

---

Artículo

[Ricardo Paiva](#) · 30 sep, 2021 Lectura de 5 min

## Cómo usar InterSystems API Manager para equilibrar la carga de una API

InterSystems API Manager (IAM) es una nueva función de InterSystems IRIS que permite monitorizar, controlar y dirigir el tráfico hacia y desde las APIs web dentro de su infraestructura de TI. Si te perdiste el anuncio de su lanzamiento, puedes leerlo [aquí](#). También puedes consultar este otro [artículo](#).

En esta publicación, utilizaremos InterSystems API Manager para balancear la carga de una API.

En nuestro caso, tenemos 2 instancias de InterSystems IRIS con /api/atelier API REST que queremos publicar para nuestros clientes.

Hay muchas razones diferentes por las que podríamos querer hacer eso, como:

- Equilibrio de carga para distribuir la carga de trabajo entre los servidores
- Implementación azul-verde: tenemos dos servidores, uno "prod", otro "dev" y es posible que queramos cambiar entre ellos
- Implementación de Canary: podríamos publicar la nueva versión solo en un servidor y mover el 1% de los clientes allí
- Configuración de alta disponibilidad
- etc.

Aun así, los pasos que debemos dar son bastante similares.

### Requisitos previos

- 2 instancias de InterSystems IRIS
- Instancia de InterSystems API Manager

### Vamos

Esto es lo que tenemos que hacer:

#### 1. Crear un upstream.

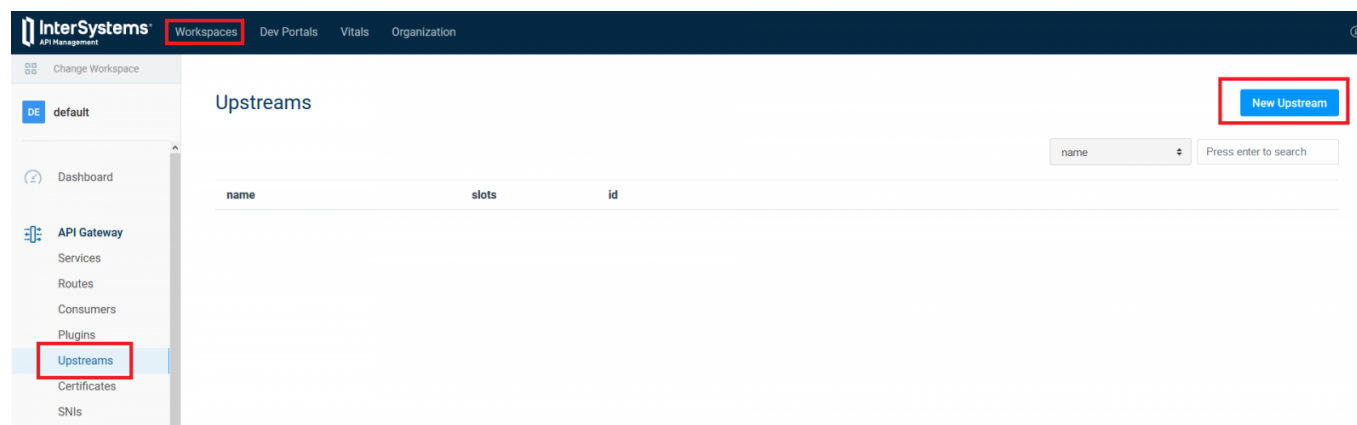
El upstream representa un hostname virtual y se puede utilizar para equilibrar en varios servicios (objetivos) la carga de las solicitudes que entran. Por ejemplo, un upstream llamado service.v1.xyz recibiría solicitudes para un Servicio cuyo servidor es service.v1.xyz. Las solicitudes de este Servicio se dirigirían a los objetivos definidos en el upstream.

Un upstream también incluye un verificador de estado, que puede habilitar y deshabilitar objetivos en función de su capacidad o incapacidad para atender solicitudes.

Para empezar:

- Abre el Portal de Administración de IAM
- Ve a los espacios de trabajo (workspaces)

- Elige un espacio de trabajo
- Abre Upstreams
- Haz clic en el botón "New Upstream"



Después de hacer clic en el botón "New Upstream", verás un formulario en el que podrás introducir información básica sobre el upstream (hay muchas más propiedades):

Introduce name - es un hostname virtual que nuestros servicios podrían utilizar. No está relacionado con los historiales del DNS. Recomiendo establecerlo en un valor que no exista para evitar confusiones. Si quieres leer sobre el resto de propiedades, [puedes consultar la documentación](#). En la captura de pantalla, puedes ver cómo nombré al nuevo upstream como myupstream.

## 2. Crear objetivos.

Los objetivos son servidores backend que ejecutan las solicitudes y envían los resultados al cliente. Ve a Upstreams y haz clic en el nombre del upstream que acabas de crear (y NO en el botón de actualización):

Verás todos los objetivos existentes (ninguno hasta ahora) y el botón "New Target". Haz clic:

Y en el nuevo formulario define un objetivo. Solo hay dos parámetros disponibles:

- target - host y puerto del servidor backend
- weight - prioridad relativa que se le da a este servidor (a más weight- más solicitudes se envían a este objetivo)

Yo he añadido dos objetivos:

## 3. Crear un servicio.

Ahora que tenemos nuestro upstream, necesitamos enviarle solicitudes. Para ello, utilizamos un Servicio.

Las entidades de Servicio, como su nombre indica, son abstracciones de cada uno de tus servicios de upstream. Ejemplos de Servicios serían un microservicio de transformación de datos, una API de facturación, etc.

Vamos a crear un servicio dirigido a nuestra instancia de IRIS. Ve a Servicios y haz clic en el botón "New Service":

Establece los siguiente valores:

campo	valor	descripción
nombre	myservice	el nombre lógico de este servicio
host	myupstream	nombre del upstream
path	/api/atelier	ruta raíz que queremos proporcionar
protocol	http	los protocolos que queremos admitir

Mantén los valores predeterminados para todo lo demás (incluyendo el puerto: 80).

Después de crear el servicio, lo verás en una lista de servicios. Copia el ID del servicio en algún sitio, vamos a necesitar eso más tarde.

#### 4. Crear una ruta.

Las rutas definen las reglas para hacer coincidir las solicitudes de los clientes. Cada Ruta está asociada a un Servicio, y un Servicio puede tener varias Rutas asociadas con él. Cada solicitud que coincida con una Ruta determinada se redirigirá a su Servicio asociado.

La combinación de Rutas y Servicios (y la separación de problemas entre ellos) ofrece un potente mecanismo de enrutamiento con el que es posible definir puntos de entrada detallados en IAM que conducen a diferentes servicios upstream de tu infraestructura.

Ahora vamos a crear una ruta. Ve a Rutas y haz clic en el botón "New Route".

Establece los valores en el formulario de creación de rutas:

campo	valor	descripción
path	/api/atelier	ruta raíz que queremos proporcionar
protocol	http	los protocolos que queremos admitir
service.id	guid from 3	valor id del servicio (guid del paso anterior)

¡Y terminamos!

Envía una solicitud a <http://localhost:8000/api/atelier/> (con una barra al final) y será presentada por uno de nuestros dos backends.

## Conclusión

IAM ofrece una infraestructura de administración de API altamente personalizable, lo que permite que los desarrolladores y administradores tomen el control de sus API.

## Enlaces

- [Documentación](#)
- [Anuncio de IAM](#)
- [Presentación de IAM](#)

## Pregunta

¿Qué función quieres ver configurada con IAM?

[#API](#) [#API REST](#) [#InterSystems API Manager \(IAM\)](#) [#SOAP](#) [#InterSystems IRIS](#)

---

URL de  
fuente: <https://es.community.intersystems.com/post/c%C3%B3mo-usar-intersystems-api-manager-para-equilibrar-la-carga-de-una-api>