consulta xQuery y de la plantilla XSLT y Jerónimo Esquivel fue el principal encargado de la presentación.

Conclusión

En conclusión, se lograron los objetivos planteados de recolección de datos de una API en la Web, filtrado de información mediante una consulta XQUERY y generación y transformación de los archivos con el uso de XSLT y los recursos brindados por el Apache FOP, para finalmente presentar los datos prolijamente en un documento PDF, que cumple con la consigna brindada por la cátedra.