Artículo
David Reche · 27 ago, 2019 Lectura de 11 min

Presentación de InterSystems API Manager

¡Hola a tod@s!

Como posiblemente haya oído, acabamos de presentar InterSystems API Manager (IAM); una nueva característica de InterSystems IRIS Data Platform™, que permite monitorizar, controlar y gobernar el trafico desde y hacia APIs basadas en web, dentro de su infraestructura de TI. En caso de que se lo haya perdido, aquí está el enlace al anuncio.

En este artículo, mostraré como habilitar IAM y resaltar algunas de las muchas capacidades a las que IAM permite sacar provecho.

InterSystems API Manager trae todo lo que se necesita para:

- Monitorizar el tráfico de APIs basadas en HTTP y entender quién está usando tus APIs; cuál es la API más popular y cuál requiere retoques
- Controlar quién está usando tus APIs y restringir el uso en diferentes maneras. Permite un control detallado y capacidad de reacción rápida desde para una simple restricción de acceso hasta para una regulación del canal del tráfico de la API (throttlling) y afinado del volumen de solicitudes
- Proteger sus APIs con mecanismos de seguridad centralizados como OAuth2.0 o autenticación basada en Token
- Enrolar desarrolladores de terceros y proporcionarles una experiencia de usuario superior desde el inicio, proporcionando un Portal para Desarrolladores diseñado para sus necesidades
- Escalar la demanda de su API y procesar respuestas con baja latencia

Estoy ansioso por mostrar un primer vistazo de IAM, así que... ¡pasemos a la acción!

Comenzando

IAM está disponible como descarga desde el sitio de distribución de software de WRC y se despliega como un contenedor Docker. Así que hay que asegurarse de que cumplimos con los requisitos mínimos:

- Docker engine. Versión mínima soportada 17.04.0+.
- Herramienta docker-compose CLI. Versión mínima soportada 1.12.0+.
- IRIS versión mínima 2019.2+.

El primer paso es cargar la imagen descargada con el siguiente comando:

```
docker load -i iam_image.tar
```

Esto hace que la imagen IAM esté disponible para usar en contenedores que se ejecutan en su máquina. IAM se ejecuta en un contenedor separado del backend implementado mediante InterSystems IRIS, de manera que se puede escalar de forma independiente. Para poner en marcha IAM requiere acceso a la instancia de IRIS para cargar la información de licencia requerida. Para que IAM pueda tener acceso a la instancia de IRIS debemos iniciar sesión en IRIS y realizar las siguientes acciones:

1. Habilitar la aplicación web /api/IAM (desde seguridad/aplicaciones)
2. Habilitar el usuario IAM (desde seguridad/usuarios)
3. Establecer una contraseña para el usuario IAM

Al arrancar IAM, trata de conectarse a una instancia de IRIS y acceder a la API anterior mediante el usuario IAM para obtener los detalles de la licencia. Por lo tanto, al arrancar IAM debe conocer la URL de la API mediante (la dirección IP y el puerto sobre el que se ejecuta la API) y el nombre y contraseña del usuario IAM. Toda esta información se establece en una variable de entorno ISCIRISURL que posteriormente usará el contenedor. Algo así como la siguiente:
export ISCIRISURL=" http://IAM:password@servidor:puerto/api/iam/license"

Se dispone de un fichero docker-compose para facilitar la ejecución del contenedor, pero se debe conocer el nombre de la imagen del IAM que se va a ejecutar. Para ello también se utiliza una variable de entorno:
export ISCIAMIMAGE= intersystems/iam:0.34-1-1

La buena noticia es que hay un script que nos facilita la tarea de establecer esas variables de entorno. Dentro del fichero de IAM descargado, al descomprimir vemos que hay una carpeta "scripts" y dentro de ella está el fichero de docker-compose y los scripts para Windows y Unix (Linux/Mac).

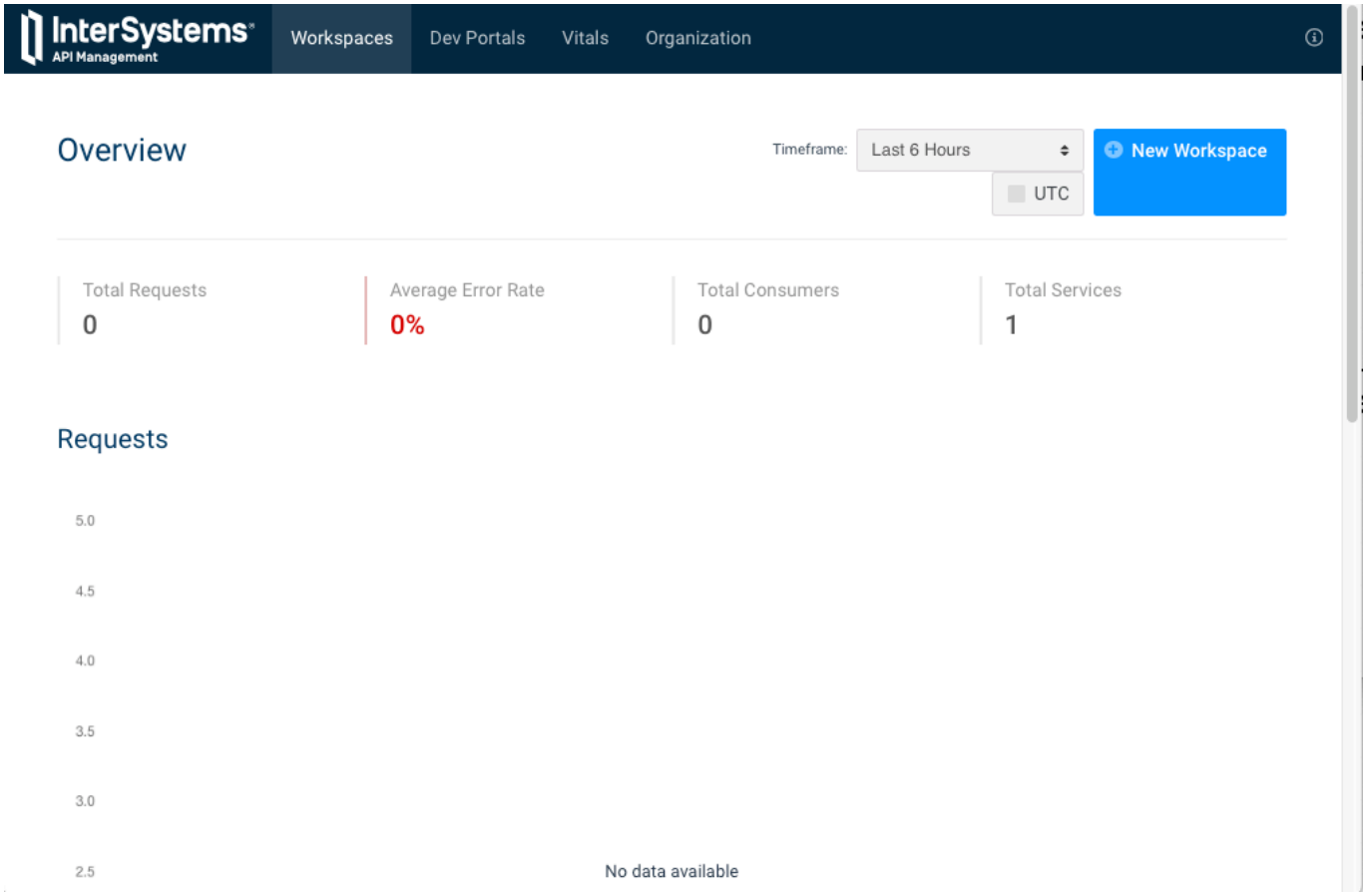
En mi caso, uso Mac y el script se debe ejecutar mediante el comando source. Así, este es el ejemplo de ejecución:

```
source ./iam-setup.sh
Welcome to the InterSystems IRIS and InterSystems API Manager (IAM) setup script.
This script sets the ISC_IRIS_URL environment variable that is used by the IAM
container to get the IAM license key from InterSystems IRIS.
Enter the full image repository, name and tag for your IAM docker image:
intersystems/iam:0.34-1-1
Enter the IP address for your InterSystems IRIS instance. The IP address has to be
accessible from within the IAM container, therefore, do not use "localhost" or
"127.0.0.1" if IRIS is running on your local machine. Instead use the public IP
address of your local machine. If IRIS is running in a container, use the public IP
address of the host environment, not the IP address of the IRIS container.
xxx.xxx.xxx.xxx
Enter the web server port for your InterSystems IRIS instance: 52773
Enter the password for the IAM user for your InterSystems IRIS instance:
Re-enter your password:
Your inputs are:
Full image repository, name and tag for your IAM docker image:
intersystems/iam:0.34-1-1
IP address for your InterSystems IRIS instance: xxx.xxx.xxx.xxx
Web server port for your InterSystems IRIS instance: 52773
Would you like to continue with these inputs (y/n)? y
Getting IAM license using your inputs...
Successfully got IAM license!
The ISC_IRIS_URL environment variable was set to:
http://IAM:****************@xxx.xxx.xxx.xxx:52773/api/iam/license
WARNING: The environment variable is set for this shell only!
To start the services, run the following command in the top level directory: docker-
compose up -d
To stop the services, run the following command in the top level directory: docker-
compose down
URL for the IAM Manager portal: http://localhost:8002
```

Ya podemos arrancar el contenedor de IAM mediante el siguiente comando:

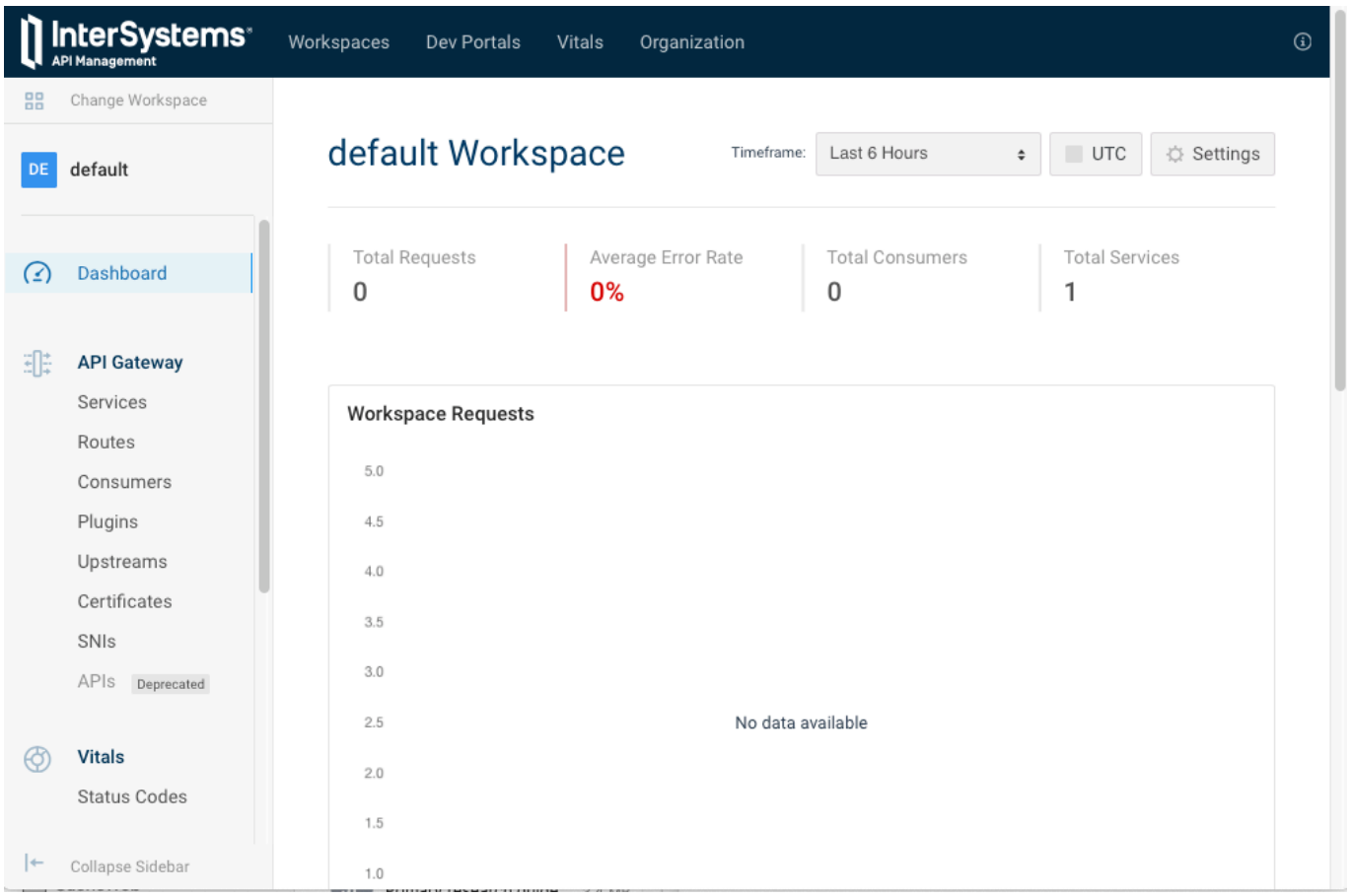
```
docker-compose up -d
```

Opcionalmente, si no usas la opción "-d", podrás ver el registro de acciones del contenedor para ver que todo se inicia correctamente. Si todo ha ido bien, podremos acceder al portal de IAM en la URL http://localhost:8002.



Lo primero que vemos es el informe global que no arroja ninguna información ya que este es un nuevo nodo. Cambiaremos esto en breve.

IAM soporta el concepto de workspace (espacio de trabajo) para separar nuestro trabajo en módulos y/o equipos. Si hacemos scroll hacia abajo podemos seleccionar el workspace por defecto (default) y accederemos al cuadro de mando para este cuadro de mando. Bien, usaremos este workspace para nuestro primeros experimentos.



De nuevo, el número de solicitudes para este workspace sigue siendo cero. Sin embargo, en el menú de la izquierda podemos ver los principales conceptos del API Gateway. Los primeros dos elementos son los más importantes: Servicios y Rutas. Un servicio es una API que queremos exponer a consumidores. Por lo tanto, una API REST en tu instancia de IRIS se considera un Servicio.

Una ruta decide a cuál servicio debe dirigir las solicitudes de entrada. Cada ruta tiene un conjunto de condiciones y si la solicitud cumple con esas condiciones, esta es dirigida al servicio asociado. Por ejemplo, una ruta puede requerir coincidencia con la IP o el dominio del emisor, con métodos HTTP, partes de la URI o una combinación de lo anterior.

Podemos comenzar creando un servicio con los siguientes valores:

campo	valor	descripción
name	test-iris	El nombre que le damos al servicio
host	xxx.xxx.xxx.xxx	Dirección IP pública de nuestra instancia
port	52773	El puerto utilizado para nuestra solicitud HTTP
protocol	http	El protocolo que soporta

Para todos los demás campos, mantendremos los valores por defecto.

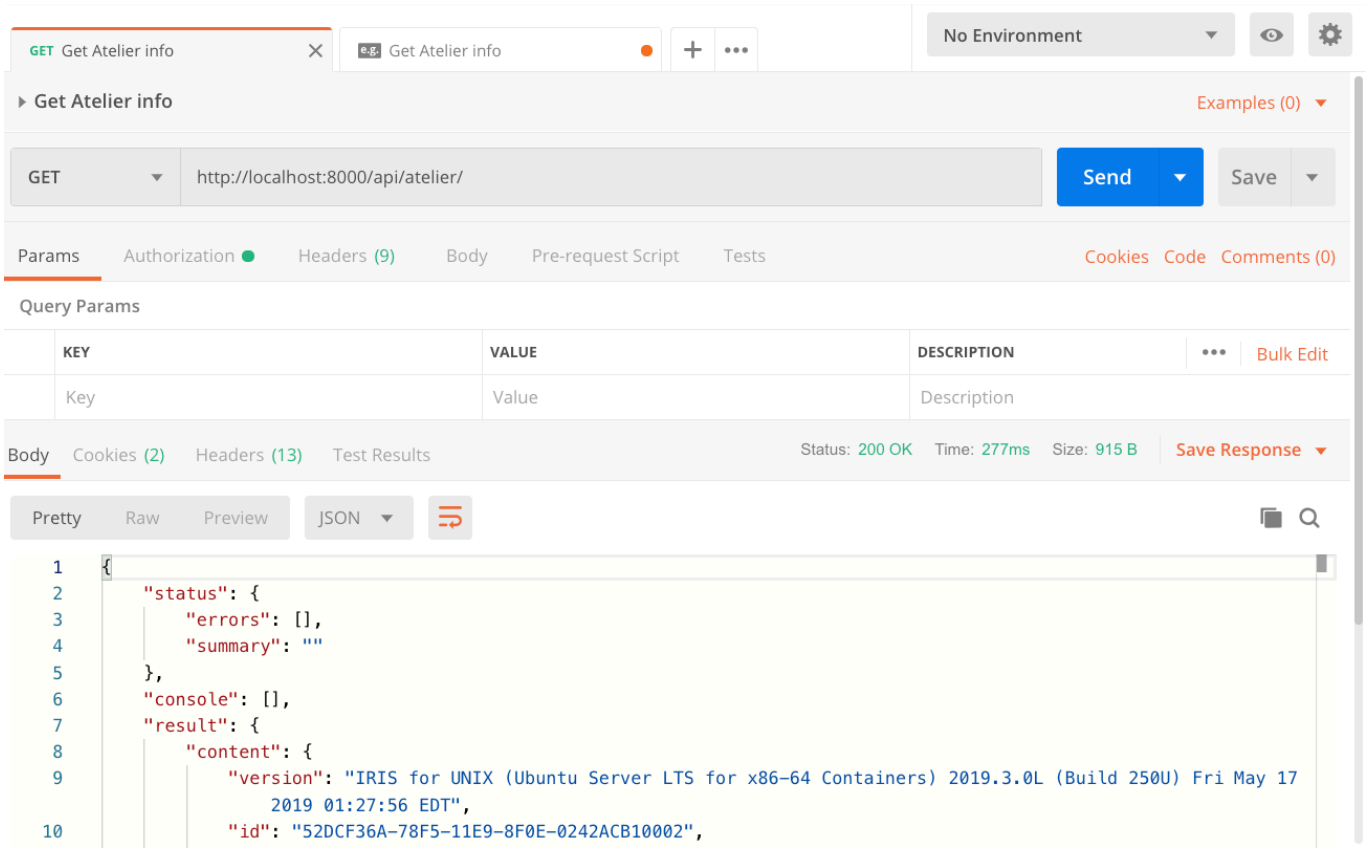
Ahora crearemos una ruta:

campo	valor	descripción
paths	/api/atelier	Las solicitudes con esta ruta serán dirigidas al servicio definido
protocols	http	El protocolo que se soporta
serviceid	xxxx	Id del servicio al que serán dirigidas las solicitudes entrantes

De nuevo, mantén los valores por defecto para todo lo demás.

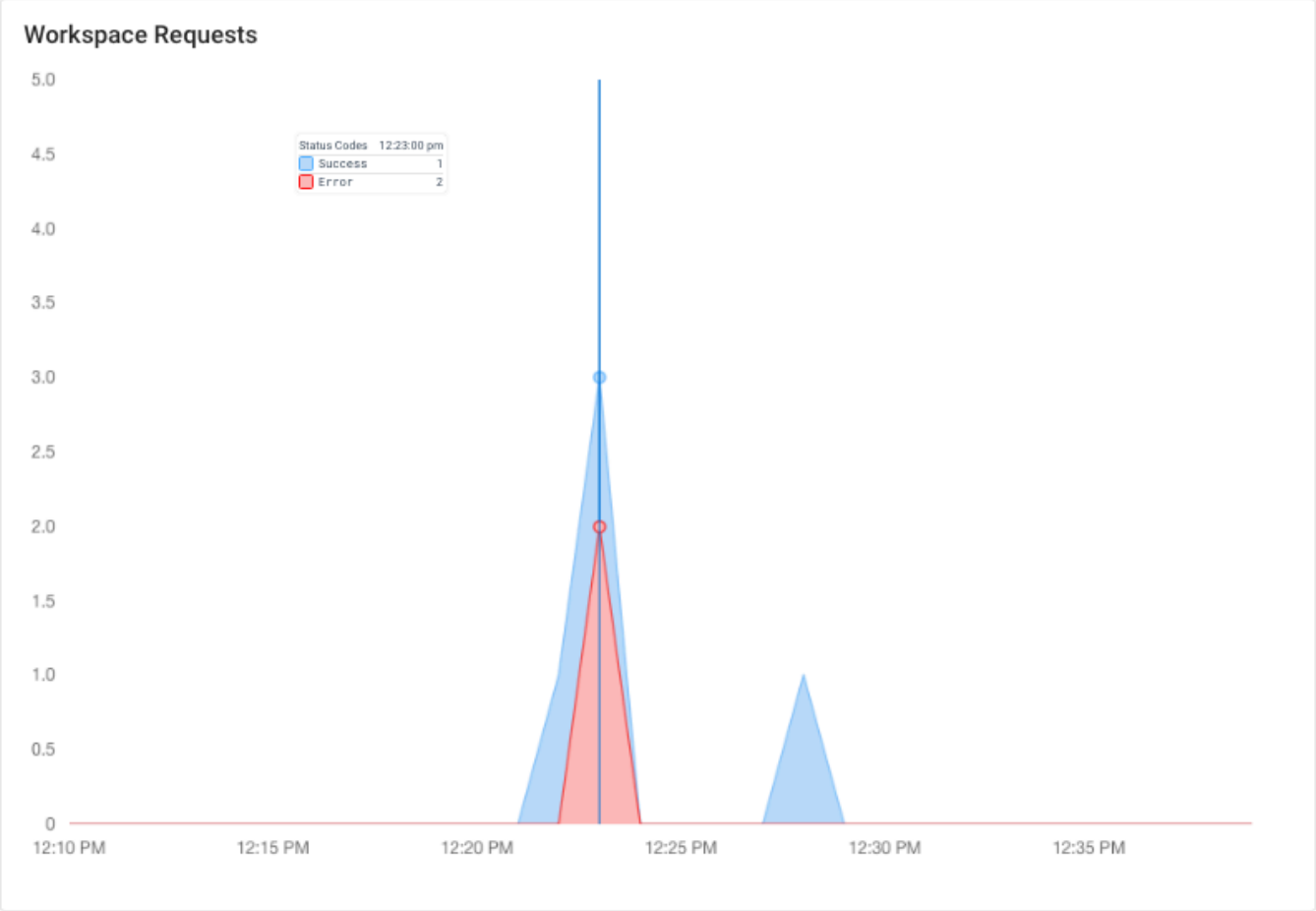
IAM por defecto está escuchando en el puerto 8000 para las solicitudes entrantes por defecto. Desde este momento las solicitudes que se envíen a http://localhost:8000 y comiencen con la ruta */api/atelier* serán dirigidas a nuestra instancia IRIS.

Probemos el funcionamiento con un cliente REST (estoy usando Postman aquí).



Enviamos un HTTP GET a http://localhost:8000/api/atelier/ y efectivamente devuelve una respuesta de nuestra Instancia IRIS. Cada solicitud se procesa a través del IAM y se monitorizan métricas como el código de estado HTTP, la latencia y el consumidor (si se configura).

Realicemos un par de solicitudes más (incluyendo solicitudes donde no existe el endpoint como */api/atelier/test/*) y podremos ver todas ellas agregadas en el cuadro de mando:



Trabajando con complementos (plugins)

Ahora que tenemos una ruta básica definida, podemos comenzar a administrar el tráfico. Además ya podemos comenzar a agregar comportamientos que complementen nuestro servicio. Ahora sucede la magia.

La forma más común de imponer un determinado comportamiento es agregar un complemento. Los complementos aíslan una cierta funcionalidad y, por lo general, se pueden conectar a ciertas partes de IAM. Pueden asociarse a nivel global o solo a partes como a un usuario (grupo), un servicio o una ruta. Comenzaremos agregando un complemento de limitación de volumen a nuestra ruta. Lo que necesitamos para establecer el enlace entre el complemento y la ruta es el identificador único de la ruta. Puede buscarlo viendo los detalles de la ruta.

Copie el identificador que tenga tu ruta para usarlo en el siguiente paso.

Haga clic en Complementos (Plugins) en el menú de la barra lateral izquierda. Por lo general, verá todos los complementos activos en esta pantalla, pero como este nodo es relativamente nuevo, todavía no hay complementos activos. Por lo tanto, continúe seleccionando "Agregar nuevo complemento".

El complemento que buscamos está en la categoría "Control de tráfico" y se llama "Limitación de volumen" (Rate Limiting). Hay bastantes campos que puede definir aquí, ya que los complementos son muy flexibles, pero ahora solo nos interesan dos campos:

campo	valor	descripción
routeid	*****	Identificador de tu ruta
config.minute	5	Número de llamadas permitidas por minuto

Eso es. El complemento está configurado y activo. Probablemente haya visto que podemos elegir entre una variedad de intervalos de tiempo, como minutos, horas o días, pero deliberadamente usé minutos, ya que esto nos permite comprender fácilmente el impacto de este complemento.

Si vuelve a enviar la misma solicitud en Postman, se darás cuenta de que la respuesta vuelve con 2 encabezados adicionales. XRateLimit-Limit-minute (valor 5) y XRateLimit-Remaining-minute (valor 4). Esto le dice al cliente que puede hacer hasta 5 llamadas por minuto y tiene 4 solicitudes más disponibles en el intervalo de tiempo actual.

Body Cookies (2) Headers (15) Test Results Status: 200 OK Time: 289ms

KEY	VALUE
Content-Type	application/json; charset=utf-8
Content-Length	388
Connection	keep-alive
X-RateLimit-Limit-minute	5
X-RateLimit-Remaining-minute	4

Si sigue haciendo la misma solicitud una y otra vez, eventualmente se quedará sin su cuota disponible y en su lugar obtendrá un código de estado HTTP 429 con la siguiente información:

Body Cookies (2) Headers (7) Test Results Status: 429 Too Many Requests Time: 9ms

Pretty Raw Preview JSON

```
1  {
2      "message": "API rate limit exceeded"
3  }
```

Si espera hasta que termine el minuto, podrás pasar solicitudes de nuevo. Este es un mecanismo bastante útil que permite lograr un par de cosas:

1. Proteger su backend de picos de actividad
2. Establecer una expectativa para el cliente sobre cuántas llamadas puede realizar de manera transparente para sus servicios
3. Monetizar (potencialmente) en función del tráfico de la API mediante la introducción de niveles (por ejemplo, 100 llamadas por hora en el nivel de bronce e ilimitado con oro)

Puede establecer valores para diferentes intervalos de tiempo y de este modo suavizar el tráfico API durante un período determinado. Digamos que permites600 llamadas por hora para una ruta determinada. Eso es 10 llamadas por minuto en promedio. Pero no está evitando que los clientes utilicen todas sus 600 llamadas en el primer minuto de su hora. Tal vez eso es lo que quiere. O tal vez le gustaría asegurarse de que la carga se distribuya más equitativamente a lo largo de la hora. Al establecer el campo configminute en 20, se asegura de que los usuarios no realicen más de 20 llamadas por minuto y 600 por hora. Esto permitiría algunos picos en el intervalo de nivel de minutos, ya que solo pueden hacer 10 llamadas por minuto en promedio, pero los usuarios no pueden usar la cuota por hora en un solo minuto. Ahora les tomará al menos 30 minutos si alcanzan su sistema con plena capacidad. Los clientes recibirán encabezados adicionales para cada intervalo de tiempo configurado, por ejemplo:

Cabecera	Valor
X-RateLimit-Limit-hour	600
X-RateLimit-Remaining-hour	595
X-RateLimit-Limit-minute	20
X-RateLimit-Remaining-minute	16

Por supuesto, hay muchas formas diferentes de configurar los límites de uso dependiendo de lo que se quiera lograr.

Me detendré en este punto, ya que probablemente sea suficiente para el primer artículo sobre el InterSystems API Manager. Hay muchas más cosas que puedes hacer con IAM, ¡acabamos de usar solo uno de los más de 40 complementos y aún no hemos usado todos los conceptos básicos!

Aquí hay un par de cosas que puede hacer también y que podríamos cubrir en futuros artículos:

- Agregar un mecanismo de autenticación central para todos sus servicios
- Escalar horizontalmente mediante solicitudes de balanceo de carga a múltiples destinos que admiten el mismo conjunto de APIs
- Presentar nuevas funciones o correcciones de errores a un público más pequeño y monitorizar cómo funciona antes de abrirlo a todos
- Incorporar desarrolladores internos y externos y proporcionarles un portal de desarrolladores dedicado y personalizable que documente todas las API a las que tienen acceso
- Cachear respuestas que se solicitan a menudo para reducir la latencia de respuesta y la carga en los sistemas de servicio

No deje de probar IAM y cuéntenos qué le parece en los comentarios a este artículo. Hemos trabajado duro para ofrecer esta funcionalidad y estamos ansiosos por saber qué desafíos se pueden superar ahora con esta tecnología.

Más recursos

El comunicado de prensa oficial se puede encontrar aquí: InterSystems IRIS Data Platform 2019.2 introduces API Management capabilities

Un pequeño video animado de introducción: What is InterSystems API Manager

Un video de 8 minutos, explicando brevemente algunas de las cuestiones principales: Introducing InterSystems API Manager

La documentación forma parte de la documentación regular de IRIS: InterSystems API Manager Documentation

#API #API REST #InterSystems API Manager (IAM) #SOAP #InterSystems IRIS