

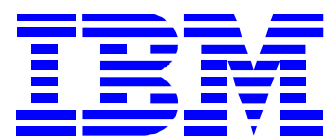
## **IBM API Connect v5.0**

### **API 어셈블(조립/조합) 2**

(2016. 06. )

IBM Hybrid Cloud Technical Sales

이정운 차장(juwlee@kr.ibm.com)

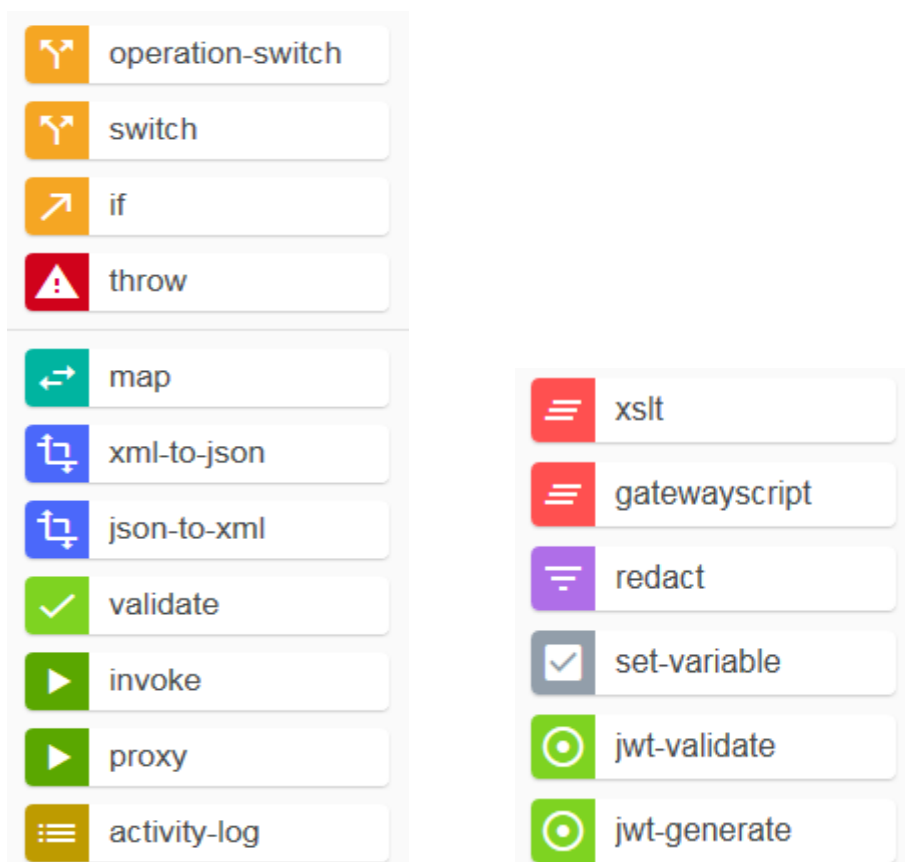


## 0) API 어셈블(조립/조합) 2 에서는?

안녕하세요 이정운 입니다.

지난 강좌에서 API 어셈블에 대해서 살펴봤는데 테스트 해보시면서 느끼셨겠지만 API 어셈블을 위한 디자이너의 왼쪽 팔레트에는 이전 강좌에서 사용된 컴포넌트 이외에 미리 만들어진 다양한 컴포넌트가 추가로 많다는 것을 확인하셨을 것 입니다.

- 현재 버전(APIC v5.0.1) 의 API 어셈블 디자이너에서 제공하는 컴포넌트 리스트 -



기본 정책 - IBM API Connect 5.0 온라인 메뉴얼

[https://www.ibm.com/support/knowledgecenter/ko/SSMNED\\_5.0.0/com.ibm.apic.toolkit.doc/rapim\\_ref\\_oob\\_policies.html](https://www.ibm.com/support/knowledgecenter/ko/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/rapim_ref_oob_policies.html)

이번 강좌에서는 지난번에 사용된 컴포넌트를 제외하고 간단하게 자주 사용되고 쓸 만한 컴포넌트 몇가지를 추가로 설명 및 테스트해볼 수 있는 시간을 가지도록 하겠습니다.

그럼 백문이 불여일타! API 어셈블2 에 대한 강좌를 진행하도록 하겠습니다.

## 1) if 컴포넌트와 gatewayscript 컴포넌트

If 컴포넌트는 말그대로 if 문의 역할을 수행하는 컴포넌트로 '조건' 을 입력할 수 있고 그 조건이 true 인 경우 내부 flow 를 순서대로 수행하게끔 설정할 수 있습니다.

if | if
📌 <> +

**제목**  
if

---

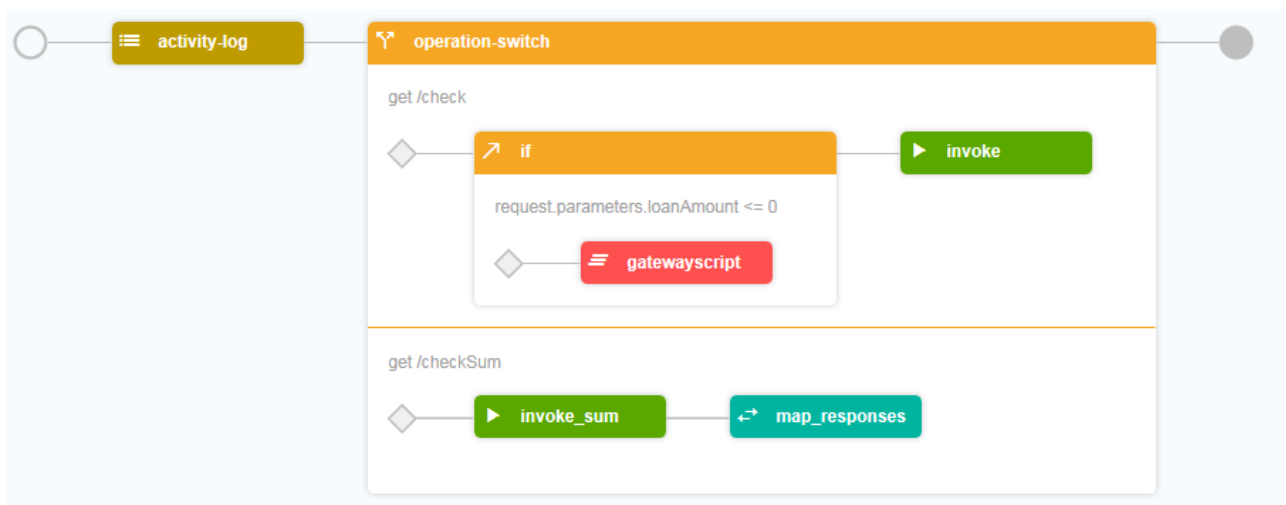
**설명**

---

**조건**

1 request.parameters.loanAmount <= 0






예를 들어 위와 같은 조건("request.parameters.loanAmount <= 0")이라면 들어오는 요청(request) 의 파라미터(parameters) 중에 loanAmount 라는 파라미터의 값이 0 보다 작으면 true 이고 해당 flow 를 수행하게 됩니다. 이때 수행을 위해서 하단과 같이 gatewayscript 를 수행하도록 할 수 있습니다.



gatewayscript 컴포넌트는 손쉽게 javascript 를 수행할 수 있는 컴포넌트로 보시면 됩니다. 이 안에는 다양한 javascript 를 작성할 수 있으며 IBM API Connect 를 위해 미리 정의된 apim.custom.js 를 바로 불러서 사용할 수도 있습니다. 간단하게 위의 if 컴포넌트와 엮어서 파라미터에서 0 보다 작은 값이 왔을 때 Error 를 발생하는 하나의 예제를 작성해보면 하단과 같습니다.

```
// Require API Connect Functions
var apic = require('local:///isp/policy/apim.custom.js');

// Throw Error
apic.error('Parameter Error', 500, 'Parameter Error', 'Parameter have to be grater than 0');
```

 gatewayscript | gatewayscript    

Title  
gatewayscript

Description

```
1 // Require API Connect Functions
2 var apic = require('local:///isp/policy/apim.custom.js');
3
4 // Throw Error
5 apic.error('Parameter Error', 500, 'Parameter Error', 'Parameter have to be
  grater than 0');
```

Response

Status code:  
500 Parameter Error

Response time:  
154ms

Headers:  
apim-debug-  
trans-id: 5756b4ebe4b0f54983fcc7e4-ffb76260-  
a771-435c-abe6-357651d57df1  
content-type: application/json

Body:  

```
{
  "httpCode": "500",
  "httpMessage": "Parameter Error",
  "moreInformation": "Parameter have to be
grater than 0"
}
```

Debug

이외에도 다양하게 javascript 코드를 작성하여 커스텀 동작을 수행하게 할 수 있습니다.

참고 : javascript 에 주석이나 내용으로 한글을 입력하고자 할 경우에는 해당 컴포넌트에서 직접 타이핑 하시는 것보다는 notepad 와 같은 text editro 에서 한글입력 후 Ctrl+C, Ctrl+V 로 붙여넣기 하시면 깔끔하게 한글을 입력할 수 있습니다.

**GatewayScript code examples**

[https://www.ibm.com/support/knowledgecenter/en/SSMNED\\_5.0.0/com.ibm.apic.toolkit.doc/rapim\\_gwscript\\_codesnip.html](https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/rapim_gwscript_codesnip.html)

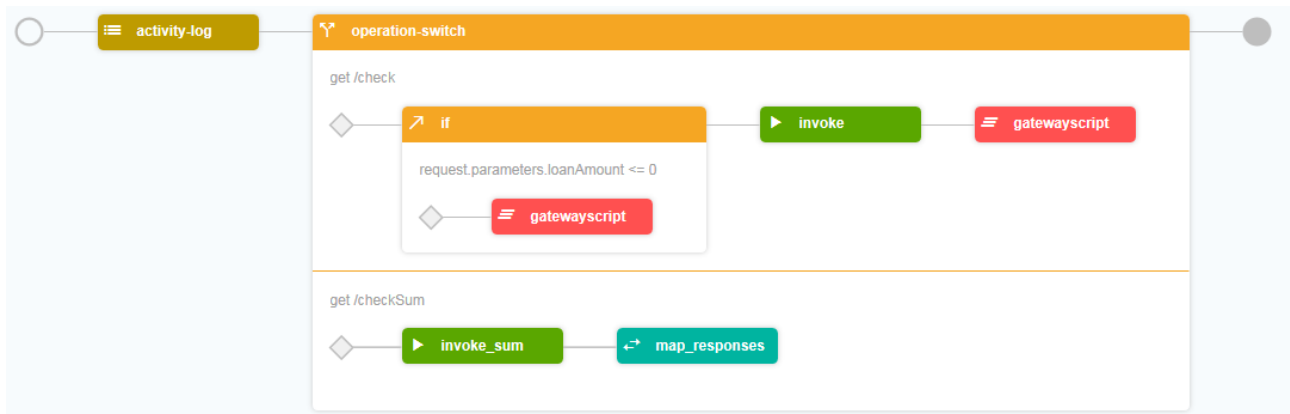
**Implementation code examples**

[https://www.ibm.com/support/knowledgecenter/en/SSMNED\\_5.0.0/com.ibm.apic.policy.doc/rapim\\_imp\\_codesnip.html](https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.policy.doc/rapim_imp_codesnip.html)

## 2) gatewayscript 컴포넌트 2

이전 파트에서 if 컴포넌트와 gatewayscript 컴포넌트를 사용해서 간단하게 해당 파라미터가 0보다 작으면 Error 를 발생하도록 테스트 해봤는데 조금 더 깊게 들어가서 gatewayscript 컴포넌트를 사용하여 응답 값을 변경하는 작업을 수행해보도록 하겠습니다.

지금까지 만들어진 flow 에 invoke 수행 후에 응답을 변경하기 위해서 gatewayscript 컴포넌트를 하단과 같이 추가합니다.



응답을 수정하기 위해서 기존에 있는 invoke 컴포넌트에 'Response object variable' 에 응답으로 사용할 변수명을 입력합니다.

▶ invoke | invoke

GET

The HTTP method to use for the invocation. If omitted, then the method from the incoming request will be used.

☐ Compression

Used to enable/disable Content-Encoding compression on upload.

Cache Type

Protocol

The cache type determines whether to cache documents, honoring or overriding the HTTP Cache Control directives received from the response of the Server. This property takes effect only when a response is received from the Server, otherwise, the policy always returns the non-expired response that was previously saved in cache.

Time to Live

900

Sets the validity period in seconds for documents in the cache. Applies only if the Cache Type is set to 'Time to Live'. Enter a value in the range 5 - 31708800. The default value is 900.

Cache Key

The string to use as a key for the cache. If omitted, the entire URL string is used as the key.

☐ Stop on error

Defines whether the flow stops when a particular error is thrown during the policy execution. Errors not specified here will trigger a catch flow.

Response object variable

invokeRes02

The name of a variable that will be used to store the response data from the request. This can then be referenced in other actions, such as 'Map'.

이제 간단하게 invoke 컴포넌트를 통한 호출 후에 응답을 받아서 monthlyPaymentAmount 의 값만 소수점 이상으로 반올림(javascript 의 Math.round() 함수 사용) 하도록 javascript 를 작성해보면 하단과 같습니다.

```

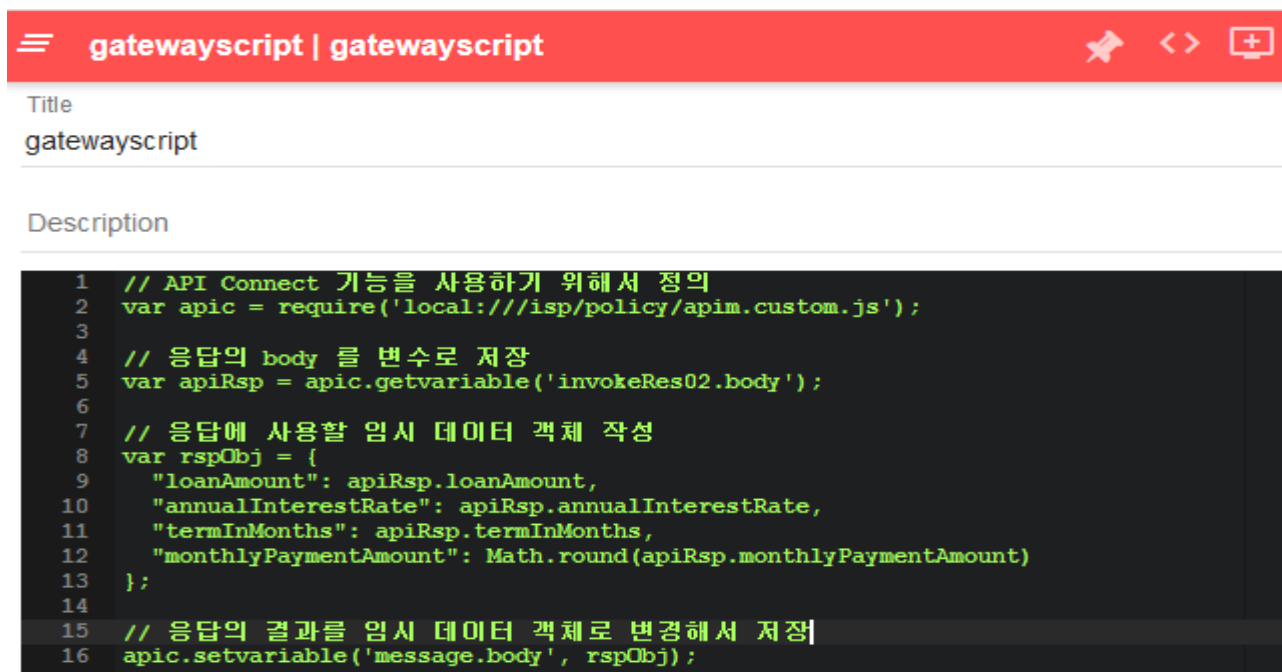
// API Connect 기능을 사용하기 위해서 정의
var apic = require('local:///isp/policy/apim.custom.js');

// 응답의 body 를 변수로 저장
var apiRsp = apic.getvariable('invokeRes02.body');

// 응답에 사용할 임시 데이터 객체 작성
var rspObj = {
    "loanAmount": apiRsp.loanAmount,
    "annualInterestRate": apiRsp.annualInterestRate,
    "termInMonths": apiRsp.termInMonths,
    "monthlyPaymentAmount": Math.round(apiRsp.monthlyPaymentAmount)
};

// 응답의 결과를 임시 데이터 객체로 변경해서 저장
apic.setvariable('message.body', rspObj);

```



이제 저장을 수행하고 테스트를 하게 되면 하단과 같이 monthlyPaymentAmount 결과를 소수점에서 반올림하여 정수로 나오는 것을 확인할 수 있습니다.

### Response

#### Status code:

200 OK

#### Response time:

414ms

#### Headers:

content-type: unknown

x-global-transaction-id: 2549200

x-ratelimit-limit: 100

x-ratelimit-remaining: 99

apim-debug-

trans-id: 5756b4ebe4b0f54983fcc7e4-d9d9efbc-  
7774-468e-bfc0-abc846cea8f8

#### Body:

```
{  
  "loanAmount": 10000,  
  "annualInterestRate": 0.1,  
  "termInMonths": 10,  
  "monthlyPaymentAmount": 1000  
}
```



### 3) validate 컴포넌트

validate 컴포넌트는 이름 그대로 스키마를 통해서 해당 값에 대한 검증을 수행할 수 있는 컴포넌트입니다. 예를 들어 웹 서비스에는 wsdl 이라는 스키마가 있어서 요청(request) 가 들어올 때 해당 요청이 정상적인지 악의적인 수정이 없었는지에 대한 기본 검증이 가능합니다. 이와 마찬가지로 name:value 쌍인 json 에 대해서도 기업에서 사용하기 위해 Swagger 표준을 이용하여 스키마를 지정할 수 있고 그 스키마를 통해서 요청의 body 에 대한 json 데이터를 사전 검증할 수 있습니다. (당연히 문제가 있는 요청 자체를 legacy 로 보내지 않고 안정적으로 보호하는데 그 목적이 있습니다.)

좀 더 보강해서 예를 들면 요청의 body 에 loanAmount 라는 값은 대출 금액이라 양의 정수의 값이 들어와야 합니다. 이를 어떤 사용자들이 or 악의적이거나 잘못된 요청에 의해서 음수로 들어오는 것을 json 형태라고 해도 Swagger 2.0 표준을 이용해서 스키마를 정의하고 사전에 검증할 수 있습니다.

APIC 관리콘솔에 해당 API 로 들어가서 하단과 같이 요청에 대한 정의를 작성합니다. (이전 강좌에서 파라미터로 받는 3개의 변수를 json 요청의 body 에 들어온다고 가정하고 정의를 만든 것입니다.)

#### 정의

+

**requestLoan**
<>

이름  
requestLoan

유형  
object

---

설명

---

특성 특성 추가

* 특성 이름	설명	유형	Example	조치
<input checked="" type="checkbox"/> loanAmount		integer ▼	10000	<>
<input checked="" type="checkbox"/> annualInterestRate		float ▼	0.1	<>
<input checked="" type="checkbox"/> termInMonths		integer ▼	52	<>

☒ 추가 특성 허용

추가 특성 허용을 on 한 후에 loanAmount 라는 특성의 조치의 <> 아이콘을 클릭하면 Swagger 2.0 표준에서 정의된 yaml 형태의 텍스트를 확인할 수 있습니다. 해당 정의에 양의 정수를 검증하기 위해 minimum:0 을 하단과 같이 추가합니다.

```

200 - application/json
201 definitions:
202   requestLoan:
203     properties:
204       loanAmount:
205         type: integer
206         format: int32
207         example: 10000
208         minimum: 0
209       annualInterestRate:
210         type: number
211         format: float
212         example: 0.1
213       termInMonths:
214         type: integer
215         format: int32
216         example: 52
217     required:
218       - loanAmount
219       - annualInterestRate
220       - termInMonths
221     additionalProperties: true
  
```

참고 : Swagger 표준을 확인해보면 스키마의 정의에 넣을 수 있는 minimum, maximum 과 같은 다양한 제한 조건을 하단과 같이 확인할 수 있습니다.

<http://swagger.io/specification/>

Getting Started   Swagger Tools   Community   Blog			
			MUST conform to the defined <code>type</code> for the data type.
maximum	number		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor17">http://json-schema.org/latest/json-schema-validation.html#anchor17</a> .
exclusiveMaximum	boolean		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor17">http://json-schema.org/latest/json-schema-validation.html#anchor17</a> .
minimum	number		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor21">http://json-schema.org/latest/json-schema-validation.html#anchor21</a> .
exclusiveMinimum	boolean		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor21">http://json-schema.org/latest/json-schema-validation.html#anchor21</a> .
maxLength	integer		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor26">http://json-schema.org/latest/json-schema-validation.html#anchor26</a> .
minLength	integer		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor29">http://json-schema.org/latest/json-schema-validation.html#anchor29</a> .
pattern	string		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor33">http://json-schema.org/latest/json-schema-validation.html#anchor33</a> .
maxItems	integer		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor42">http://json-schema.org/latest/json-schema-validation.html#anchor42</a> .
minItems	integer		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor45">http://json-schema.org/latest/json-schema-validation.html#anchor45</a> .
uniqueItems	boolean		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor49">http://json-schema.org/latest/json-schema-validation.html#anchor49</a> .
enum	[*]		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor76">http://json-schema.org/latest/json-schema-validation.html#anchor76</a> .
multipleOf	number		See <a href="http://json-schema.org/latest/json-schema-validation.html#anchor14">http://json-schema.org/latest/json-schema-validation.html#anchor14</a> .

정의를 통해서 요청에 대한 스키마를 작성했으면 이제 해당 테스트를 위해서 새로운 API 경로를 하단과 같이 생성합니다. 이전 강좌의 내용과 동일하게 경로를 '+' 아이콘으로 추가하면 되며 요청의 유형만 방금전에 작성한 정의로 선택하시면 됩니다.

## 경로

+

경로

/checkPut

조작 추가

매개 변수

매개 변수가 정의되지 않음

매개 변수 추가

PUT /checkPut

태그 추가

요약

조작 ID

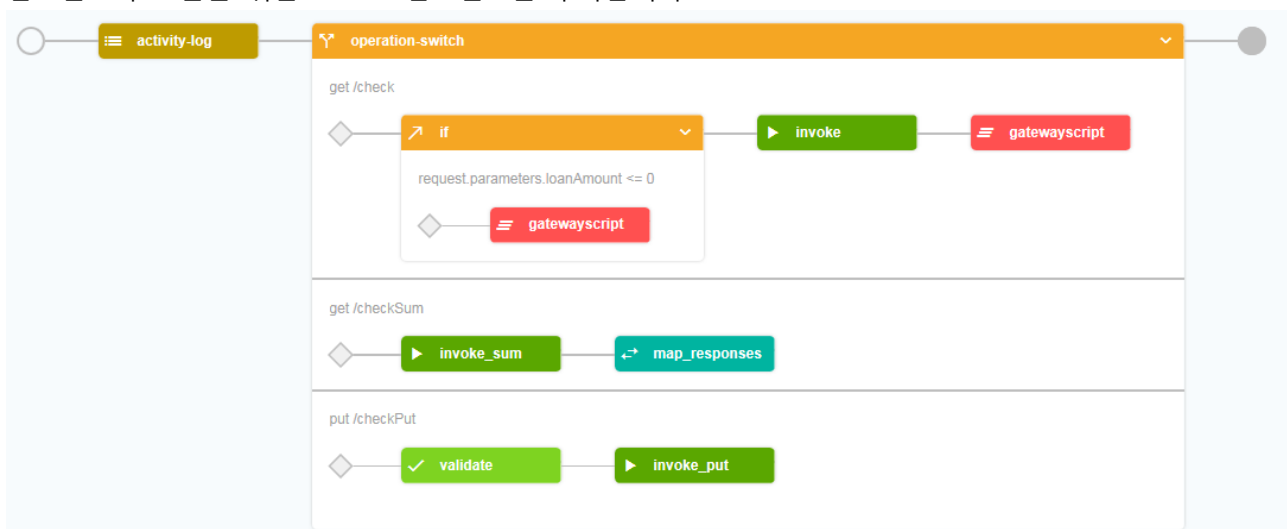
설명

매개 변수

매개 변수 추가

이름	위치	설명	필수	유형
queryBody	본문		<input checked="" type="checkbox"/>	requestLoan

API 어셈블의 flow 에 operation-switch 에 방금 작성한 경로를 추가하고 스키마 검증을 위한 validate 컴포넌트와 호출을 위한 invoke 컴포넌트를 추가합니다.



여기서는 요청을 사전 검증할 것이므로 validate 컴포넌트를 클릭하여 정의를 request 로 선택합니다.

✓ validate | validate

제목

validate

설명

Definition

request

Specify an OpenAPI (Swagger) schema definition with which to validate the payload ('request', 'response', '#/definitions/XXXX' or XXXX).

호출을 위한 invoke 컴포넌트를 클릭하여 하단의 내용을 URL 정보로 넣어줍니다. 이전 강좌와 다른 점은 json body 로 요청을 주기 때문에 파라미터를 request.body 변수 형태로 넣었다는 것입니다.

[http://192.168.225.10:9071/loans/v1/quote?loanAmount=\\${request.body.loanAmount}&annualInterestRate=\\${request.body.annualInterestRate}&termInMonths=\\${request.body.termInMonths}](http://192.168.225.10:9071/loans/v1/quote?loanAmount=${request.body.loanAmount}&annualInterestRate=${request.body.annualInterestRate}&termInMonths=${request.body.termInMonths})

▶ invoke\_put | invoke

제목

invoke\_put

설명

URL

annualInterestRate=\${request.body.annualInterestRate}&termInMonths=\${request.body.termInMonths}

The URL to be invoked.

이제 이전과 같이 테스트를 수행해봅니다. 참고로 put 의 경우에는 스키마가 있으니 테스트시에 '생성'을 통해서 요청을 스키마에 맞추어서 자동 생성할 수 있으며 값만 변경해주면 됩니다. 이때 loanAmount 를 0보다 작은 음수로 넣고 테스트합니다.

#### 매개변수

Content-Type

application/json

Accept

application/json

queryBody

```
{
  "loanAmount": 30947267,
  "annualInterestRate": 37718171.60200533,
  "termInMonths": 85119206
}
```

스키마 표시 | **생성**

해당 테스트를 정상적으로 수행하면 하단과 같이 validate 컴포넌트의 스키마 정의를 통해서 사전 검증에 걸려 400 Invalid Error 가 발생한 것을 확인 가능합니다.

### Response

#### Status code:

400 Invalid

#### Response time:

174ms

#### Headers:

apim-debug-

trans-id: 5756b4ebe4b0f54983fcc7e4-

cfdec2f8-493f-48fe-a641-c6b083fe86ef

content-type: application/json

#### Body:

```
{
  "httpCode": "400",
  "httpMessage": "Invalid",
  "moreInformation": "Validate REST: [JSV008] Invalid number: -1 must be greater than or equal to 0."
}
```

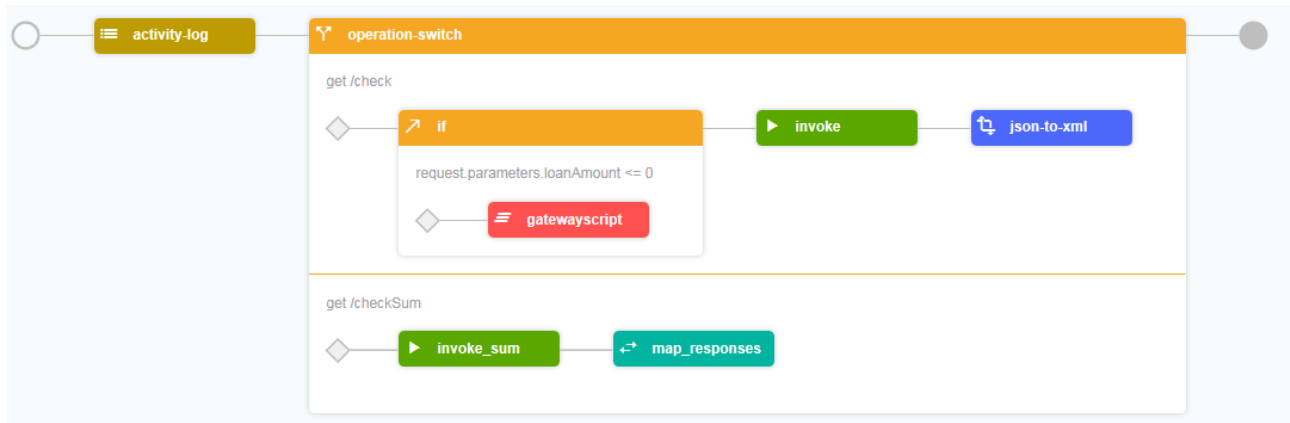
#### validate

[https://www.ibm.com/support/knowledgecenter/en/SSMNED\\_5.0.0/com.ibm.apic.toolkit.doc/rapim\\_ref\\_ootb\\_policyschemaval.html](https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/rapim_ref_ootb_policyschemaval.html)

## 4) json-to-xml 컴포넌트

json to xml 컴포넌트도 역시 이름 그대로 단순히 json 응답을 xml 형태로 변경하는 컴포넌트를 의미합니다.

API 어셈블 디자이너에서 json-to-xml 컴포넌트를 invoke 컴포넌트 뒤에 추가합니다.



기존에 있던 invoke 컴포넌트의 'Response object value' 항목의 지정된 변수명을 삭제하여 그대로 응답으로 반환하게 합니다.

▶ invoke | invoke

The password to use for HTTP Basic authentication.

HTTP Method  
GET

The HTTP method to use for the invocation. If omitted, then the method from the incoming request will be used.

☐ Compression

Used to enable/disable Content-Encoding compression on upload.

Cache Type  
Protocol

The cache type determines whether to cache documents, honoring or overriding the HTTP Cache Control directives received from the response of the Server. This property takes effect only when a response is received from the Server, otherwise, the policy always returns the non-expired response that was previously saved in cache.

Time to Live  
900

Sets the validity period in seconds for documents in the cache. Applies only if the Cache Type is set to 'Time to Live'. Enter a value in the range 5 - 31708800. The default value is 900.

Cache Key

The string to use as a key for the cache. If omitted, the entire URL string is used as the key.






☐ Stop on error

Defines whether the flow stops when a particular error is thrown during the policy execution. Errors not specified here will trigger a catch flow.

Response object variable

The name of a variable that will be used to store the response data from the request. This can then be referenced in other actions, such as 'Map'.

다음으로 json-to-xml 컴포넌트는 수정없이 기본값으로 설정합니다.


**json-to-xml | json-to-xml**





---

제목  
json-to-xml

---

설명

Root XML Element Name  
**json**

---

The root element name of the resultant XML document. This is only required if the input JSON document is not hierarchical and has more than one top level property.

☐ Always output the root element

Output the root element even if it is not required to make the XML document well formed.

위와 같이 하고 테스트를 수행하면 하단과 같이 동일 요청에 대한 동일 응답을 json 형태가 아닌 xml 형태로 받는 것을 확인할 수 있습니다.

## Test

### Status code:

200 OK

### Response time:

159ms

### Headers:

content-type: application/xml  
 x-global-transaction-id: 2343920  
 x-ratelimit-limit: 100  
 x-ratelimit-remaining: 97  
 apim-debug-  
 trans-id: 5756b4ebe4b0f54983fcc7e4-6befd91f-  
 8d8a-4c9f-bb5c-a71a303fea4d

### Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<json>
  <loanAmount>10000</loanAmount>
  <annualInterestRate>0.1</annualInterestRate>
  <termInMonths>10</termInMonths>
  <monthlyPaymentAmount>1000.46</monthlyPaymentAmount>
</json>
```

Debug

**json-to-xml**

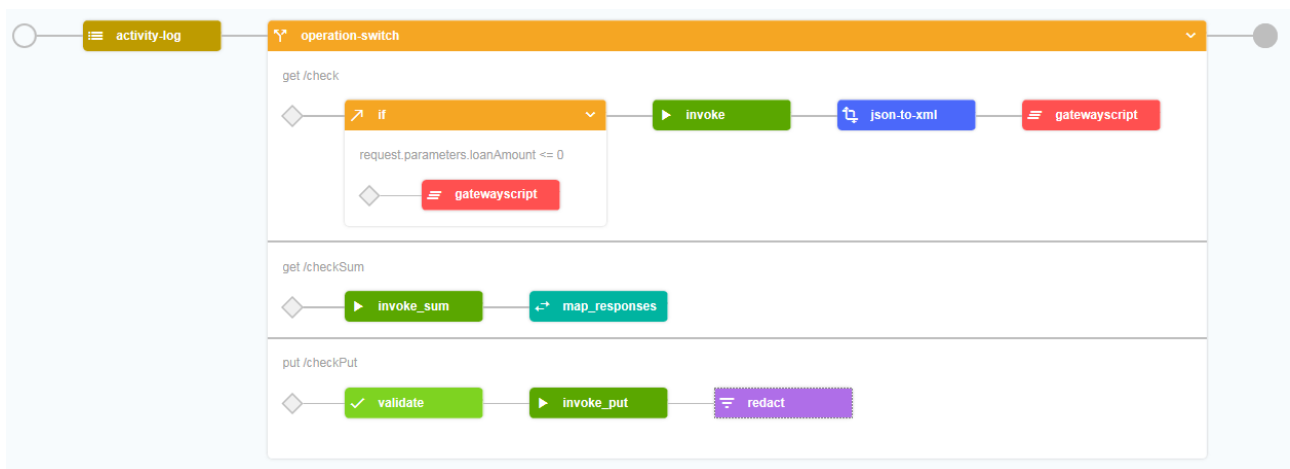
[https://www.ibm.com/support/knowledgecenter/en/SSMNED\\_5.0.0/com.ibm.apic.toolkit.doc/rapim\\_ref\\_oob\\_policyjson2xml.html](https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/rapim_ref_oob_policyjson2xml.html)



## 5) redact 컴포넌트

redact 컴포넌트는 API Connect 입장에서는 쓰임새가 많은 컴포넌트인데 해당 컴포넌트는 손쉽게 요청이나 응답의 특정 필드를 제거하거나 수정하는 역할을 수행합니다. 예를 들어 법적, 보안 또는 기타 이유로 인해 민감한 데이터(예: 신용카드 세부사항)를 제거하거나 차단하는 데 이 정책이 유용할 수 있습니다.

예를 들어 이전까지 테스트한 내용에서 보안상의 이슈로 이자율인 `annualInterestRate` 정보가 응답으로 나가야 하지 않는다면 redact 컴포넌트 추가로 해당 기능을 바로 구현할 수 있습니다.



redact 컴포넌트를 클릭하여 XPath 형태로 `Path( "//*[@name='annualInterestRate']" : name` 이 `annualInterestRate` 인 위치)를 넣은후에 Action 으로 '제거'를 선택합니다. 마지막으로 개정 위치를 'response' 를 선택하면 됩니다.

redact | redact
📌 <> +

**제목**  
 redact

**설명**

**Path**  
 [/@name='annualInterestRate']  
 Use XPath to specify the property to redact or remove.

**Action**

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

response ×    + 개정 위치...

제거

+ 조치

이렇게 redact 컴포넌트를 추가한 후 테스트를 해보면 하단과 같이 기존 응답에서 annualInterestRate가 제외된 채 응답으로 나오는 것을 확인할 수 있습니다.

### Response

#### Status code:

200 OK

#### Response time:

1028ms

#### Headers:

content-type: application/json  
x-global-transaction-id: 31024  
x-ratelimit-limit: 100  
x-ratelimit-remaining: 98  
apim-debug-  
trans-id: 5756b4ebe4b0f54983fcc7e4-  
eb355ae3-129c-4a85-92ee-262152fabf1a

#### Body:

```
{
  "loanAmount": 10000,
  "termInMonths": 10,
  "monthlyPaymentAmount": 1000.46
}
```

#### redact

[https://www.ibm.com/support/knowledgecenter/en/SSMNED\\_5.0.0/com.ibm.apic.toolkit.doc/rapim\\_ref\\_oob\\_policyredaction.html](https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/rapim_ref_oob_policyredaction.html)

## 9) 참고 자료

1. IBM API Connect 5.0 온라인 메뉴얼(Knowledge Center) – 한글 제공  
[https://www.ibm.com/support/knowledgecenter/ko/SSMNED\\_5.0.0/mapfiles/ic\\_home.html](https://www.ibm.com/support/knowledgecenter/ko/SSMNED_5.0.0/mapfiles/ic_home.html)
2. The Assemble view - IBM API Connect 5.0 온라인 메뉴얼  
[https://www.ibm.com/support/knowledgecenter/en/SSMNED\\_5.0.0/com.ibm.apic.toolkit.doc/capi\\_m\\_assembly\\_ui.html](https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/capi_m_assembly_ui.html)