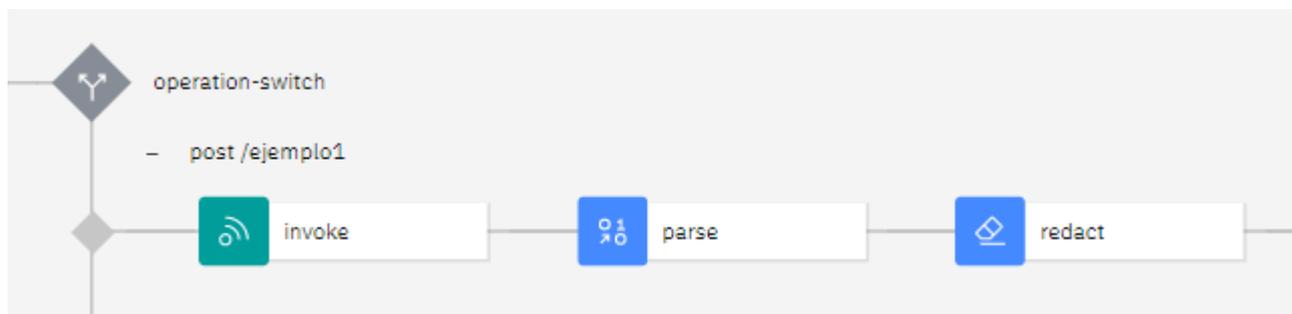


## Flujo



## Policy Invoke

URL: <https://reqres.in/api/users/2>

**invoke** [edit] [share] [close]

Title  
invoke

Description

URL \*  
The URL to be invoked.  
https://reqres.in/api/users/2

Backend type \*  
JSON

HTTP Method \*  
The HTTP method to use for the invocation. If omitted or set to "Keep", then the method from the incoming request will be used.

GET

## Policy Parse

1
parse
↶ ↷ ✕

---

Title

parse

---

Description

---

Use Content Type

Specifies whether to attempt a parse with the specified content-type. This is only applicable if the expected content-type is either json or xml. With this option enabled, a parse action will result in failure if content-type does not match the parse-settings (i.e Content-type:application/json and parse-settings is configured for xml). If the parse-setting is configured to detect input content, it will be forced to the specified content-type.

Parse settings object reference

Specifies an existing valid object from which to retrieve default property values for the dynamic object. Warning: if you change from the default string, the parse setting object specified must be available on the DataPower appliance.

Default

Reset

Parse settings literal configuration

Specifies a literal string as serialized XML or JSON properties that are merging into the dynamic object. These properties take precedence over any existing default properties.

---

Parse settings URL reference

Specifies a URL or string that represents a named context from which to retrieve the serialized XML or JSON properties that are merging into the dynamic object. These properties take precedence over any existing literal or default properties.

Message for parsing

Specifies the name of a variable in the API context. The content of the "body" field of the variable is the input to the action. The default variable name is "message".

---

Message for resulting parsed data

Specifies the name of a variable in the API context. The content of the "body" field of the variable is the output of the parse. The parse metrics of the parsed document can be stored in a different part of the message. The default variable name is whatever the Input name is, so by default the Input message will be overwritten with the Output message.

## Policy Redact

- Action 1
  - Path: message.body.data.email
  - Action: Redact
- Action 2
  - Path: message.body.data.avatar
  - Action: Remove

1
redact
↶ ↷ ✕

---

Title

redact

---

Description

Root

Specifies the data source that contains the content to redact or remove. If the root is not specified, the action is applied to the entire API context. You can use any supported JSONata path expression. Redaction of request and response data may be performed with a root of "message.body". If the assembly redact action is used in an assembly after an assembly log action that specifies gather-only mode, the root can specify "log.request\_body" for the logged request payload or "log.response\_body" for the logged response payload.

---

Path \*

Use JSONata to specify the property to redact or remove

message.body.data.email

---

Action \*

Select whether to remove the property completely, or to redact the field by using "\*"s to block out the data.

Redact

Path \*

Use JSONata to specify the property to redact or remove

message.body.data.avatar

---

Action \*

Select whether to remove the property completely, or to redact the field by using "\*"s to block out the data.

Remove

## Código YAML (Redact)

```
- redact:
  version: 2.0.0
  title: redact
  redactions:
    - action: redact
      path: message.body.data.email
    - action: remove
      path: message.body.data.avatar
```

## Postman sin Redact

```
1  {}
2  "data": {
3    "id": 2,
4    "email": "janet.weaver@reqres.in",
5    "first_name": "Janet",
6    "last_name": "Weaver",
7    "avatar": "https://reqres.in/img/faces/2-image.jpg"
8  },
9  "support": {
10   "url": "https://reqres.in/#support-heading",
11   "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
12 }
13 {}
```

## Postman con Redact



The screenshot shows the Postman interface with the 'Body' tab selected. The response is displayed in 'Pretty' format as JSON. The status is '200 OK'. The JSON structure is identical to the one shown in the 'Postman sin Redact' section, but the 'email' field is redacted to '\*\*\*\*\*' and the 'avatar' field is missing.

```
1  {}
2  "data": {
3    "id": 2,
4    "email": "*****",
5    "first_name": "Janet",
6    "last_name": "Weaver"
7  },
8  "support": {
9    "url": "https://reqres.in/#support-heading",
10   "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
11 }
12 {}
```

---

## **ArquitecturaIBM Consulting**

