

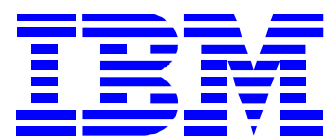
## **IBM API Connect v5.0**

### **API 어셈블(조립/조합)**

(2016. 06. )

IBM Hybrid Cloud Technical Sales

이정운 차장(juwlee@kr.ibm.com)

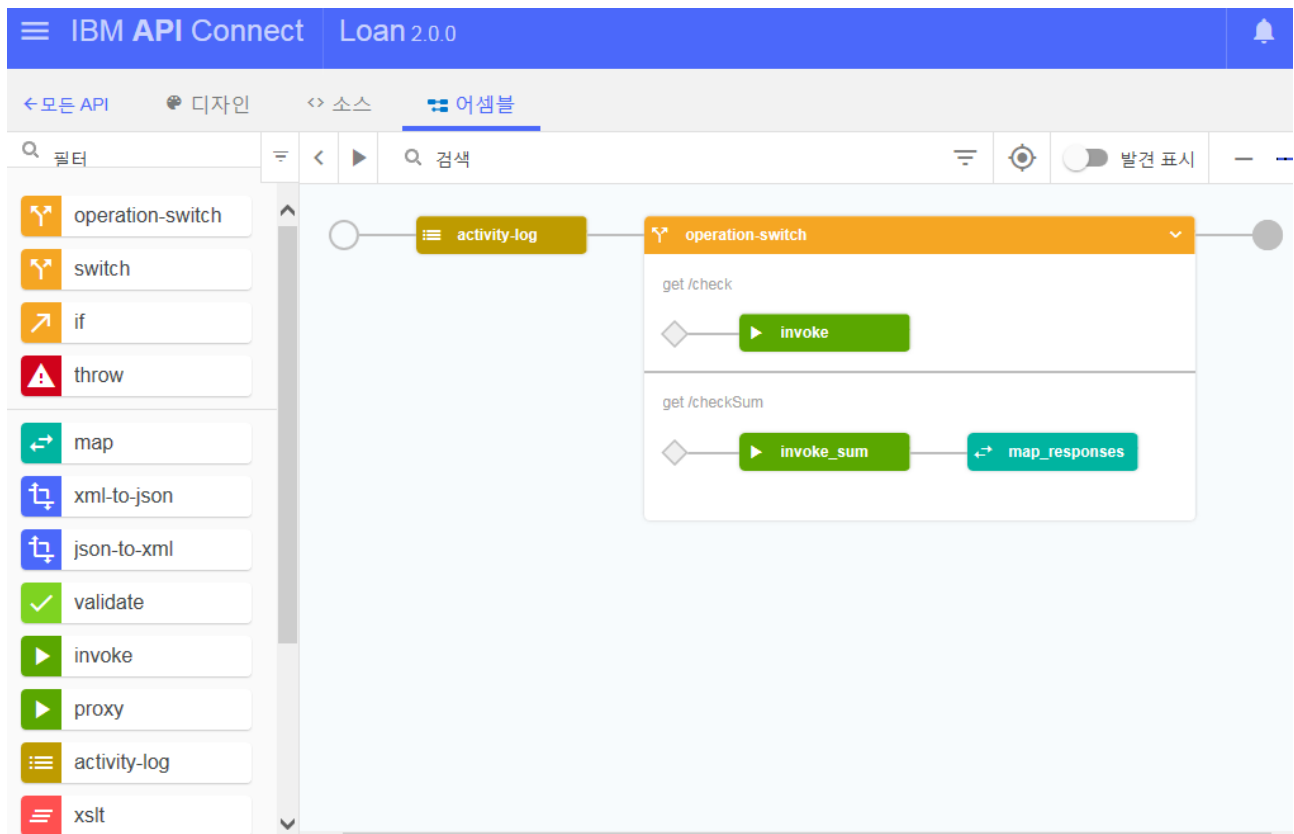


## 0) API 어셈블(조립/조합)이란?

안녕하세요 이정운 입니다.

API 의 경우에는 계속적으로 생성/폐기 될뿐만 아니라 요건에 따라서 여러 개의 API 를 조합/조립해서 하나의 결과로 통합해서 응답을 주어야하는 경우나, Legacy 의 서비스를 그대로 노출하지 않고 전체 응답중에 특정 필드만 API 의 응답으로 제공해 주어야 하는등의 다양한 요건이 존재합니다. 이 때 API 어셈블(Assemble) 이라는 기능이 활용되며 좀 더 쉽게 이야기 드리면 API 조합/조립을 의미합니다.

IBM API Connect 는 IBM 이 오픈 소스로 공개한 Node-Red 기반의 API 어셈블 GUI 를 제공하여 API 조합/조립이 필요할 때 가급적 코딩을 지양하고 이보다는 미리 만들어진 컴포넌트를 드래그&드랍(Drag&Drop) 해서 손쉽게 API 를 조합/조립할 수 있는 방안을 제공합니다.



이번 강좌에서는 이전 강좌에서 응답을 그대로 노출한 Loan API 에 API 어셈블을 추가하여 기존 4개의 value 로 표시되는 응답을 연산해서 하나의 value 만 실제 응답으로 반환하게 하는 형태로 진행해 보도록 하겠습니다.

그럼 백문이 불여일타! API 어셈블에 대한 강좌를 진행하도록 하겠습니다.

## 1) API 어셈블 추가하기

이 강좌는 기본적으로 이미 진행된 "[APIC]본격적으로 살펴보는 IBM API Connect 01 - API 등록 및 테스트" 강좌([http://www.websphere.pe.kr/xe/new\\_lecture/56058](http://www.websphere.pe.kr/xe/new_lecture/56058)) 에서 만들었던 제품과 계획을 활용할 예정입니다.

APIC 의 관리콘솔의 초안 메뉴에서 기존에 만들어둔 Loan 1.0.0 제품 메뉴로 들어와서 우측 상단에 있는 '...' 아이콘을 클릭하여 새 버전으로 저장을 선택합니다.

The screenshot shows the IBM API Connect console interface. The top navigation bar includes 'IBM API Connect' and 'Loan 1.0.0'. The left sidebar lists various menu items like '정보' (Info), '스킴' (Schema), '호스트' (Host), etc. The main content area displays the details for the 'Loan' product, including its name, version (1.0.0), and description. A dropdown menu is open, showing options: '새 버전으로 저장' (Save as new version), '업데이트' (Update), '다운로드' (Download), and '삭제' (Delete). Below this, a modal dialog titled '새 버전으로 저장' (Save as new version) is shown, with a text input field containing '2.0.0' and buttons for '취소' (Cancel) and '새 버전으로 저장' (Save as new version).

API 리스트 화면으로 돌아와 보면 Loan 1.0.0 과 동일하게 '스냅샷' 형태로 복제된 Loan 2.0.0 을 확인할 수 있습니다.

The screenshot shows the IBM API Connect console interface. The top navigation bar includes 'IBM API Connect' and '초안' (Draft). The left sidebar lists various menu items like '제품' (Product), 'API', etc. The main content area displays a table of APIs. The table has columns for '제목' (Title), '마지막 수정 날짜' (Last modified date), '유형' (Type), and '유형' (Type). The table lists four APIs: AccountService 1.0.0 (SOAP), Branches 1.0.0 (REST), Loan 2.0.0 (REST), and Loan 1.0.0 (REST). The 'Loan 2.0.0' entry is highlighted, indicating it is the newly created version. A sidebar on the right shows filters for API types: REST, SOAP, and OAuth 2.0.

제목	마지막 수정 날짜	유형
AccountService 1.0.0	3일 전	SOAP
Branches 1.0.0	3일 전	REST
Loan 2.0.0	몇 초 전	REST
Loan 1.0.0	3일 전	REST

버전 업된 Loan 2.0.0 API 를 클릭하여 왼쪽 메뉴에 정의(Definition)를 선택한 후에 '+' 아이콘을 클릭하여 정의를 하나 추가합니다. 하단에 보이는 것과 같이 이름은 loanResponse 로 하고 특성으로 loanAmount, annualInterestRat, terminMonths, monthlyPayments 를 추가합니다. 이때, 반드시 해당 특성의 이름은 이전 Loan API 의 응답에 있는 JSON 데이터의 name 과 동일해야 합니다.(향후, Map 작성시에 자동으로 매핑을 위해서)

The screenshot shows the 'loanResponse' definition in the API Designer. The left sidebar contains a menu with options like '정보', '스킴', '호스트', '기본 경로', '이용', '생성', '라이프사이클', '정책 어셈블리', '보안 정의', '보안', 'Swagger 확장기능', '특성', '경로', '/check', '매개 변수', '정의', 'paymentResponse', and 'loanResponse'. The main area displays the 'loanResponse' definition with the following details:

- 이름:** loanResponse
- 유형:** object
- 설명:**
- 특성:**

* 특성 이름	설명	유형	Example	조치
<input type="checkbox"/> loanAmount		integer	10000	<> [icon]
<input type="checkbox"/> annualInterestRat		float	0.1	<> [icon]
<input type="checkbox"/> terminMonths		integer	52	<> [icon]
<input type="checkbox"/> monthlyPayment		float	123.45	<> [icon]
- 추가 특성 허용:** ☐

또한, 정의를 하나 더 추가하여 float 형의 특성을 하나 추가합니다. 해당 정의는 추가되는 호출의 응답으로 정의할 것입니다.(기 언급한 것처럼 이전에 JSON 형태로 각각 4개의 value 응답을 1개로 통합해서 반환하기 위해서 사용됩니다.)

### 정의

The screenshot shows the 'paymentResponse' definition in the API Designer. The left sidebar contains a menu with options like '정보', '스킴', '호스트', '기본 경로', '이용', '생성', '라이프사이클', '정책 어셈블리', '보안 정의', '보안', 'Swagger 확장기능', '특성', '경로', '/check', '매개 변수', '정의', 'paymentResponse', and 'loanResponse'. The main area displays the 'paymentResponse' definition with the following details:

- 이름:** paymentResponse
- 유형:** object
- 설명:**
- 특성:**

* 특성 이름	설명	유형	Example	조치
<input type="checkbox"/> R_payment		float	123.45	<> [icon]
- 추가 특성 허용:** ☐

이제 새로운 API 호출 오퍼레이션을 생성하기 위하여 경로에서 '+' 아이콘을 클릭하여 /checkSum 이라는 경로로 Get 오퍼레이션을 하나 추가하여 기존에 check 와 동일하게 매개변수를 지정합니다.

GET
/checkSum

태그 추가

요약

조작 ID

설명

매개 변수

매개 변수 추가

이름	위치	설명	필수	유형	
loanAmount	조회 ▼	대출금	<input checked="" type="checkbox"/>	float ▼	
annualInterestRate	조회 ▼	이자율	<input checked="" type="checkbox"/>	float ▼	
termInMonths	조회 ▼	대출기간	<input checked="" type="checkbox"/>	integer ▼	

그리고 /checkSum 의 Get 호출의 응답에 대한 스키마를 기본값인 object 에서 지금 작성한 paymentResponse 정의로 변경합니다.

응답

응답 추가

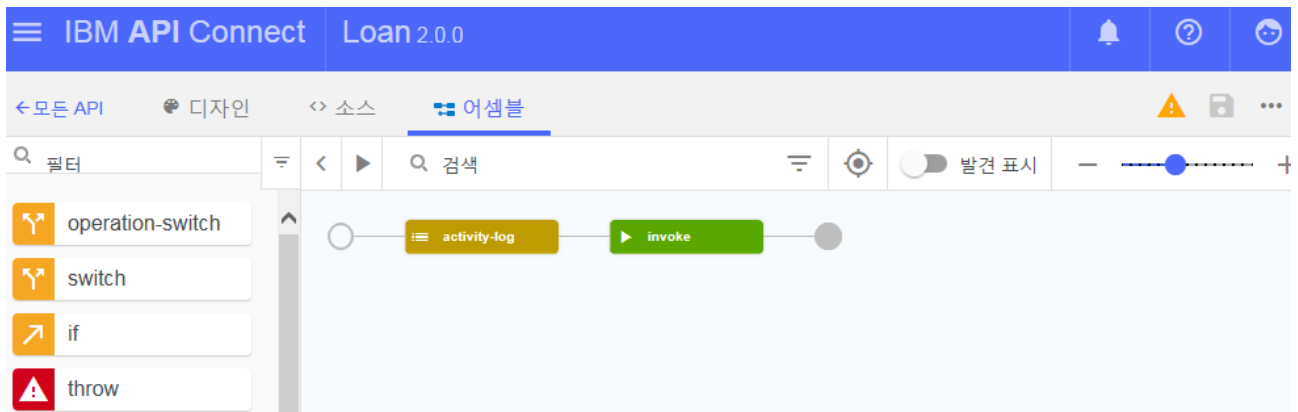
상태 코드	설명	스키마	
200	200 OK	paymentResponse ▼	

보안

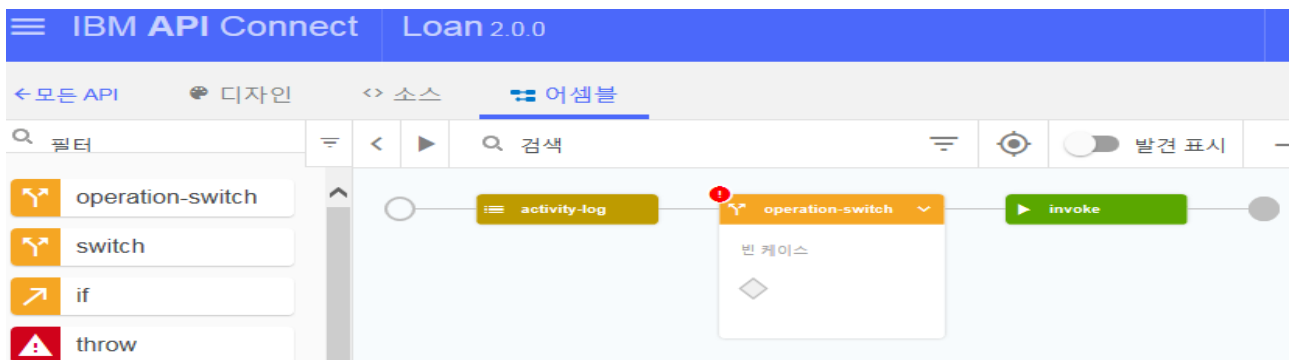
보안 정의 사용

☒ API 보안 정의 사용
☐ clientID (API 키)

이제 정의와 응답에 대한 준비가 완료되었으니 실제로 API 어셈블을 수행해보도록 하겠습니다. 어셈블 메뉴를 선택하여 GUI 디자이너 화면을 확인합니다.



operation-switch 를 하나 추가해서 기존에 있던 invoke 컴포넌트를 이동시킵니다. (operation-switch 는 하나의 API 에 여러 operation 이 존재할 경우 이를 어셈블 차원에서 스위치 하는 역할을 수행)



operation-switch 를 클릭하여 케이스에 이미 만들어진 'get /check' 를 선택합니다. 이렇게 하면 'get /check' 를 위한 하나의 flow 가 생성됩니다.

operation-switch | operation-switch

제목  
operation-switch

설명

케이스	액션	삭제
케이스 0	get /check	×

조작 검색...

+ 케이스

activity-log

operation-switch

get /check

invoke\_get

다시 operation switch 를 클릭하여 이제 'get /checkSum' 에 대한 flow 를 추가 생성합니다.

operation-switch | operation-switch

제목  
operation-switch

설명

케이스	액션	삭제
케이스 0	get /check	×
케이스 1	get /checkSum	×

조작 검색...

+ 케이스

activity-log

operation-switch

get /check

invoke

get /checkSum

추가된 /checkSum flow 에 invoke 컴포넌트를 추가한 후 이름을 변경하고 이를 클릭하여 Response object variable 에 변수로 사용할 이름을 추가합니다.

▶
invoke\_sum | invoke

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

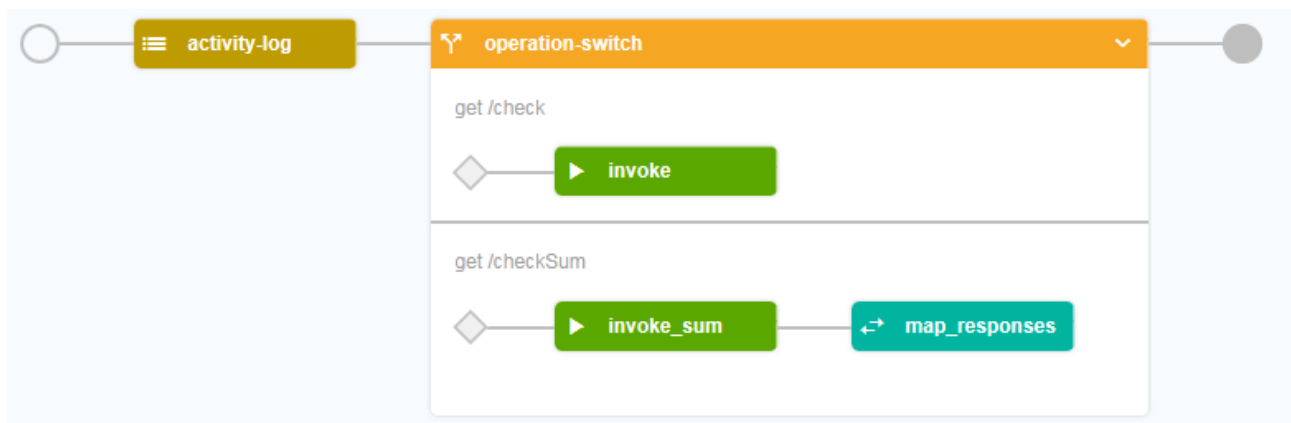
Defines whether the flow stops when a particular error is thrown during the policy execution. Errors not specified here will trigger a catch flow.

Response object variable

responseLoanOut

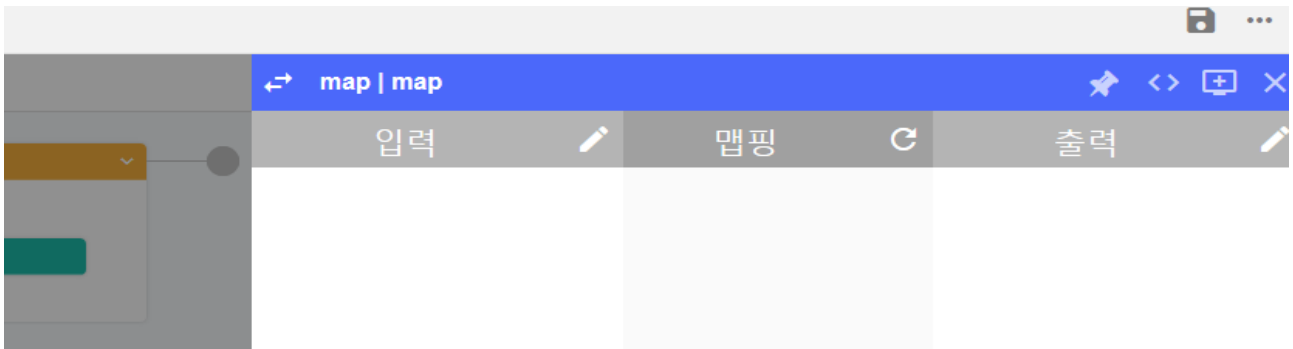
The name of a variable that will be used to store the response data from the request. This can then be referenced in other actions, such as 'Map'.

또한, 실제로 기존 응답을 하나의 value 로 손쉽게 매핑하기 위해서 왼쪽 메뉴에 있는 map 을 드래그&드랍하여 해당 flow 에 추가하고 이름을 변경합니다.

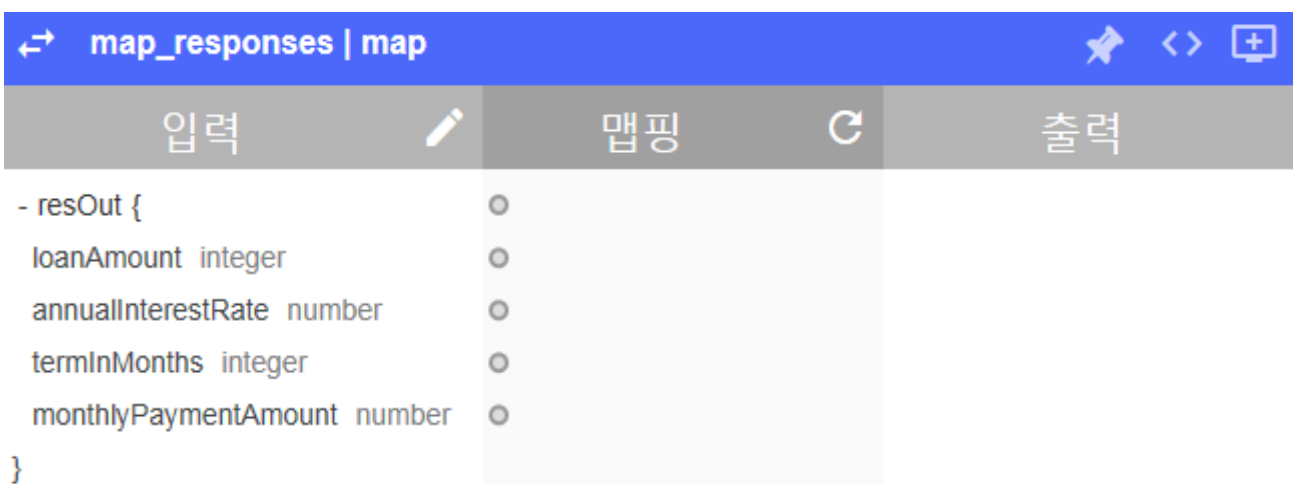
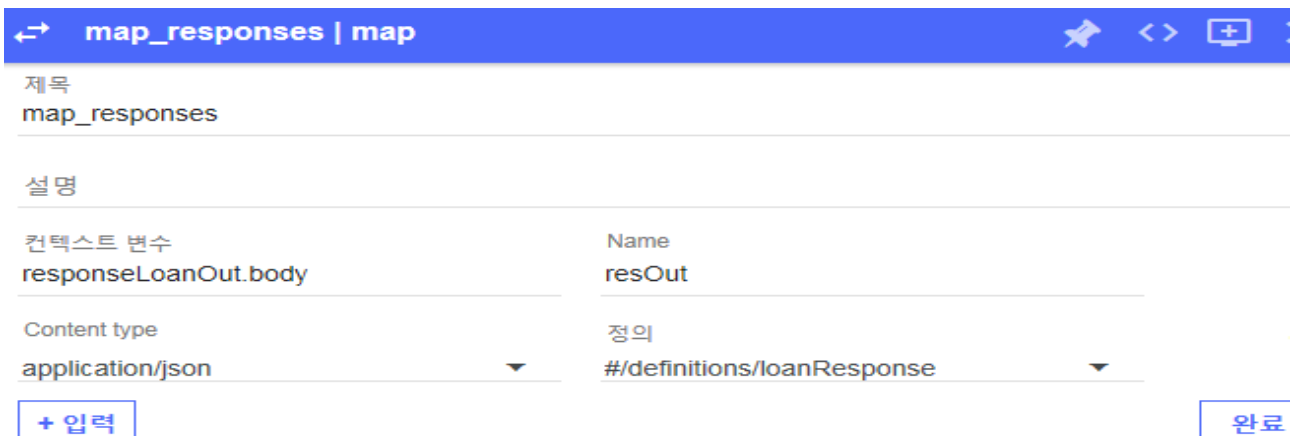




이제 매핑정보를 넣기 위해 Map 을 클릭해보면 하단과 같이 입력과 맵핑, 출력이 나오는 것을 확인할 수 있습니다.



입력 옆에 있는 연필 아이콘을 클릭하여 컨텍스트 변수로 invoke 컴포넌트의 응답으로 지정된 변수의 body 를 지정하고 참조할 이름을 넣어준 후 이전에 만들어둔 정의를 선택하고 완료를 클릭합니다. 그러면 정의를 기반으로 자동으로 입력칸이 완성된 것을 확인할 수 있습니다.



마찬가지로 출력 옆에 있는 연필 아이콘을 클릭하여 이전에 응답을 위해 만들어둔 정의를 넣어둡니다. 참고적으로 여기서 컨텍스트 변수로 사용된 `message.body` 는 응답의 `body` 를 의미함으로 그냥 나누시면 됩니다.

↔ map\_responses | map
📌 <> +

컨텍스트 변수 message.body	Name output
Content type application/json	정의 #/definitions/paymentResponse

+ 출력
완료

↔ map\_responses | map
📌 <> + ✕

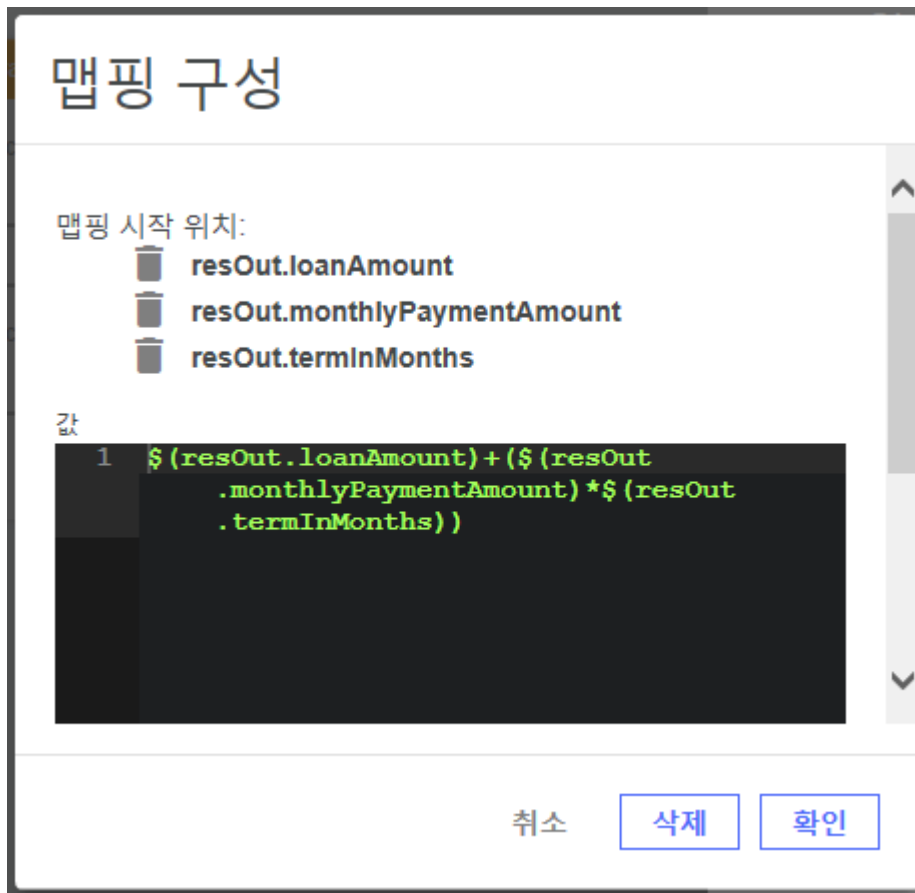
입력	맵핑	출력
- resOut { loanAmount integer annualInterestRate number termInMonths integer monthlyPaymentAmount number }		- output { R_payment number }

이렇게 입력과 출력에 대한 준비가 완료되면 입력쪽에 변수의 우측의 동그란 아이콘을 클릭하여 내가 연계하고자 하는 변수끼리 선을 연결합니다. 예를 들어 `loanAmount` 값과 `annualInterestRate`, `monthlyPaymentAmount` 값을 사용해서 계산을 한다면 세 개의 변수를 출력의 인풋으로 받아야 하기 때문에 하단과 같이 선을 연결하면 됩니다.

↔ map\_responses | map
📌 <> + ✕

입력	맵핑	출력
- resOut { loanAmount integer annualInterestRate number termInMonths integer monthlyPaymentAmount number }		- output { R_payment number }

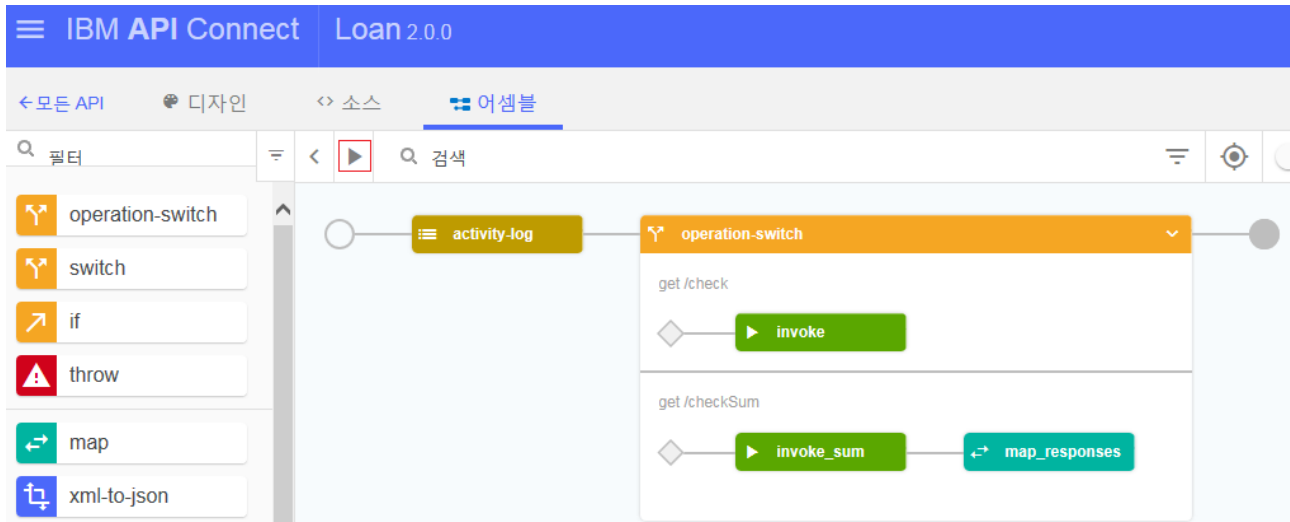
연결선이 완료되면 출력에 있는 변수(R\_payment) 왼쪽의 작은 동그라미를 클릭합니다. 그러면 맵핑 구성화면이 나타나며 javascript 형태로 간단한 맵핑구성과 관련된 액션을 추가할 수 있습니다. 여기서는 하단과 같이 간단한 계산식을 수행하도록 작성합니다. (참고적으로 IBM API Connect 는 변수들을 실제 사용할 때 \$(변수) 와 같은 구문으로 사용합니다.)



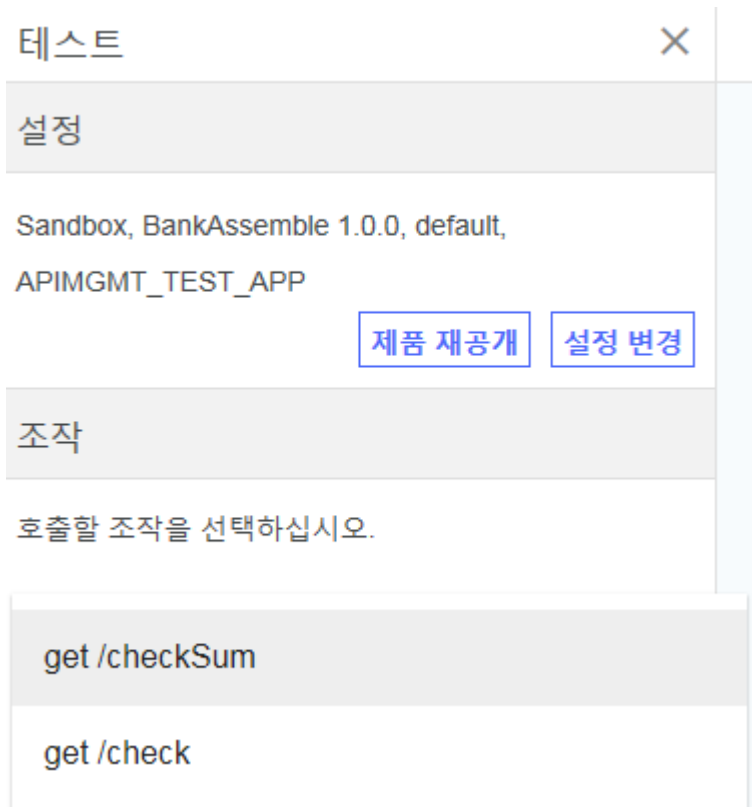
이제 해당 API 를 저장하면 API 어셈블에 대한 준비는 다 완료된 것입니다.

## 2) API 어셈블 테스트하기

이전 강좌에서 언급한 것처럼 이번 강좌에서 만든 API 를 테스트하기 위해서는 제품에 추가한 후 배포 작업을 수행해야 합니다. 해당 제품에 대해서 배포가 완료되면 API 어셈블 메뉴에서 ▶ 아이콘을 클릭 하며 바로 테스트 해볼 수 있습니다.



테스트를 수행하기 위해서 호출한 조작을 선택하고(get /checkSum) 테스트 값을 입력한 후 호출을 수행 합니다.



테스트

Content-Type  
application/json

Accept  
application/json

대출금  
loanAmount  
10000  
생성

이자율  
annualInterestRate  
0.1  
생성

대출기간  
terminMonths  
10  
생성

해당 호출이 정상적으로 수행되면 이전 결과와 다르게 맵핑 구성에 넣은 것처럼 계산된 결과만이 JSON으로 하나의 value로 응답되는 것을 확인할 수 있습니다.

Response

Status code:  
200 OK

Response time:  
202ms

Headers:  
content-type: application/json  
x-global-transaction-id: 2011504  
x-ratelimit-limit: 100  
x-ratelimit-remaining: 99  
apim-debug-  
trans-id: 5756b4ebe4b0f54983fcc7e4-5b76630c-  
70c4-4715-8e07-f1e9c85456f7

Body:  

```
{  
  "R_payment": 20004.6  
}
```

Debug

참고 : API 어셈블 관련 작업중에 좀 더 자세한 디버깅 로그를 확인하려면 하단과 같이 API Gateway 의 관리콘솔로 들어가서 'View Logs' 를 선택하여 좀 더 자세한 로그를 확인 가능합니다.

The screenshot shows the IBM DataPower Gateway Control Panel. The left sidebar contains a 'Control Panel' section with a 'Blueprint Console' and a search bar. Below the search bar is a list of navigation items: Status, Services, Network, Administration, and Objects. The main content area is titled 'Control Panel' and features a 'Services' section with four icons: Web Service Proxy, Multi-Protocol Gateway, XML Firewall, and Web Application Firewall. Below this is a 'Monitoring and Troubleshooting' section with three icons: View Logs (highlighted with a red box), Troubleshooting, and View Status. The bottom of the sidebar shows firmware information: IDG.7.5.0.1, Build: 276222, IBM DataPower Gateway, Copyright IBM Corporation 1999-2016, and a link to the license agreement.

The screenshot shows the IBM DataPower Gateway System Log. The left sidebar is identical to the previous screenshot. The main content area is titled 'System Log' and features a 'Refresh Log' button, a 'Target' dropdown (set to 'default-log'), a 'Filter' dropdown (set to '(none)'), and a 'Help' link. Below this is a table of log entries. The table has columns for time, category, level, tid, direction, client, msgid, and message. The log entries are dated Friday, June 10, 2016. The first four entries are warnings from 'xsalt' at 6:10:07 PM, all with the same message: 'mpgw (webapi): Pattern letter z ignored, because time zone was not specified.' The next three entries are errors from 'apiconnect' at 5:51:47 PM, all with the same message: 'mpgw (webapi): Map: invalid value expression and no default configured.' The final two entries are errors from 'apiconnect' at 5:51:47 PM, both with the same message: 'mpgw (webapi): Map: expression: resOut.loanAmount+(resOut.monthlyPaymentAmount\*resOut.termInMonths) results in NaN'.

time	category	level	tid	direction	client	msgid	message
6:10:07 PM	xsalt	warning	2011504	request	192.168.225.1	0x80a00387	mpgw (webapi): Pattern letter z ignored, because time zone was not specified.
6:10:07 PM	xsalt	warning	2011472	request	192.168.225.1	0x80a00387	mpgw (webapi): Pattern letter z ignored, because time zone was not specified.
5:52:03 PM	xsalt	warning	1997568	request	192.168.225.1	0x80a00387	mpgw (webapi): Pattern letter z ignored, because time zone was not specified.
5:52:03 PM	xsalt	warning	1996688	request	192.168.225.1	0x80a00387	mpgw (webapi): Pattern letter z ignored, because time zone was not specified.
5:51:47 PM	apiconnect	error	1996640	request	192.168.225.1	0x8580005c	mpgw (webapi): Map: invalid value expression and no default configured.
5:51:47 PM	apiconnect	error	1996640	request	192.168.225.1	0x8580005c	mpgw (webapi): Map: expression: resOut.loanAmount+(resOut.monthlyPaymentAmount*resOut.termInMonths) results in NaN
5:51:47 PM	apiconnect	error	1996640	request	192.168.225.1	0x8580005c	mpgw (webapi): Map: resOut.termInMonths is undefined
5:51:47 PM	apiconnect	error	1996640	request	192.168.225.1	0x8580005c	mpgw (webapi): Map: resOut.monthlyPaymentAmount is undefined

## 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)
3. Swagger and assembly components - IBM API Connect 5.0 온라인 메뉴얼  
[https://www.ibm.com/support/knowledgecenter/en/SSMNED\\_5.0.0/com.ibm.apic.toolkit.doc/rapi\\_m\\_api\\_components.html](https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/rapi_m_api_components.html)
4. Gatewayscript - IBM API Connect 5.0 온라인 메뉴얼  
[https://www.ibm.com/support/knowledgecenter/ko/SSMNED\\_5.0.0/com.ibm.apic.toolkit.doc/rapi\\_m\\_ref\\_ootb\\_policygatewayscript.html](https://www.ibm.com/support/knowledgecenter/ko/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/rapi_m_ref_ootb_policygatewayscript.html)