

Proyecto REST

Profesor: Sebastián Salazar

Asignatura: Computación Paralela y Distribuida

Integrantes: Nicole Sáez

Vicente Sepúlveda

Fecha: 11/07/2021

**Introducción**

La expresión Application Programming Interface o en español, Interfaz de Programación de Aplicaciones, originó el acrónimo API.

Las API son «traductores» cuya función es conectar sistemas, software y aplicaciones. Con las API es posible ofrecerle una experiencia de uso más familiar a las personas.

Las API le permiten al usuario final utilizar una aplicación, software o incluso una simple hoja de cálculo, consultando, cambiando y almacenando datos de diferentes sistemas, sin que el usuario tenga que ingresar a ellos, directamente.

El propósito de una API es intercambiar datos entre diferentes sistemas, la mayoría de las veces estos intercambios de datos tienen como objetivo automatizar procesos manuales y / o permitir la creación de nuevas funcionalidades.

Por ejemplo, un software de marketing recibe datos de un cliente potencial y usando una API, se envía la información de un cliente potencial calificado al software CRM. En este momento se da un intercambio de datos para automatizar un proceso manual.

A continuación, se presentará el desarrollo de la actividad 3 de Computación Paralela y Distribuida, al cual trata de desarrollar una API REST la cual debe entregar a través de un JSON los datos entregados por sismología.cl, por lo cual presentaremos la problemática, la resolución del problema, las tecnologías usadas y unas conclusiones.

**Problemática**

En el mundo tecnológico en el que nos desenvolvemos, las aplicaciones móviles tienen cada vez más importancia en el mercado, por lo que el desarrollo de aplicaciones backend (que apoyan este tipo de aplicaciones) es de vital importancia en el mundo actual, por lo que los servicios rest han adquirido un lugar preponderante en la estrategia de este tipo de soluciones.

Dado que Chile es un país sísmico, debido a su geografía hay un extenso territorio que constantemente sufre de movimientos sísmicos, de diferentes intensidades y que afectan a las personas de distintas formas. En este proyecto se nos solicitó desarrollar una aplicación REST que sea capaz de conectarse al Centro Sismológico Nacional, dado que este servicio no ofrece una api para obtener la información que recopilan sobre distintos eventos, pero dejan un reporte de los resultados en la página: <http://www.sismologia.cl/links/ultimos_sismos.html>

El servicio rest requerido debe contar con:

* Mecanismo de autenticación: Para consumir el servicio API, se debe contar con una autenticación
* Modelo de datos: modelo de datos para almacenar la información histórica en base de datos con PostgreSQL.
* Scraping: De la pagina de sismología debemos recolectar los datos periódicamente, para almacenar la información histórica.
* Operación REST: El servicio rest debe tener al menos una operación
* Soporte Cors: El servicio debe soportar CORS
* Docker: La aplicación debe estar disponible para ser dockerizada

**Resolución de problemática**

En este proyecto utilizamos Spring Framework, partimos generando un proyecto Spring Boot con Spring Initializr, el cual nos facilitó el poder agregar las dependencias que utilizaríamos en nuestra API, así como también nos fue de gran utilidad para poder manejar los datos y demas apartados de nuestro proyecto.

Utilizamos el siguiente modelo de datos para poder almacenar los datos de los sismos que entregaremos:

pk **bigserial** **not** **null**,

fecha **varchar**(255) **not** **null**,

fechautc **varchar**(255) **not** **null**,

latitud **float8** **not** **null**,

longitud **float8** **not** **null**,

profundidad **int** **not** **null**,

magnitud **float8** **not** **null**,

agencia **varchar**(255) **not** **null**,

referencia\_geografica **varchar**(255) **not** **null**,

Para poder obtener la información de la página de sismología utilizamos la librería Jsoup, la cual nos permitió recibir y descomponer la tabla que entrega la página con el fin de poder agregar el contenido a la base de datos, función la cual se ejecuta seguidamente luego de levantar la API, por lo que se programó para que esta se recompile cada 5 horas y pueda ingresar posibles nuevos sismos.

Respecto a las autorizaciones de nuestra API utilizamos JWT de forma que sea requerido un Bearer Token para poder consumir el JSON con los sismos guardados en la base de datos, Bearer el cual se puede obtener en <http://127.0.0.1:8080/G13/Auth>. Con en verbo POST

A través de headers

* accept: application/json;charset=utf-8
* Content-Type: application/json

Body

* "app": "utem"
* "password": "1234"

El cual también se puede obtener a partir del siguiente cURL:

curl -X POST "http://127.0.0.1:8080/G13/Auth/" -H "accept: application/json;charset=utf-8" -H "Content-Type: application/json" -d "{\"app\":\"utem\",\"password\":\"1234\"}

Respecto a poder consumir la API en si con los registro de sismos en formato JSON junto al Bearer generado en la autorización se puede consumir en <http://127.0.0.1:8080/G13/earthquakes>. con el verbo GET

A través de headers:

* accept: application/json;charset=utf-8
* Content-Type: application/json
* Authentication: $bearer (Bearer obtenido en la autenticación anterior(Auth))

El cual también se puede obtener a partir del siguiente cURL:

curl -X GET "http://127.0.0.1:8080/G13/earthquakes/" -H "accept: application/json;charset=utf-8" -H "Authentication: $bearer"

Además, se implementó el servicio Swagger donde se puede ver la documentación de la API:

http://127.0.0.1:8080/G13/swagger-ui/

**Tecnologías usadas**

**Spring:** Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java

**Java:** La plataforma Java es el nombre de un entorno o plataforma de computación originaria de Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación Java u otros lenguajes que compilen a bytecode y un conjunto de herramientas de desarrollo

**Jsoup:** Jsoup es una biblioteca Java de código abierto diseñada para analizar, extraer y manipular datos almacenados en documentos HTML

**Maven:** Maven es una herramienta de software para la gestión y construcción de proyectos Java

**Docker:** Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos

**Swagger:** Swagger es un conjunto de herramientas de software de código abierto para diseñar, construir, documentar, y utilizar servicios web RESTful.

**JWT:** JSON Web Token es un estándar abierto basado en JSON propuesto por IETF para la creación de tokens de acceso que permiten la propagación de identidad y privilegios o claims en inglés

**CORS:** El intercambio de recursos de origen cruzado o CORS es un mecanismo que permite que se puedan solicitar recursos restringidos en una página web desde un dominio diferente del dominio que sirvió el primer recurso.

**PostgreSQL:** PostgreSQL, también llamado Postgres, es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL

**Conclusión**

Con este trabajo pudimos trabajar con un nuevo sistema y entender de mejor manera el funcionamiento de las API y su importancia en la actualidad la cual cada día está ganando más y más reconocimiento.

Así como conocer por primera vez un Framework para poder trabajar con APIs y el proceso de autenticaciones con JWT, a demás de trabajar con una base de datos PostgreSQL, la cual no es muy distinta a la enseñada a lo largo de la carrera como era SQLServer, así como conocer el método de poder obtener información de una página web con scrapping.

Dado los múltiples usos de las API podemos obtener nuevos recursos para nuestros futuros proyectos, por lo cual como equipo estamos agradecidos con la problemática presentada dado que nos aporta de sobremanera en nuestra formación profesional.