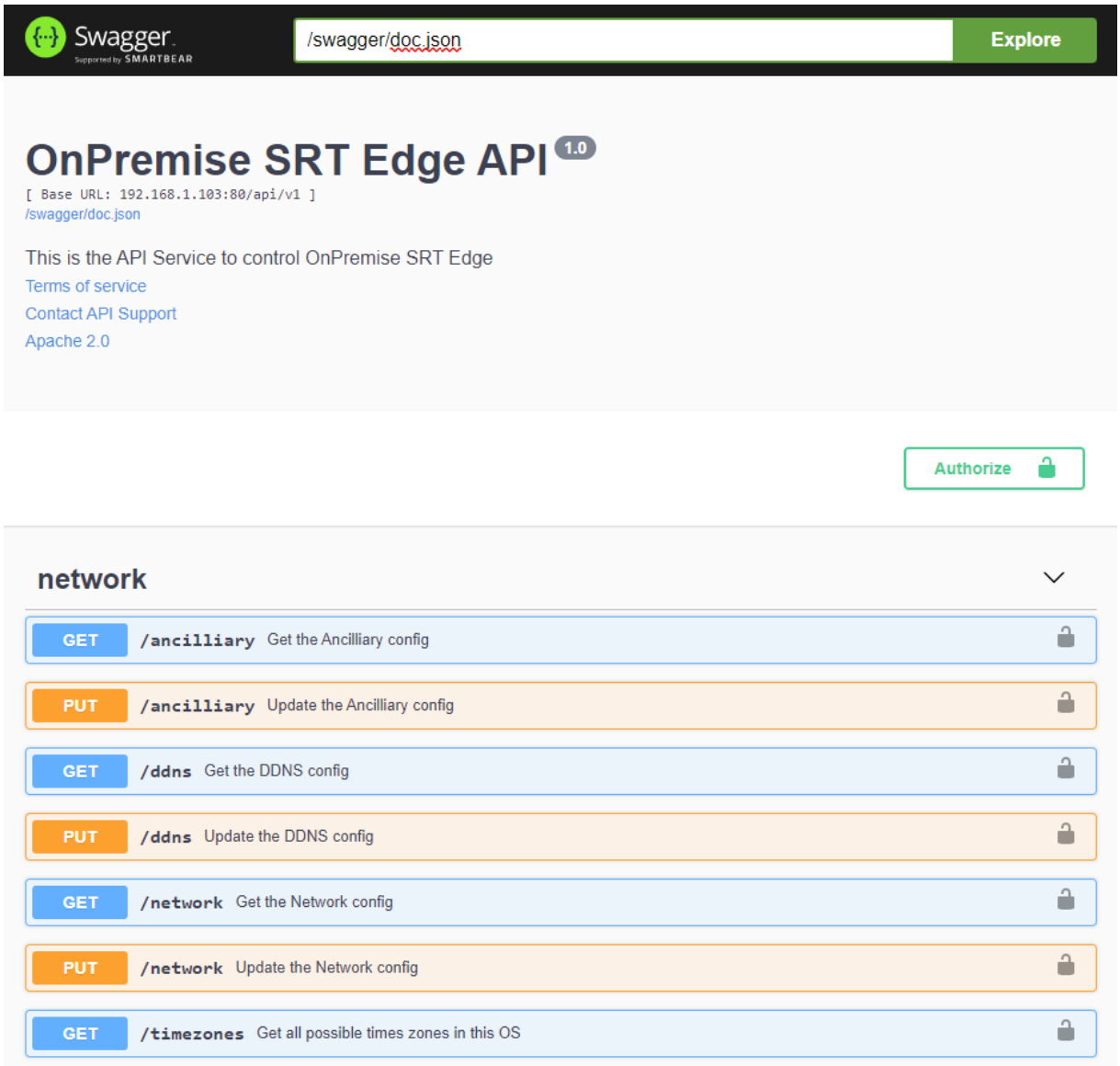
API REST OnPREMISE

Introducción

Todos los softwares de la serie OnPremise usan un API REST para desplegar toda su funcionalidad y el formato JSON para recibir y entregar los modelos de datos que intercambia.

Todas las llamadas y modelos de datos usados por el API están auto-documentados mediante Swagger 2.0 por el mismo servidor HTTP que despliega el back-end. Desde la misma red local del dispositivo con el software instalado, se puede acceder a dicha documentación con un navegador cualquiera usando la URL: http://ip_local_equipo/swagger .

El interfaz Swagger, no solo documenta cada componente, si no que también puede ser usado para probar dichos componentes. Desde la misma web se puede bajar el fichero doc.json para ser importado en Postman para su testeo pormenorizado por parte de cualquier front-end developper.

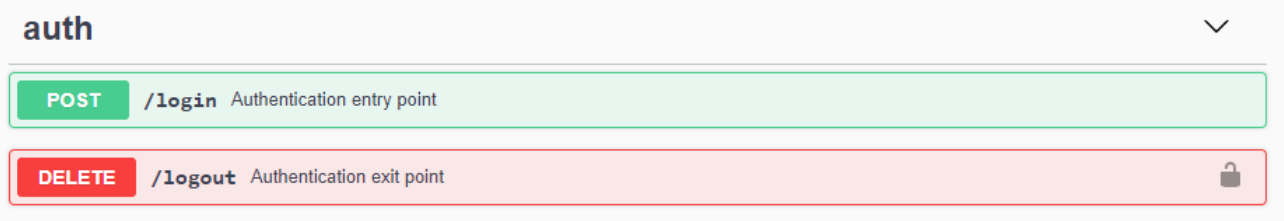


Las llamadas API que llevan candado, requieren el uso de autenticación para su funcionamiento correcto.

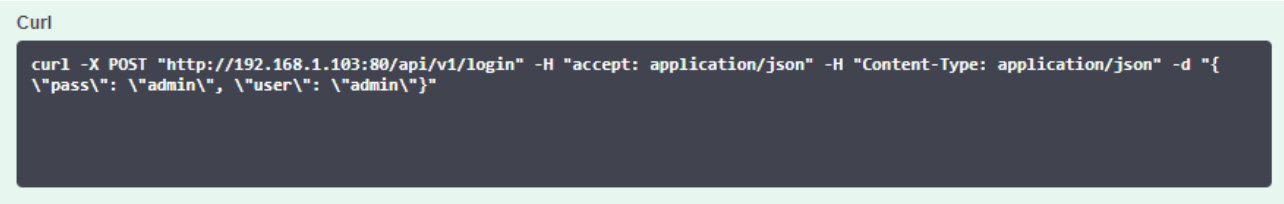
Proceso de autenticación

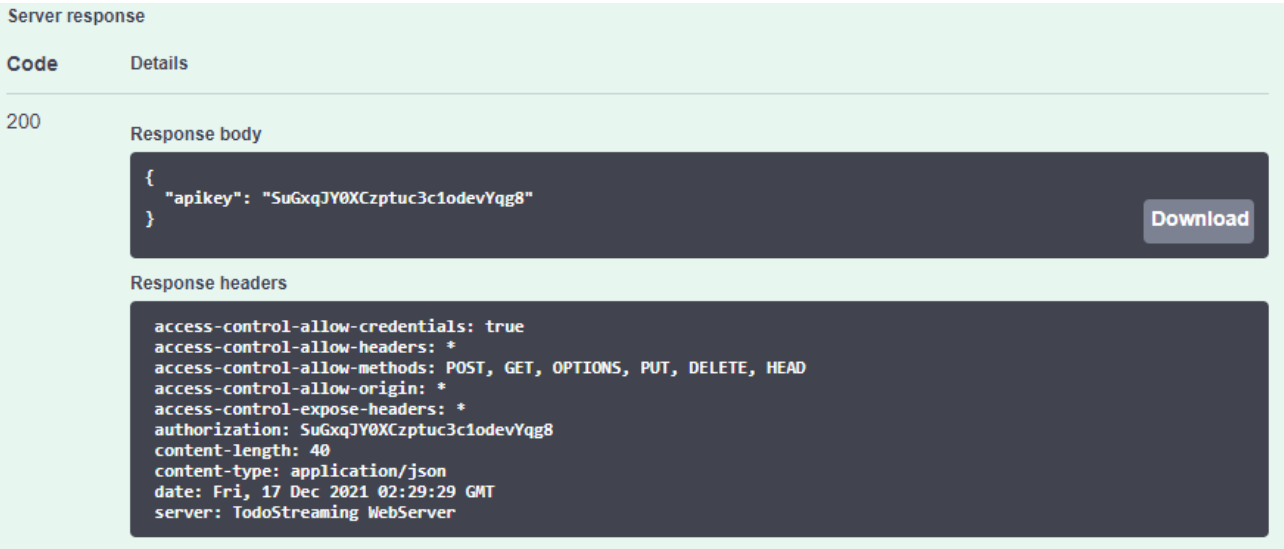
Antes de empezar a usar cualquier llamada del API protegida con autenticación, tenemos que obtener la apiKey, que usaremos en cada llamada, en la cabecera de cada request como:
Authorization: apiKey

Para ello, primeramente usaremos la llamada sin protección de la sección auth POST /login.



Usando el modelo api.Login, pasamos el nombre de usuario y la contraseña que estén guardadas en ese momento (por defecto son: admin/admin).





En este ejemplo, se recibe como respuesta con el modelo session.APIKey, el valor del apiKey SuGxqJY0XCzptuc3c1odevYqg8, que usaremos en las llamadas con candado, pasándolas en la cabecera:
Authorization: SuGxqJY0XCzptuc3c1odevYqg8

El modelo de respuesta genérico es api.HTTPResponse con esta composición:

Veamos como ejemplo de llamada autenticada, la llamada de la sección network GET /network.

Curl

```
curl -X GET "http://192.168.1.103:80/api/v1/network" -H "accept: application/json" -H "Authorization:
SuGxqJY0XCzptuc3c1odevYqg8"
```

Que nos devuelve la siguiente respuesta en el cuerpo (modelo metal.Network) y en las cabeceras:

Server response

Code	Details
200	Response body

```
{
  "interface": [
    {
      "index": 2,
      "name": "enp1s0",
      "mac": "84:47:09:08:4b:36",
      "ipv4": "192.168.1.103/24",
      "ipv6": "",
      "gateway4": "192.168.1.1",
      "gateway6": "",
      "mode4": "dhcp",
      "mode6": ""
    }
  ],
  "dns1": "212.230.135.2",
  "dns2": "212.230.135.1",
  "dns3": "",
  "mdnsif": "enp1s0",
  "monif": "enp1s0"
}
```

Download

Response headers

```
access-control-allow-credentials: true
access-control-allow-headers: *
access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, HEAD
access-control-allow-origin: *
access-control-expose-headers: *
authorization: SuGxqJY0XCzptuc3c1odevYqg8
content-length: 262
content-type: application/json
date: Fri, 17 Dec 2021 02:31:16 GMT
server: TodoStreaming WebServer
```

Cada sesión tiene una duración de 1440 segundos (24 minutos), que se actualiza en cada llamada nueva al API. Si estamos ese tiempo sin hacer llamadas que actualicen dicho timeout, la sesión caducará automáticamente y necesitaremos una nueva apiKey, con el mismo proceso descrito anteriormente.

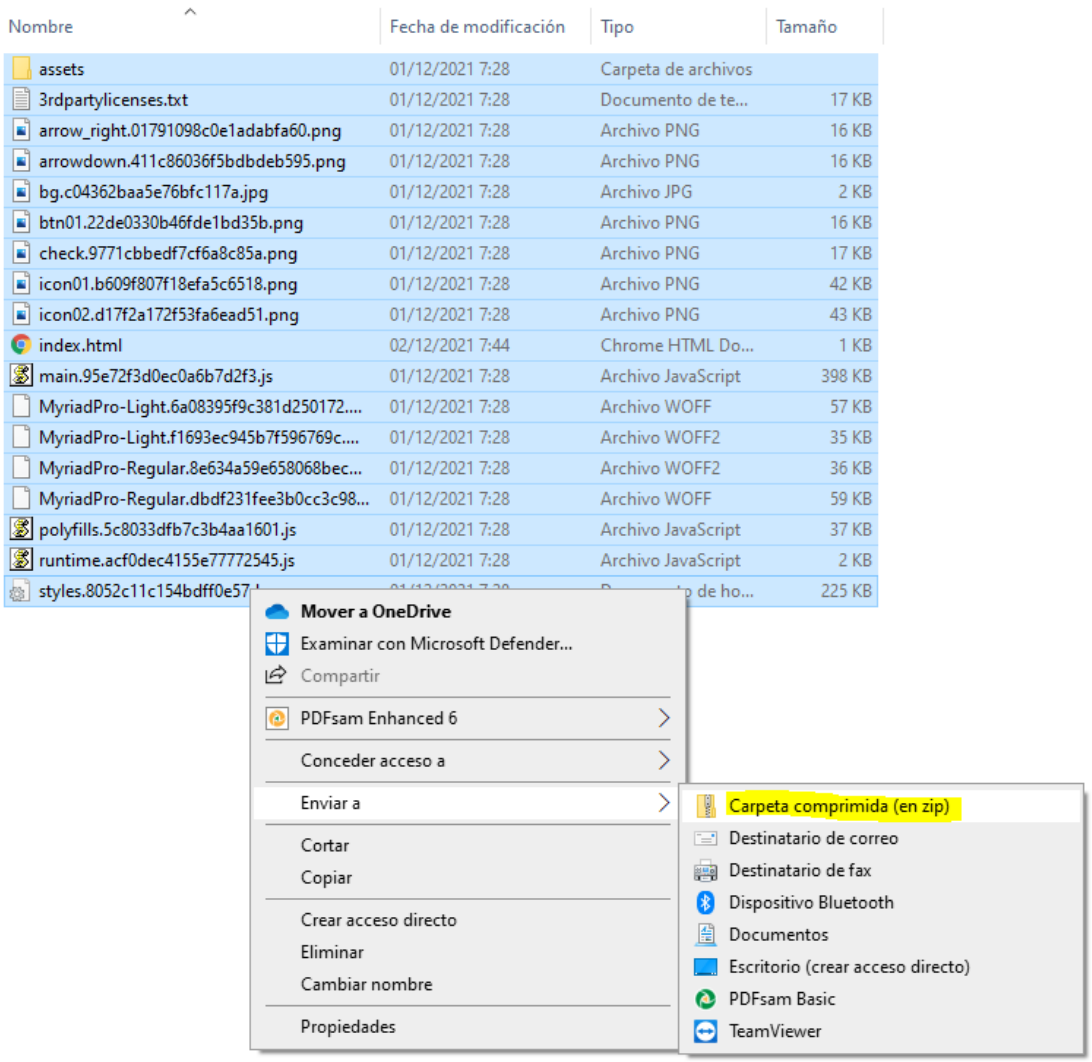
También podemos desloguearnos voluntariamente usando la llamada DELETE /logout .

El servidor HTTP cumple las reglas CORS, por lo que se puede crear una aplicación front-end que llame desde fuera al API REST completo de OnPremise.

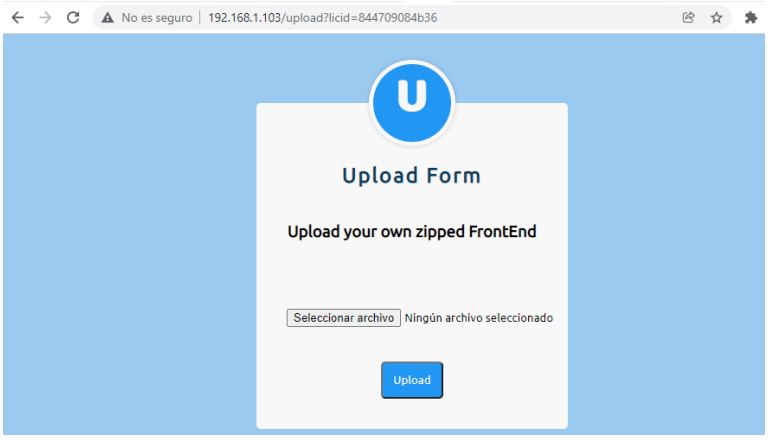
Instalar tu propio panel en el equipo

Aparte de poder usar el API desde aplicaciones externas que llamen al equipo remotamente, también puedes crear un panel front-end propio con tu propio diseño y marca. De hecho puedes probarlo antes en remoto, y si todo está como quieres, solamente hay que subirlo al equipo de la siguiente manera:

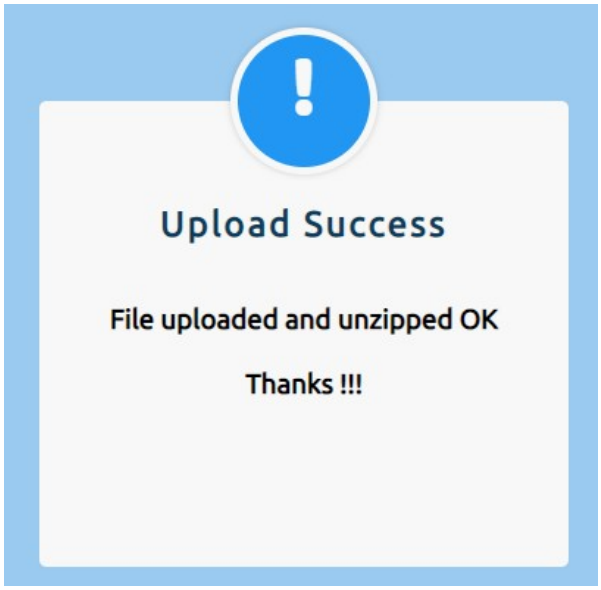
1.- Comprimimos en un zip, el raíz del front-end.



2.- Accedemos al Upload Form del equipo que está en la URL:
http://ip_local_equipo/upload?licid=licid_del_equipo



Seleccionamos el fichero zip que hemos creado en el paso 1, y pulsamos Upload.
Si todo fue bien aparecerá una pantalla como esta:



3.- Ya podemos cargar la nueva app de nuevo en nuestro navegador, actualizando la caché con Ctrl+F5 varias veces.

El panel original con el que se instalan los software OnPremise, lleva las funcionalidades más básicas y genéricas del API, y ha sido creado con Angular JS en typescript. Si estás interesado en crear tu propio panel y no dispones de personal front-end developper, puedes ponerte en contacto con nosotros para que te hagamos un presupuesto.