

Definición OpenAPI

Microservicio “Monitorización de tráfico y predicciones”

Version: 0.1

Realizado por: Enrique Fernández Sánchez

Indice

CRUD Enpoints.....	3
Networks.....	3
Resumen.....	3
Definición.....	3
Interfaces.....	3
Resumen.....	3
Definición.....	4
Forecast.....	4
Resumen.....	4
Definición.....	4
Prophet.....	5
Non-CRUD Endpoints.....	6
Query data.....	6
Resumen.....	6
Definición.....	6
Modelos de datos.....	7
Networks.....	7
Interfaces.....	7
Forecast.....	7
Reference links.....	7

CRUD Endpoints

- “/networks” → Acceso a la información de una red.
- “/interfaces” → Acceso a la información de interfaz de una red.
- “/forecasts” → Ejecución de forecast respecto a unos datos de entrada.

Networks

Resumen

CRUD asociada a la creación de una red en la que aplicaremos la monitorización. Dentro de la red, tendremos diferentes interfaces que serán dadas de alta en la aplicación para ser monitorizadas. Una única red puede tener muchas interfaces en su interior.

Propuesta de endpoint asociado: “/nets” o “/networks”

Definición

Recurso	GET	POST	PUT	DELETE
Colección de redes: /networks	Lista de redes dadas de alta	Añade una nueva red. Se le asigna ID automaticamente	*	*
Red en particular: /networks/<id_net>	Información de una red en particular	*	Modificamos la información de una red	Eliminamos una red

* Denegado el acceso

Interfaces

Resumen

CRUD asociada a la creación de interfaces dentro de una red a monitorizar. Sirve para dar de alta las interfaces que queremos que sean monitorizadas. Una interfaz puede tener muchas muestras guardadas, dichas muestras están almacenadas en la base de datos para time series InfluxDB (<https://www.influxdata.com/>)

Propuesta de endpoint asociado: “/nets/<id_net>/interfaces” o “/networks/<id_net>/inteface>”

Definición

Recurso	GET	POST	PUT	DELETE
Colección de interfaces: /networks/<id_net>/interfaces	Listado de interfaces dentro de una red	Añade una nueva interfaz a una red. Se asigna ID auto.	*	*
Interfaz en particular: /networks/<id_net>/interfaces/<id_if>	Información de una interfaz en particular	*	Modificamos la información de una interfaz	Eliminamos una interfaz

* Denegado el acceso

Forecast

Resumen

CRUD reducida asociada a la creación de predicciones dada una red a monitorizar. Permite lanzar las predicciones y obtener los resultados asociados a ellas.

Propuesta de endpoint asociado: “/forecasts/<id_forecast>”

Definición

Recurso	GET	POST	PUT	DELETE
Colección de predicciones: /forecasts	Listado de predicciones	Creamos una predicción [Usamos JSON file para delimitar la predicción]	*	*
Predicción en particular: /forecasts/<id_fore>	Información de un forecast	*	*	Eliminamos una predicción
Ejecución de una predicción: /forecasts/<id_fore>/exec	*	Ejecutamos la predicción	*	*

* Denegado el acceso

Prophet

Paquete de Python desarrollado por Facebook (Open Source: <https://github.com/facebook/prophet>), que permite realizar predicciones sobre una serie de datos, siguiendo un modelo no lineal basado en tendencias, que puede seguir diferentes tendencias a la vez (anual, semanal, diario o efectos de vacaciones).

Página principal: <https://facebook.github.io/prophet/>

Documentación: https://facebook.github.io/prophet/docs/quick_start.html

Parametros configurables dentro de Prophet:

- **“holidays”** → dataframe con columnas “ds” (fecha: %Y-%M-%d) y “holiday” (nombre). Permite configurar las fechas de vacaciones. Además, está la posibilidad de usar al función: `“add_country_holidays(country_name=‘ES’)”` para utilizar las vacaciones precargadas. [ref]
- **“seasonality”** → Valor default 10. Si queremos detectar mayor frecuencia de cambios, y que en general sean menos suaves las variaciones, podemos incrementar el valor. [ref]
- **“seasonality_mode”** → by default, prophet no utiliza más de un factor de seasonality, por lo que para tener en cuenta todos los efectos, y obtener buenos resultados en datasets que tienden a crecer con el tiempo siguiendo diferentes seasonality (como es el crecimiento del tráfico), hay que usar: `“seasonality_mode=‘multiplicative’ ”` [ref]

Propuesta → Modificar funcionamiento de Prophet según un archivo de configuración en JSON que se envía en el POST del forecast.

Propuesta → Almacenar dataframe de salida de Prophet para que el usuario pueda descargarlo en raw.

Ejemplos prácticos en Python:

- <https://www.oncraw1.com/technical-seo/forecasting-seo-traffic-prophet-python/>
- (Interesante) <https://vijayv500.medium.com/time-series-forecasting-using-facebooks-prophet-in-python-1e13ea20a52b>
- <https://www.kaggle.com/code/sudosudoohio/forecasting-web-traffic-with-prophet-in-python/notebook>
- (Interesante) <https://pbpython.com/prophet-overview.html>

Non-CRUD Endpoints

- “/query/<id_net>/values” → Realizar peticiones de información sobre la información de monitorización almacenada.

Query data

Resumen

Endpoint que permite hacer queries para obtener datos acerca de la monitorización que esta almacenada en la aplicación.

Propuesta de endpoint: “/query/<id_net>/values” o “/query/<id_net>/values/timeseries”

Definición

Para ejecutar una petición, se realiza un GET a “/query/<id_net>/values”, para filtrar los datos podemos utilizar *Query Strings*.

Query_strings disponibles:

- “keys” → Podemos seleccionar entre “tx”, “rx” ~~(o “both”?)~~
- “startTs” → Epoch de inicio para devolver los datos (ms)
- “endTs” → Epoch de fin para devolver los datos (ms)
- “limit” → Limitar el número de puntos de salida
- “interval” → Intervalo entre muestras (ejemplo: 600000 ms → 10 min)

El servidor devuelve:

```
{“<keys>”: [{“ts”: <epoch>, “value”: “<valor>”}, {“ts”: <epoch>, “value”: “<valor>”}, ...]}
```

Modelos de datos

Networks

Networks
<ul style="list-style-type: none">• “id_net”: Int, PubK, Unique• “name”: String(50)• “ip_red”: String(20)• “influx_net”: String(20)

Interfaces

Interfaces
<ul style="list-style-type: none">• “id_if”: Int, PubK, Unique• “name”: String(50)• “host”: String(20)• “ip_interface”: String(20)• “subnet_mask”: Int (o String?)• “influx_if”: String(20)

Forecast

Forecasts
<ul style="list-style-type: none">• “id_fore”: Int, PubK, Unique• “id_net”: Int• “result”: String(20)

Reference links

- <https://blog.stoplighlight.io/crud-api-design>