

**UST Global Policies Datapower- v1.0**

**Prueba de Concepto**

# Control de edición.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Actualización** | **Preparado por:** | **Revisado por:** | **Fecha** | **Cambios** |
| 15/03/2018 | Jose de la Gala | Maria Bejarano | 15/03/2018 | 1.0 Inicial |
| 19/03/2018 | Jose de la Gala | Maria Bejarano | 19/03/2018 | 1.1 Política Validación |
| 21/03/2018 | Jose de la Gala | Maria Bejarano | 19/03/2018 | 1.2 Despliegue |
| 22/03/2018 | Jose de la Gala | Maria Bejarano | 12/03/2018 | 1.3 Desencriptar |

# Índice de Contenidos.

1 Control de edición. 3

2 Índice de Contenidos. 4

3 Objetivo 6

*3.1* Destinatarios 6

4 Introducción 7

*4.1* Empaquetado y estructura 7

*4.2* Instalación y acceso 9

5 Politicas implementadas 11

*5.1* Validación de los datos de entrada 11

*5.2* Encriptación de la carga útil 13

6 Evolutivos propuestos 18

7 Solución Problemas 19

*7.1* Error al desplegar paramreq.zip y encriptar.zip 19

7.1.1 Problema 19

7.1.2 Soluciones 20

7.1.3 Notas 25

Aviso de Confidencialidad y Copyright

Copyright © 2017 by UST Global Inc. Este documento está protegido por las leyes de copyright como obra inédita y contiene información que no se puede reproducir, publicar, utilizar para la preparación de trabajos derivados, y / o distribuir, en su totalidad o en parte, por el destinatario, para cualquier otro propósito que no sea la evaluación de este documento. Asimismo, toda la información contenida en este documento es confidencial y propiedad de UST Global Inc y no podrá ser revelada a terceros; el uso de la marca UST Global en el presente documento ha sido otorgado por UST Global Inc, con domicilio social en Aliso Viejo, California, EE.UU. Las excepciones a esta notificación se podrán producir únicamente con autorización expresa y por escrito de UST Global Inc.

UST Global® es una marca registrada de UST Global Inc.

UST Global ®

UST Global España, S.A.

Calle Fernández Caro nº7

28027 - Madrid (España)

[www.ust-global.com](http://www.ust-global.com/)

# Objetivo

Se trata de realizar una descripción detallada de las políticas Datapower para el proyecto API Management MX. Se detallan características de su construcción, su valor añadido y el modo en el que se deberían testear.

## Destinatarios

ISBAN MX. Equipo de API Management.

# Introducción

Las políticas Datapower son extensiones del producto IBM API Connect que sirven para añadir valor a las lógicas de ensamblado en la fase de ejecución de los entregables.

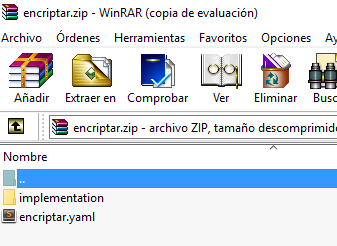
Existen políticas de caja tales como Proxy (para hacer invocaciones a sistemas externos), Map (para hacer correlaciones entre distintos esquemas), Activity-Log (para obtener nuevos campos en las analíticas), y un largo etcétera.

Sin embargo, IBM API Connect es un producto con sus puntos de mejora que deben ser cubiertos en base a las necesidades de los clientes. En este sentido, las políticas Datapower pueden aportar valor, pues pueden cubrir dicho gap. Ya sea en aspectos técnicos, tales como firmas con certificados, por ejemplo, o funcionales, tales como validaciones semánticas de esquemas, etc. A continuación, se describirán dichas funcionalidades.

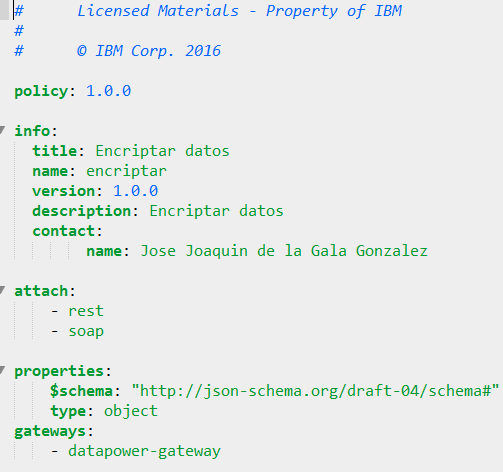
En Datapower son posibles dos tipos de políticas, las basadas en ECMAScript® 5.1 (Javascript) y las basadas en transformaciones XSLT. En el presente documento nos ocuparemos de implementaciones basadas en Javascript.

## Empaquetado y estructura

Las políticas se empaquetan en ficheros ZIP con la siguiente estructura:

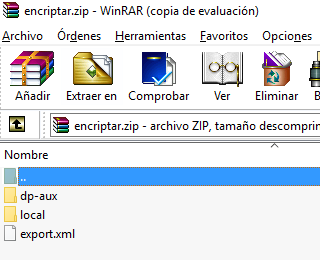


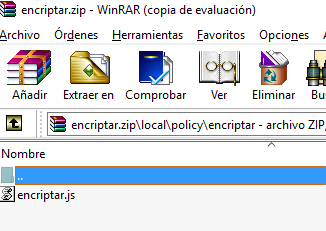
El fichero YAML (encriptar.yaml) describe las características de la implementación contenida en la carpeta al mismo nivel. El contenido de la descripción es la siguiente:



Donde se indican el nombre / título / descripción y también los campos de entrada en el caso de ser necesarios.

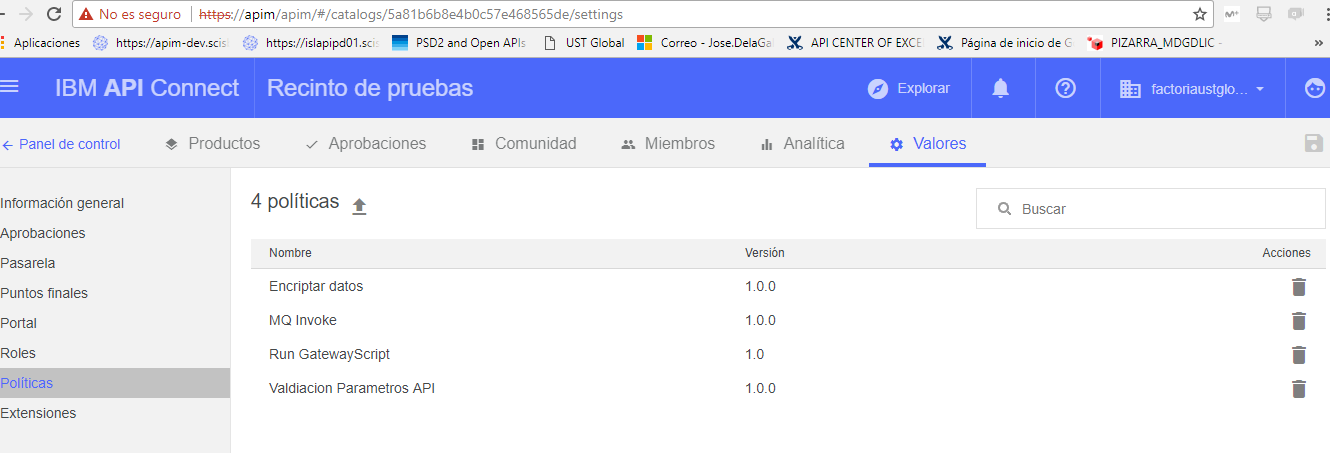
La carpeta implementación contiene otro ZIP del mismo nombre que el anterior con el código fuente, destacando el fichero JavaScript ubicado en el último nivel de la estructura de carpetas.





## Instalación y acceso

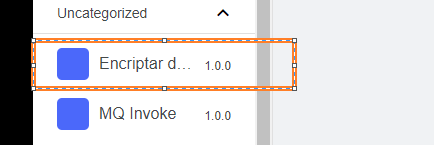
Para instalar una política en una instancia de IBM API Connect es necesario tener en cuenta que dicha instalación se realiza a nivel de catálogo. Para ello en Panel de Control 🡪sección valores del Catálogo🡪 Políticas como se ve en la imagen, están instaladas todas las políticas personalizadas.

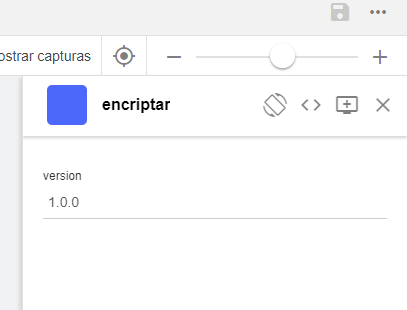
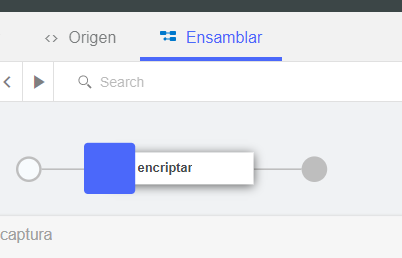


Para instalar una política nueva y correctamente empaquetada se hace mediante las siguientes opciones:



De ese modo, la política es accesible desde el Manager de IBM API Connect. Como se ve en las imágenes:





# Politicas implementadas

## Validación de los datos de entrada

La presente política nace de la necesidad de validar los datos de entrada documentados mediante Swagger y que, sin embargo, en tiempo de ejecución no se controlan de caja. En este caso, la definición del API en tiempo de diseño únicamente es válida a efectos de documentación.

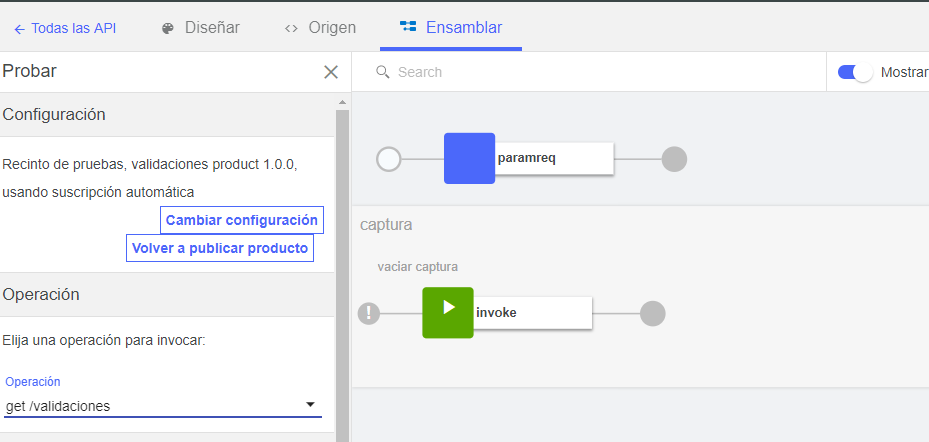
Para ello, se ha construido una política que valida los parámetros de entrada de un API, guarda todos los campos que son requeridos sin informar en la petición JSON de entrada y lanza una excepción con el código “400 Bad Request” e información del incumplimiento con la documentación.

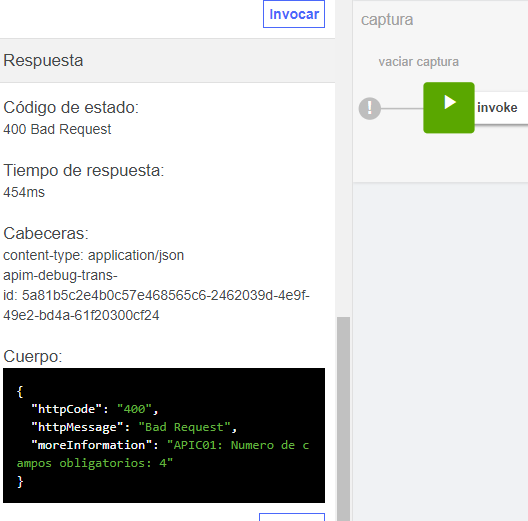
Veamos un ejemplo:

Tenemos un API con la política de validación en el ensamblaje, hemos definido dos campos específicos y dos genéricos, los cuatro requeridos.



Lanzamos la petición con el tester integrado en el Manager, y responde el siguiente resultado:





En el log de Datapower vemos las siguientes trazas:

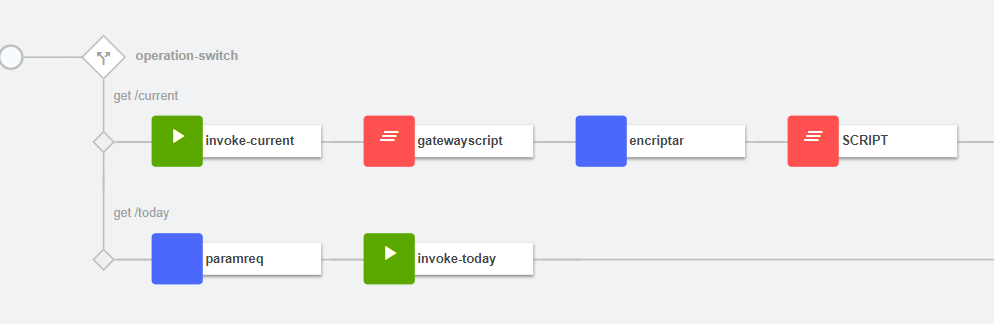


## Encriptación de la carga útil

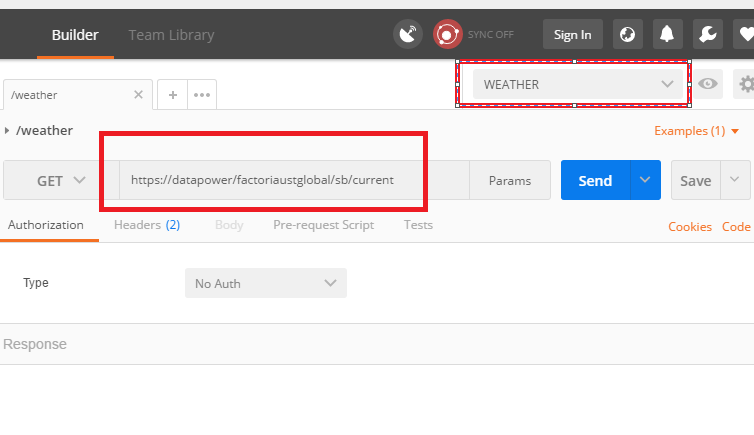
Con el objetivo de ocultar datos sensibles se ha creado una primera versión de política de encriptación del payload. Junto con la política de des-encriptación todavía por construir, serviría para aportar más seguridad a las comunicaciones entre API´s.

El funcionamiento es el siguiente. Se captura el cuerpo del mensaje y se codifica en base64. A continuación dicho valor se añade a una nueva cabecera de nombre “Message-Body” y finalmente se resetea el “body” que inicialmente procesaba la API. Veamos un ejemplo:

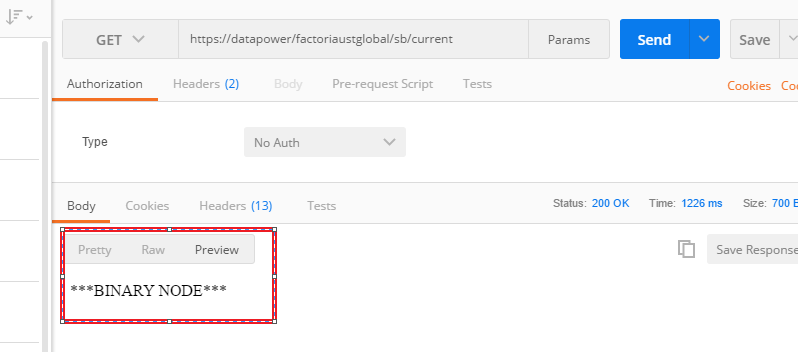
Tenemos un API que da información del tiempo con una caja “encriptar” por la rama “current”:

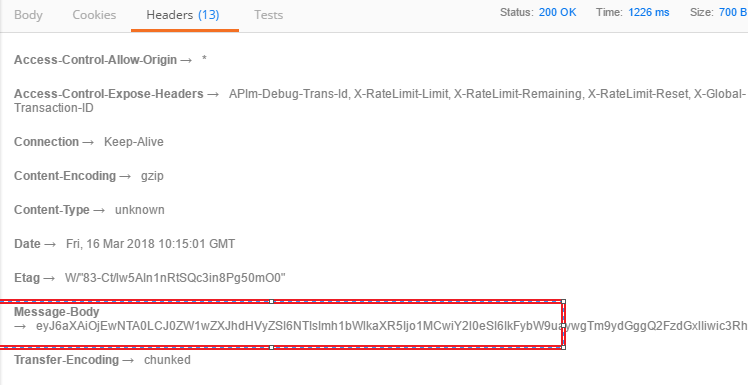


Invocamos el API desde un Postman, una vez tenemos una aplicación de prueba suscrita al producto que contiene dicha API / Plan.



Siendo la respuesta siguiente:

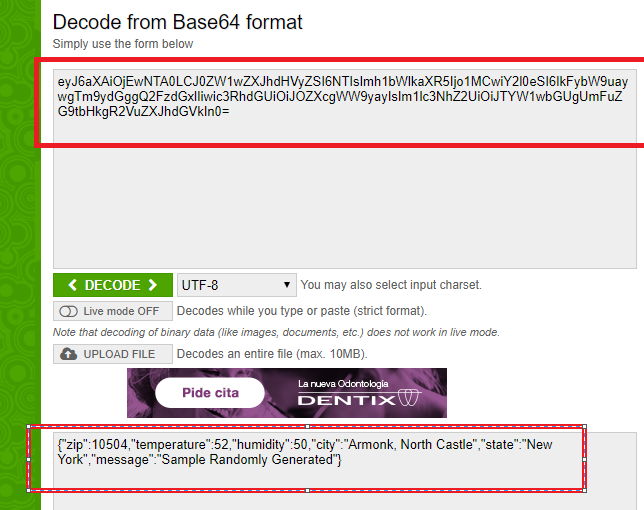




El valor de Message-Body es:

eyJ6aXAiOjEwNTA0LCJ0ZW1wZXJhdHVyZSI6NTIsImh1bWlkaXR5Ijo1MCwiY2l0eSI6IkFybW9uaywgTm9ydGggQ2FzdGxlIiwic3RhdGUiOiJOZXcgWW9yayIsIm1lc3NhZ2UiOiJTYW1wbGUgUmFuZG9tbHkgR2VuZXJhdGVkIn0=

Si dicho valor los decodificamos en una web para dicho fin (pe. <https://www.base64decode.org/>):



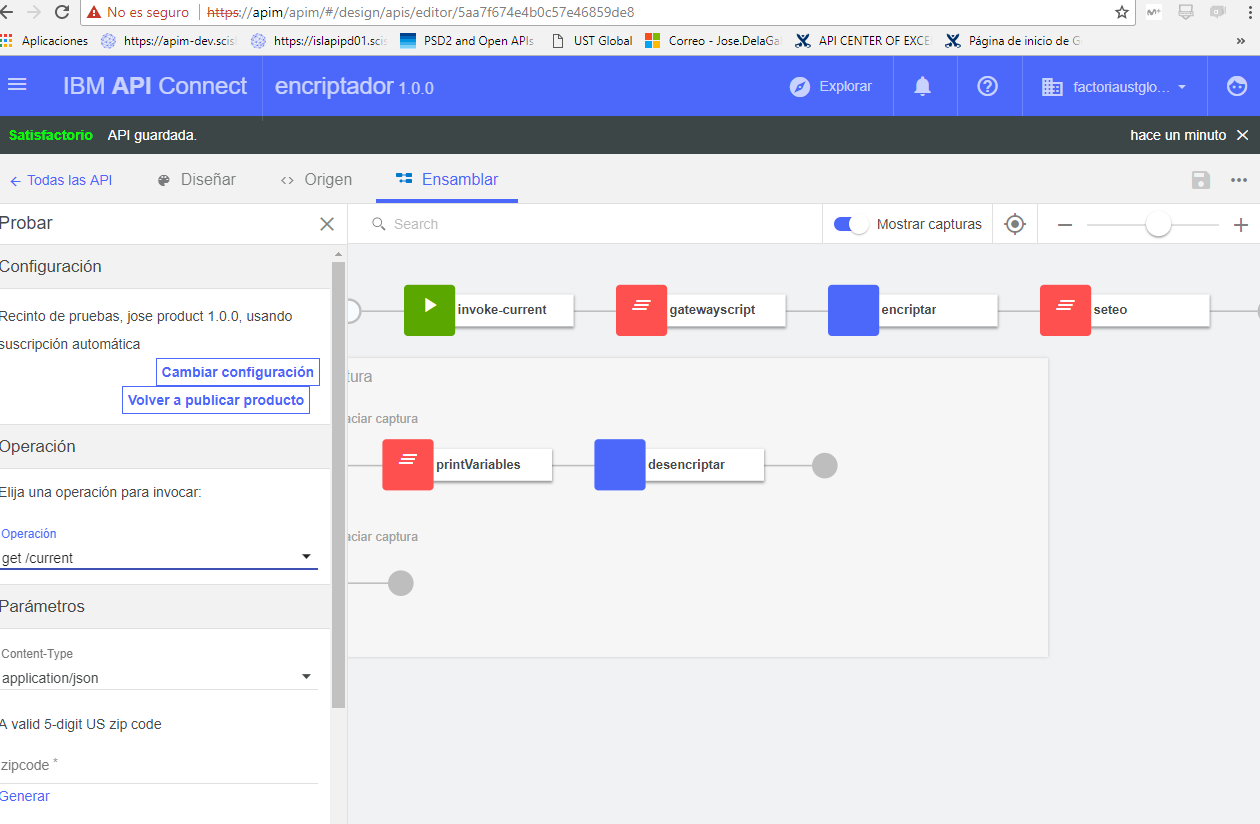
Es importante tener en cuenta que se recoge el valor “message.body” en contexto durante el pipeline del API. Se refiere a aquellos valores que se setean mediante la siguiente función Datapower:

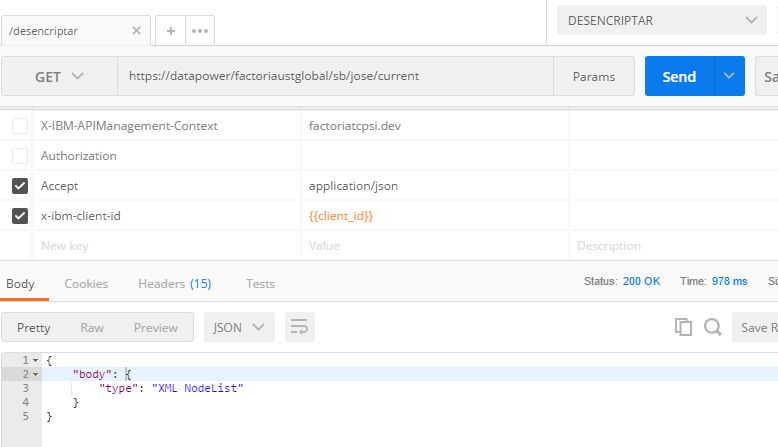


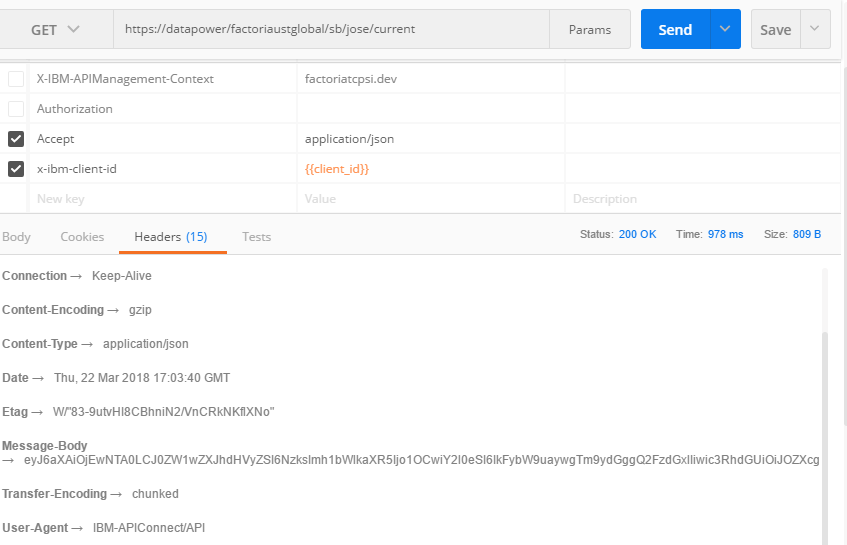
## Encriptación / Des encriptación de la carga útil

Siguiendo el ejemplo del apartado anterior, hemos preparado una política que realiza el proceso contrario que la anterior. Primero lee de la cabecera “Message-Body” con el valor encriptado en base64, lo actualiza en la carga útil de la petición en curso y finalmente borra la cabecera donde se guardaba la información.

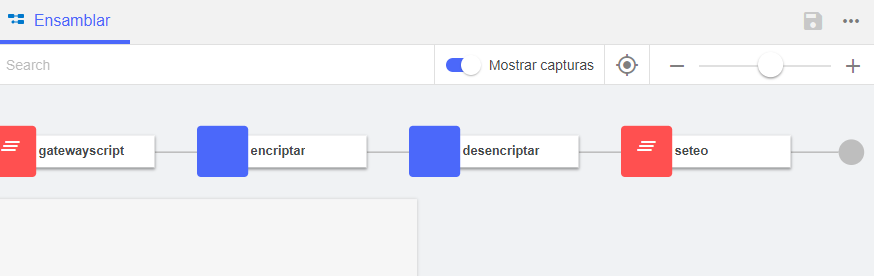
Para ilustrar dicho proceso, hemos preparado un ejemplo. Primero realizamos una petición con la política de encriptar únicamente. Y vemos la respuesta del Postman:

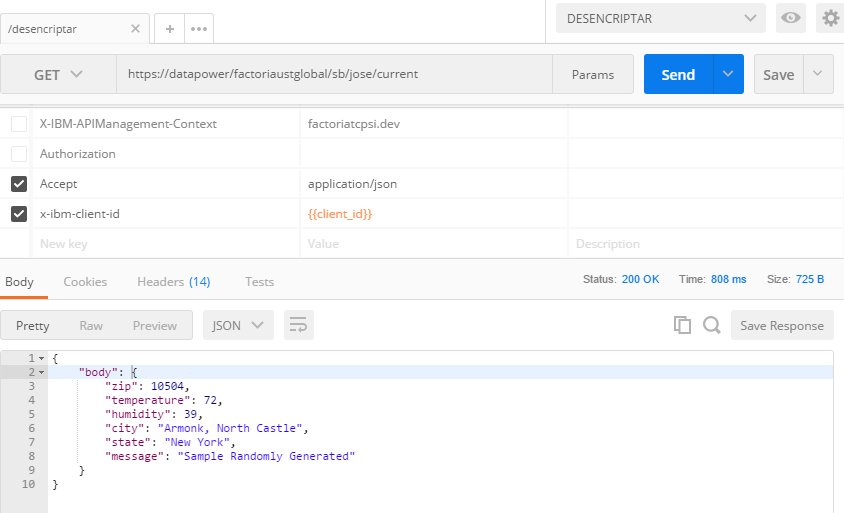


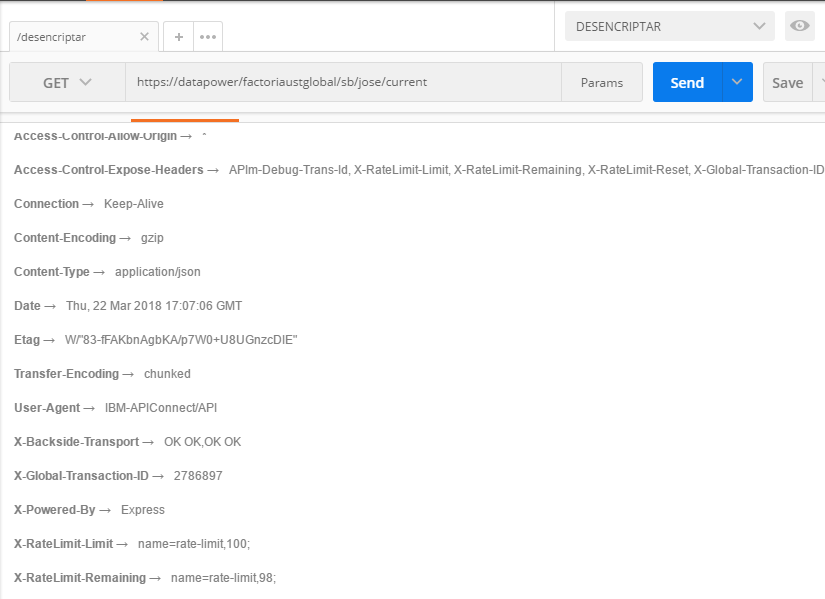




A continuación, colocamos la política de des encriptación para ver el resultado:







# Evolutivos propuestos

Desde UST Global vemos interesante seguir ampliando las funcionalidades de los apartados 4.1, 4.2. O más políticas que puedan interesar.

En el caso de la validación semántica de campos de entrada vemos las siguientes posibles funcionalidades / mejoras, por ejemplo:

* Validación del tipado de las peticiones.
* Validación de campos del “body” en peticiones POST.

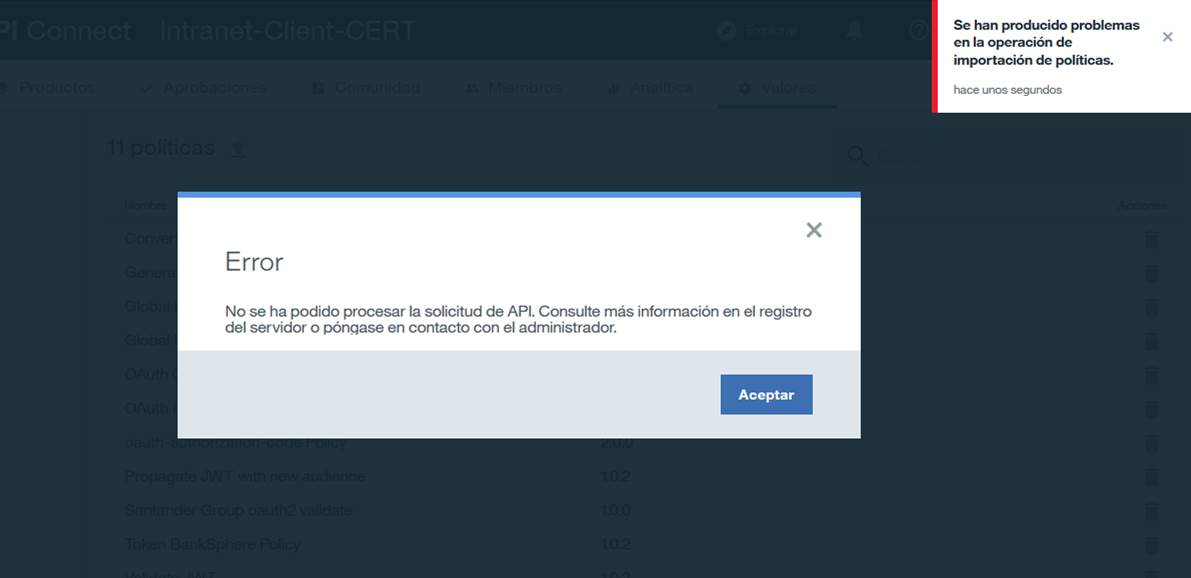
También se podría completar el tema de le des encriptación de la política 4.2, así como el uso de un cifrado más seguro en siguientes versiones.

# Solución Problemas

## Error al desplegar paramreq.zip y encriptar.zip

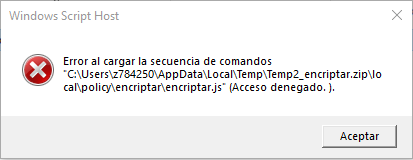
## Problema

En ambos casos arroja este error al desplegarlo en un catálogo dado:



En el caso de la política “parameq” no encuentra el archivo .js en el último nivel del zip.

En el caso de la política “encriptar” sí se encuentra el archivo al fondo del ZIP pero no permite abrirlo, devolviendo el siguiente error:



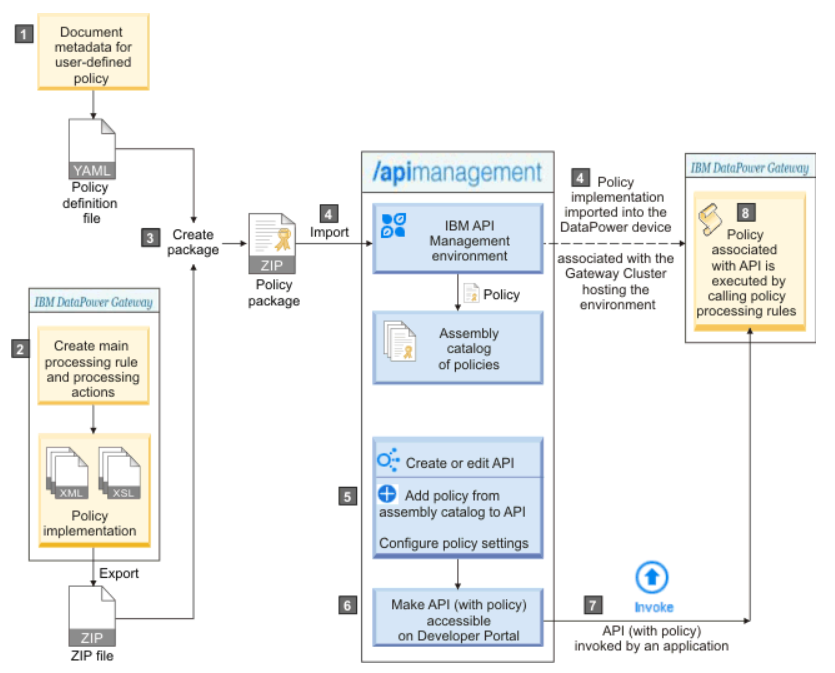
## Soluciones

En estos casos, las causas principales de los errores de despliegue podrían deberse a la implantación personalizada de las políticas de Produban.

Es decir, en entornos configurados de caja o por defecto, con importar el ZIP sería suficiente para que todo se configure de modo automático. Mientras, que en entornos de Produban son necesarios algunos pasos manuales, relacionados con el entorno específico donde queremos desplegar nuestra política. Que a continuación explicamos.

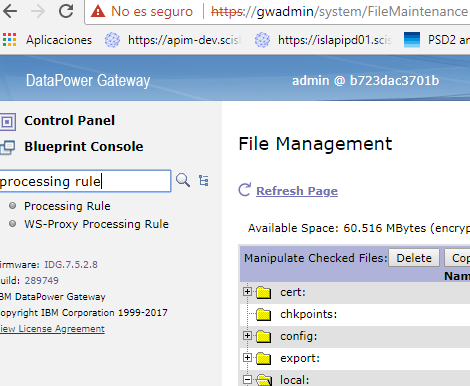
Tenemos dos casos, cuando la política ya está desplegado o cuando se despliegan por primera vez. En nuestro caso, vamos a explicar cómo se configura por primera vez y si finaliza el despliegue con éxito el ZIP de la implementación podría ser distribuido para evitar errores de despliegue en el futuro.

El flujo para nuevos despliegues es:

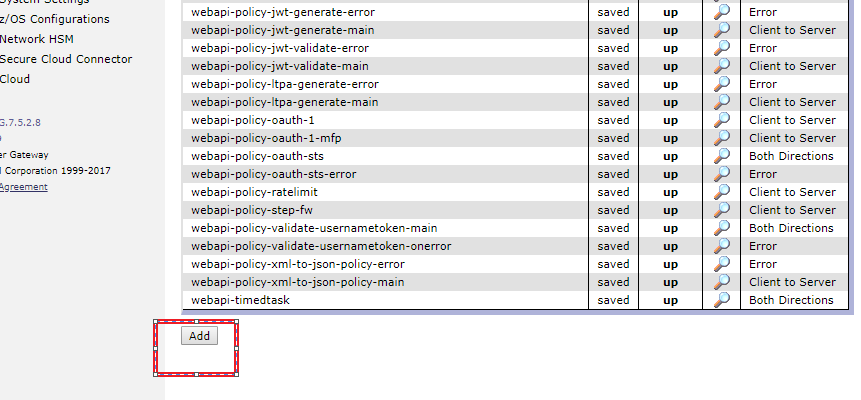


En la práctica los pasos a seguir son los siguientes:

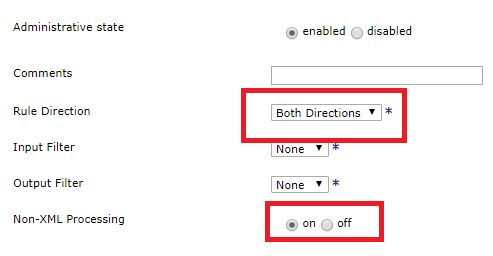
1. Acceder a Datapower donde se quieren desplegar las políticas. Buscar por “processing rule”



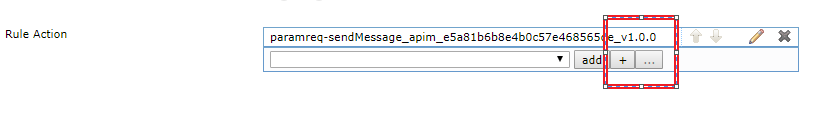
1. Añadir una nueva entrada.



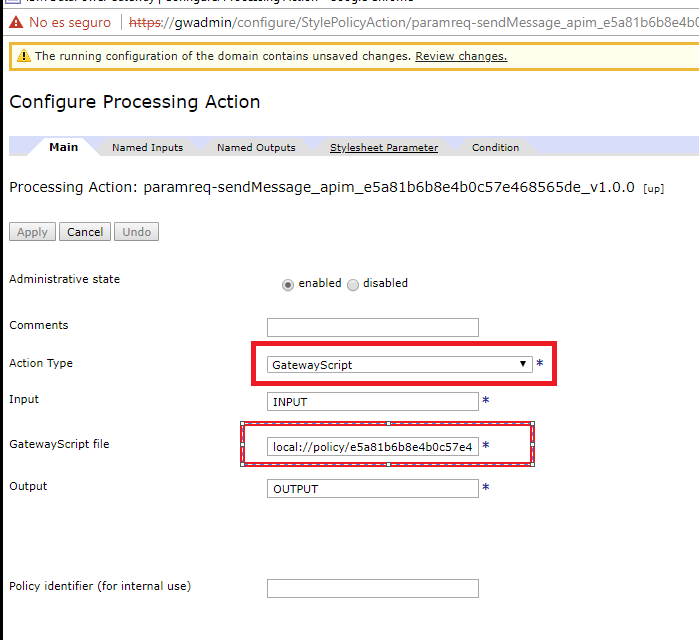
1. El nombre debe ser <nombrePolicy>-main. En nuestro caso los nombres de políticas son “paramreq” y “encriptar”. Por ejemplo, ponemos el nombre “paramreq-main” y continuamos el ejemplo con dicha política. “Rule Directions” a “Both Directions” y “Non-XML Proccessing” a “On”:



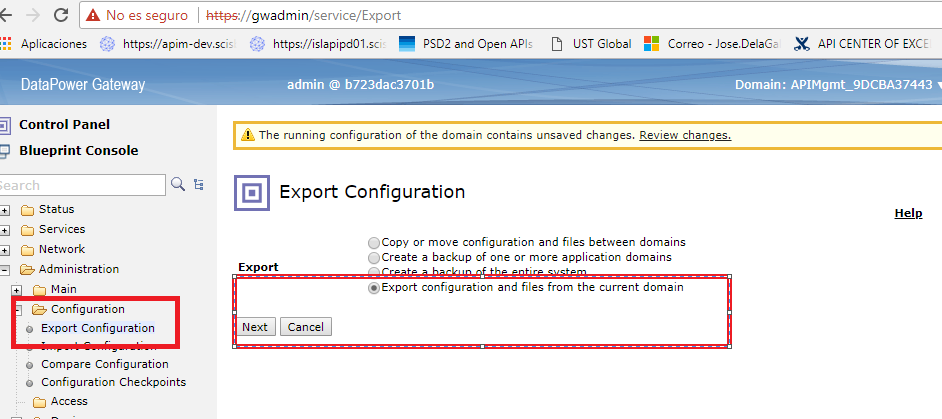
1. Ahora añadimos tantas entradas “Rule Action” como en Catálogos destinos queramos desplegar. Añadimos una entrada con el botón “Add”



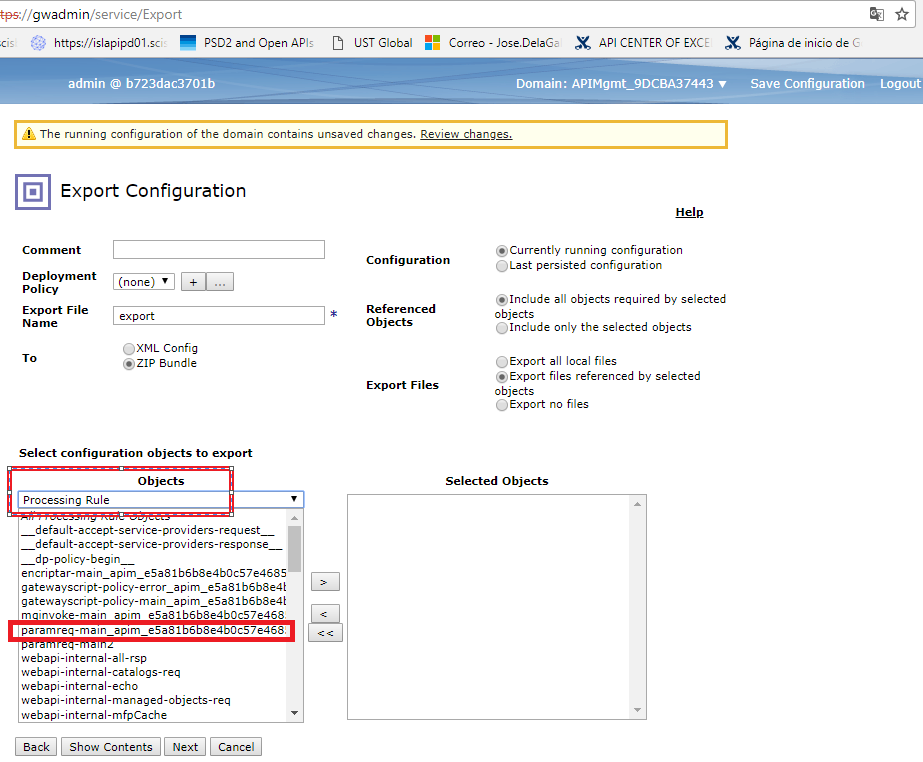
1. Configuramos “Action Type” a “GatewayScript”, y los valores “INPUT/OUTPUT” junto con la ubicación relativa en el File System de Datapower del fichero JavaScript. Es muy importante tener en cuenta que la ubicación relativa suele ser del estilo “local://policy/e5a81b6b8e4b0c57e468565de/paramreq\_v1.0.0/paramreq.js” donde el identificados “e5a81b6b8e4b0c57e468565de” se corresponde con la organización propietaria del catálogo, donde queremos desplegar. Por tanto, si queremos desplegar en digamos 3 organizaciones / catálogos diferentes dichas políticas que se corresponden con una única instancia de Datapower tendríamos aquí 3 entradas. De nombre podemos configurar “local”, o según la nomenclatura de Produban, si para ello tuviese procedimiento.



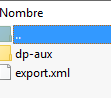
1. Después de aplicar todos los cambios a la configuración deben quedar todos los parámetros de configuración “Up” y pasamos a exportar la implementación de la política. Vamos a Configuración 🡪Export



1. Se seleccionan de entre todas las “Proccessing Rules” la nuestra, y se genera el ZIP con la implementación de nuestra política “parameq”. Dicho ZIP es el que incluye dentro de la carpeta “implementation” junto con el YAML de definición de dicha política.



Comentar que el ZIP descargado contiene los siguientes ficheros:

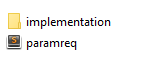


Falta la carpeta “local” con la parametrización de “export.xml” que añadimos manualmente. Se trata en realizada del desarrollo JavaScript de la política.

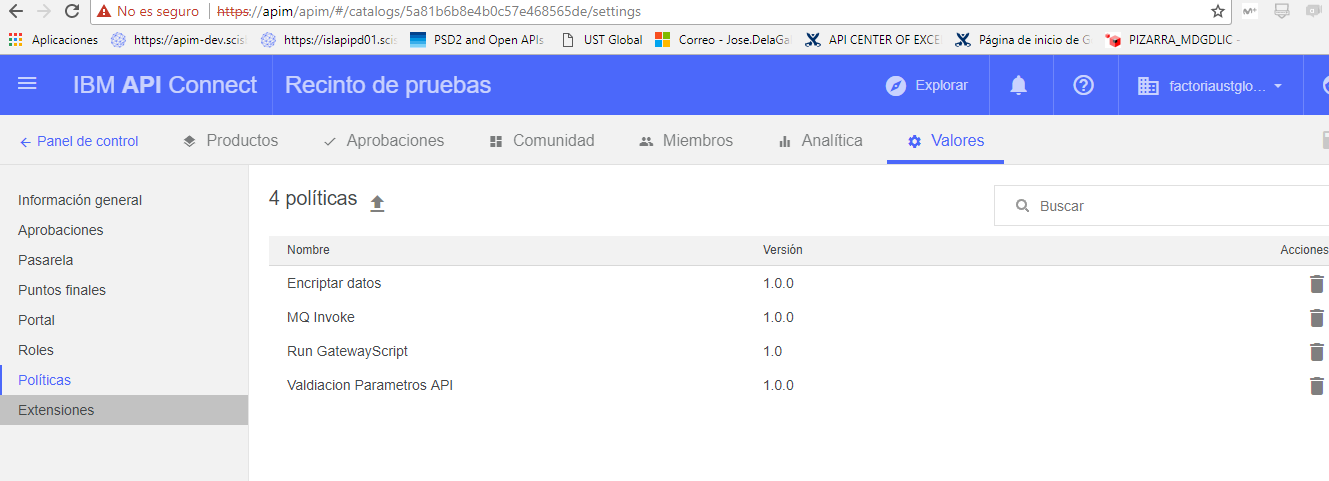
En este punto la tenemos un ZIP que contiene la carpeta “implementation” más el YAML. Y dicha carpeta contiene el ZIP de implementación con la siguiente estructura como vimos antes:



Siendo el ZIP padre así:



Después de configurar dichos parámetros ya se podría incluir en catálogo, previamente parametrizado, el ZIP con la política. Siempre que el usuario tenga permisos de administración sobre el catálogo. La importación debería funcionar pues la parte de implementación se ha adaptado previamente en Datapower. Es necesario re empaquetar en otro ZIP, la carpeta “implementation” más el YAML.



Una vez que se ha logrado desplegar con éxito, el fichero ZIP se podría distribuir a las personas que construyen las políticas para no tener que realizar la configuración descrita cada vez que se despliega un nuevo ZIP. De modo que el despliegue funciona a la primera.

## Notas

El ZIP que te nos descargamos del Datapower tiene la configuración en el fichero export.xml. En dicho ZIP falta la carpeta “local” a la que apunta al fichero JS, y parametrizada en export.xml. Dicha ruta parametrizada tiene que contenedir el ID de la organización propietaria del catálogo. Pe. Local🡪policy🡪e5a81b6b8e4b0c57e468565de🡪paramreq\_v1.0.0🡪paramreq.js

La otra duda, era relativa al YAML de definición. Debe estar contenido en el ZIP con el que se distribuye la politica, el cual contiene la carpeta “implementation” con el ZIP de antes y el YAML de definicion.

También hay que tener en cuenta que podemos crear varias acciones de procesamiento por cada organización / catálogo implicados, si fuese necesario.

Por tanto, el único objetivo de la configuración / descarga del primer ZIP de implementación es que al desplegar la politica con el ZIP padre, las rutas correspondan con las configuradas.

Todo este proceso de costrucción / despliegue entiendo que se podría automatizar con algún framework tipo maven, etc… para que el proceso sea menos manual.

En el siguiente video, vemos un ejemplo de despliegue manual: <https://1drv.ms/f/s!Amv3cD5ztmbgibxp1aFMYJmjTikVWA>

Como resumen:

* El proceso es manual tal como lo planteamos.
* Tenemos dos ZIP uno el de la propia politica y el otro dentro de la carpeta “implementation”. Al mismo nivel está el YAML.
* El ZIP de implementation contiene la descarga de Datapower más la carpeta “local” que costruimos en base a la parametrización de export.xml. Fichero que describe la configuración inicial Datapower.