

Proyecto REST

10 de julio del 2021

VISIÓN GENERAL

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. Estas aplicaciones pueden estar pensadas para presentar información directamente a los usuarios finales, para interoperar con otros sistemas o una mezcla de estos comportamientos.

OBJETIVOS

1. Comprender el funcionamiento del protocolo HTTP (sus verbos y estados).
2. Comprender el funcionamiento de aplicaciones stateless, mecanismos asíncronos y funcionamiento REST.

ESPECIFICACIONES

Chile es un país sísmico, debido a su geografía hay un extenso territorio que constantemente sufre de movimientos sísmicos, de diferente intensidad y que afectan a las personas de diferentes formas. En este proyecto se solicita que desarrolle una aplicación rest que sea capaz de conectarse al Centro Sismológico Nacional¹ (dependiente de la Universidad de Chile), este servicio **NO** ofrece un api para obtener la información que recopilan sobre distintos eventos, pero dejan un reporte de los resultados en la siguiente página:

http://www.sismologia.cl/links/ultimos_sismos.html

¹<http://evtdb.csn.uchile.cl/>

Su servicio rest debe contar:

- **Mecanismo de autenticación.** Para consumir el servicio API, esta debe al menos usar un `api_key`, aunque `jwt` y `auth` serán bien considerados. Cada grupo es libre de implementar las operaciones que necesite para cumplir con este punto.
- **Modelo de datos.** Será necesario definir el uso de un modelo de datos, para almacenar la información histórica. La restricción es que se debe usar PostgreSQL.
- **Scraping** de la página del Centro Sismológico Nacional de la Universidad de Chile, lo cual permitirá disponer de información histórica.
 - http://www.sismologia.cl/links/ultimos_sismos.html
 - Es importante que este proceso se ejecute de forma periódica, para disponer de una base de datos actualizada.
- **Operación REST.** Este servicio rest debe disponibilizar al menos una operación (protegida con algún mecanismo de autenticación):
 - `{protocolo}://{ip}:{puerto}/{contexto}/earthquakes`
 - `{protocolo}` Indica si es `http` o `https`.
 - `{ip}` Indica la ip (pública) para realizar la conexión.
 - `{puerto}` Indica el puerto en el cual el servicio está escuchando.
 - `{contexto}` Es una ruta base específica por grupo.
 - La operación REST de consulta debe ser fija en `/earthquakes`
 - El verbo debe ser `GET`
- **Soporte Cors.** El servicio debe soportar CORS.²
- **Salida Exitosa:**
 - Código HTTP: 200
 - Tipo MIME: `application/json`
 - Estructura de datos: Colección de Objetos
 - Objeto atómico:
 - **Fecha.** Fecha y hora local expresada en la zona horaria America/Santiago del evento sismológico.
 - **Latitud.** Escalar de tipo “double”.
 - **Longitud.** Escalar de tipo “double”.
 - **Profundidad.** Escalar de tipo “int”, la unidad que representa son Km.
 - **Magnitud.** Escalar de tipo “double”, la unidad que representa son MI.
 - **Referencia Geográfica.** String con una pequeña descripción.
- **Salida errónea.**
 - Se debe discriminar según el problema detectado, en función de eso se debe generar una código HTTP acorde al problema, ejemplo:

² <https://developer.mozilla.org/es/docs/Web/HTTP/CORS>

-
- 401. Sin autorización.
 - 403. Sin permiso.
 - 404. No encontrado.
 - 410. Expirado.
 - 412. Precondición Fallida
 - ...
 - Tipo MIME: application/json
 - Estructura de datos: Objeto
 - Objeto atómico:
 - **Fecha:** Fecha y hora local expresada en la zona horaria America/Santiago de la ejecución del problema (Hora actual).
 - **Mensaje:** Glosa descriptiva, en lenguaje natural del error que se pueda presentar al cliente final.
 - **Docker.** La aplicación debe estar disponible para ser dockerizada. El resultado de esta aplicación será usada en la evaluación móvil, todos los grupos compartirán el mismo servidor aunque cada uno tendrá un puerto específico.

Equipo.

El proyecto deberá ser realizado grupal, con los equipos conformados en clases.

Informe.

Se debe presentar un informe con la siguiente estructura:

- Debe tener una introducción.
- Debe indicar la forma de resolver el problema.
- Debe indicar y justificar la tecnología usada.
- Debe tener una conclusión clara y acorde a lo establecido en el desarrollo del informe.
- No debe superar las 15 páginas.

Código.

El código debe ser entregado a más tardar, el 10/07/2021 hasta las 23:59:59.999 horas de Santiago de Chile, este proyecto debe estar publicado en repositorio grupal de github (privado con invitación al docente).

EVALUACIÓN

Documentación.

Parte de la evaluación consiste en la documentación de las funciones. Que debe ser clara, concisa y descriptiva de lo que el código realiza. El formato puede ser swagger o openapi.

Código

El código debe ser claro, fácil de leer, ordenado y cumplir con buenas prácticas de programación, se inspeccionará el código fuente.

Resultados.

Un criterio de evaluación que se tomará en consideración: el tiempo de ejecución de la tarea y el cumplimiento de los objetivos planteados. En aquellos puntos con medida cuantitativa menos es mejor. La evaluación es porcentual.

Implementación.

El proyecto es grupal, debe realizarse por los distintos equipos formados en la asignatura.

Cada grupo puede implementar la aplicación en el lenguaje que prefiera, siempre y cuando sea un lenguaje open source, y que se pueda instalar nativamente (desde el repositorio base) en Ubuntu 20.04 LTS de 64 bits. El proyecto debe cumplir con todas las normativas de licenciamiento aplicables.