

# Incident Response Platform Integrations

# Rate Limit V1.0.0

Release Date: Mayo 2021

## **Descripción**

Esta integración permite controlar mediante funciones utilizadas dentro de Workflows, si se están ejecutando determinados flujos en una cantidad no deseada por periodo de tiempo.

Así, por ejemplo, se pueden proteger distintos recursos contra posibles denegaciones de servicio o ejecución descontrolada de automatizaciones.

Los recursos a proteger se configuran en una tabla de base de datos MySQL, para asegurar persistencia fuera de Resilient. Como parámetros de configuración se debe ingresar para cada recurso, la cantidad aceptables de requests por periodo de tiempo.

Cada vez que se invoca la función, se registra un nuevo request, que es persistido en la base de datos y es evaluado globalmente en toda la plataforma independientemente del Incidente que lo genero.

Algunas características:

* Función instalada en resilient circuits.
* Persistencia en base de datos MySQL, tanto de configuración de recursos protegidos como de invocaciones.
* Registro histórico de invocaciones.
* Cantidad ilimitada para agregar recursos a proteger.
* Posibilidad de habilitar o deshabilitar recursos globalmente, sin necesidad de modificar los Workflows donde son evaluados. Si un recurso esta deshabilitado en la base de datos, la función devuelve rate limit no superado.
* Si se invoca un recurso inexistente o deshabilitado, o si la base de datos no está disponible, la función responde como rate limit no superado.

## **Instalación y configuración**

En el servidor donde se encuentra resilient circuits:

Instalar el paquete:

$ pip install --upgrade fn\_rate\_limit-x.x.x.tar.gz

Importar las configuraciones al app.config:

$ resilient-circuits config -u -l fn-rate-limit

Importar los objetos de la aplicación a Resilient:

$ resilient-circuits customize -y -l fn-rate-limit

Configurar los parámetros de la aplicación para acceder a la base de datos:

$ vi ~/.resilient/app.config

Config Required Example

db\_user Yes ratelimit

db\_password Yes PASSWORD

db\_host Yes 127.0.0.1

db\_port Yes 3306

database Yes Rate\_limit

Guardar y Cerrar app.config.

Iniciar o restartear el servicio resilient-circuits:

$ resilient-circuits run

## **Preparación de la base de datos**

Loguearse al motor de base de datos.

En el archivo de query “**DB\_Rate\_Limit\_definition.sql**”

Editar los parámetros según corresponda:

* Password de usuario de la Base de Datos
* Periodo de tiempo de mantenimiento de la tabla Data, y parámetro de retención de datos históricos.

Ejecutar el contenido del query.

Una vez finalizado, se habrán creado:

* Usuario de base de datos con permiso de acceso local y las tablas **Data** y **Settings**.
* Base de Datos **Rate\_limit**, y tablas **Data** y **Settings**.
* Evento periódico de mantenimiento de datos históricos de tabla **Data**.

## **Descripción de funciones**

**Rate Limit: Add resource request**

Esta función registra un nuevo request del tipo indicado en el parámetro de input “rate\_limit\_resource”, y verifica si se superó la cantidad máxima de requests permitida por tiempo, según lo configurado en Settings.

Si se superó, devuelve True en el atributo “rate\_limit\_exceeded”, caso contrario devuelve False.

**Inputs:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Example** |
| rate\_limit\_resource | String | Yes | deshabilitar\_usuario\_ad |
| rate\_limit\_event\_data | String | No | Registrado en Incidente id {} |

**Output:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| results.success | Boolean | True si la operacion fue exitosa.  False en caso de producirse algun error capturable, por ejemplo:  Resource no existe en Settings, Resource existe pero esta deshabilitado, Base de Datos no disponible. |
| results.rate\_limit\_exceeded | Boolean | True si con el request enviado se supero rate limit configurado para el resource.  False si no se supero rate limit. |
| results.resource | String | Nombre del resource. |

## **Administración de resources para controlar rate limit**

Los recursos que se necesita controlar por rate limit, se encuentran listados en la tabla **Settings**.

Para agregar, quitar o modificar los mismos, es necesario modificar dicha tabla con los siguientes valores:

* **resource**: Nombre de recurso a proteger. Este es el nombre que se debe incluir en el parámetro “**rate\_limit\_resource**” de la Función “**Rate Limit: Add resource request**” al incluirla en los Workflows.
* **rate\_limit\_number\_requests**: Cantidad máxima de requests permitidas en el periodo de tiempo, antes de determinar que se superó umbral de rate limit.
* **rate\_limit\_time\_period**: Tiempo en segundos en el que se permitirán la cantidad seteada de requests antes de determinar que se superó umbral de rate limit.
* **Description**: Descripción del resource. Texto libre.
* **enabled**: Indicador de resource habilitado (1) o deshabilitado (0). Si esta seteado en 0, cualquier llamada a este resource devolverá:

'success': False, 'rate\_limit\_exceeded': False.

* **date\_created**: Fecha de creación del resource.