

Frontend

A partir de la API DOC entregada, construya un Front que consuma los datos proporcionados por la API entregada por A2G. Puede utilizar tecnología o Framework preferencia (Angular, React, Vue, Vanilla JS).

Tenga en cuenta que algunas de las habilidades evaluadas en esta prueba son las siguientes:

- Leer documentación de API
- HTML + CSS + JS
- Consumo de datos - API
- Authentication JWT
- Endpoints con paginación
- Manejo básico de lenguaje de Backend
- Entre otros.

Breve descripción de la API

La API proporciona información de 3 entidades (Platform, Sensor, Record). Considere que cada plataforma puede tener 1 o más sensores y que cada sensor cuenta con una secuencia de datos o serie de tiempo. A continuación, se encuentra una imagen de los modelos de datos utilizados

Platform	Sensor	Record
+ id : string - guid	+ id : string - guid	id : string - guid
+ name : string	+ platformId : string - guid	sensorId : string - guid
+ fleet : string	+ name : string	ts : string - datetime
+ img : string - url	+ type : string	value : number
+ lastReport : string - url		

Para simplificar, tenga en cuenta los siguientes puntos

Platform:

- fleet: campo string, pero este solo puede tomar 6 valores distintos "Flota 1", "Flota 2", "Flota 3", "Flota 4", "Flota 5", "Flota 6"

Sensor:

- type: campo string, pero este solo puede tomar 4 valores distintos "Flota 1", "Flota 2", "Flota 3", "Flota 4", "Flota 5", "Flota 6"

El diseño de interfaces o visual es decisión de usted, pero como sugerencia considere 3 vistas:

1. Login
2. Dashboard de Plataformas
3. Detalles de Plataforma:

Ejercicio Adicional

En los datos de tipo Platform, cuando se consume el endpoint `"/api/Platforms/{id}"`, hay un campo llamado `"lastReport"` el cual es una url de S3. Se pide añadir un botón que al presionarlo renderice dinámicamente el contenido de la url en la vista actual.

Nota: No es necesarios en esmerarse en los detalles de visuales (efectos, paleta de colores, etc).

Mini prueba

Ocupando el lenguaje de Backend de preferencia (Python, C#, JS-Node, etc.), resuelva los siguientes ejercicios en un script.

Pregunta 1

Suponga que el sensor con Id `"fb76277a-5872-4d74-a80b-4cce592c9e12"` es un sensor de ruido, los valores del sensor de ruido se clasifican como muestra la siguiente tabla

Categoría	Rango
Bajo	[0, 60)
Medio	[60, 120)
Alto	[120, 200]

Analizando la serie de tiempo completa del sensor, ¿Cuántos registros hay en cada categoría?

Pregunta 2

Ocupando los datos del mismo sensor `"fb76277a-5872-4d74-a80b-4cce592c9e12"`, ¿cuántos datos hay entre el 10-04-2023 08:00 y las 11-04-2023 20:00 ($t1 \leq TS \leq t2$) en zona horaria de Santiago?, tenga en cuenta que la fecha en el campo TS son entregados en zona horaria UTC.

Ejercicio 3

Teniendo los resultados de la pregunta 1 y 2, envíe sus resultados mediante una petición HTTP al endpoint POST `"/api/Result"`. Recuerde que el endpoint se encuentra protegido (Ver API DOC).