



HCL Volt MX

Integration サービス

Student Guide



HCL Software Academy for HCL Digital Solutions

Creating a new generation of experts

もくじ

Integration サービス	3
前提条件	3
Foundry で Integration サービスを設定する	3

Integration サービス

このレッスンでは、HCL Volt MX Foundry の Integration サービスについて紹介します。このレッスンでは、以下について学習します。

- Foundry で Integration サービスを構成する方法

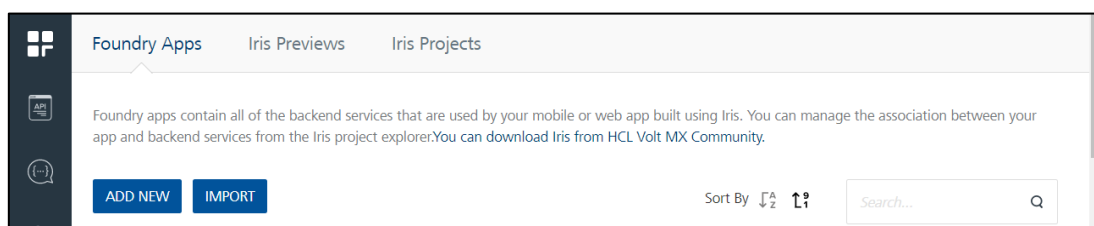
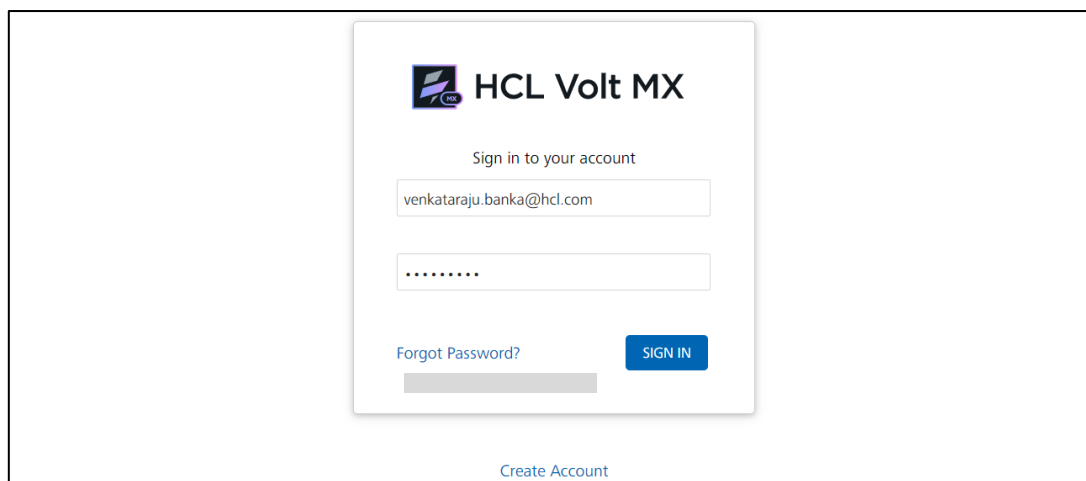
このドキュメントでは、このレッスンの実習部分の詳細な手順を説明します。

前提条件

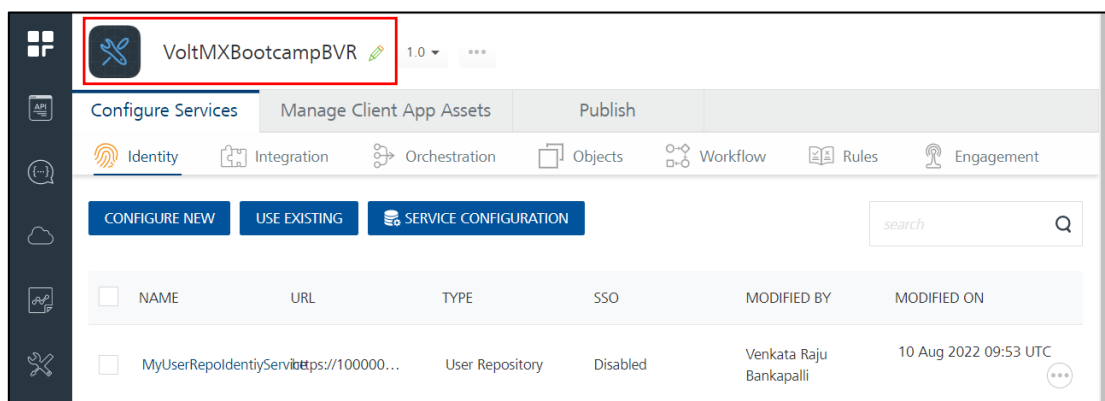
- HCL Volt MX Foundry コンソールにアクセスすることができること。
- 前のレッスンのハンズオンステップを完了していること。

Foundry で Integration サービスを設定する

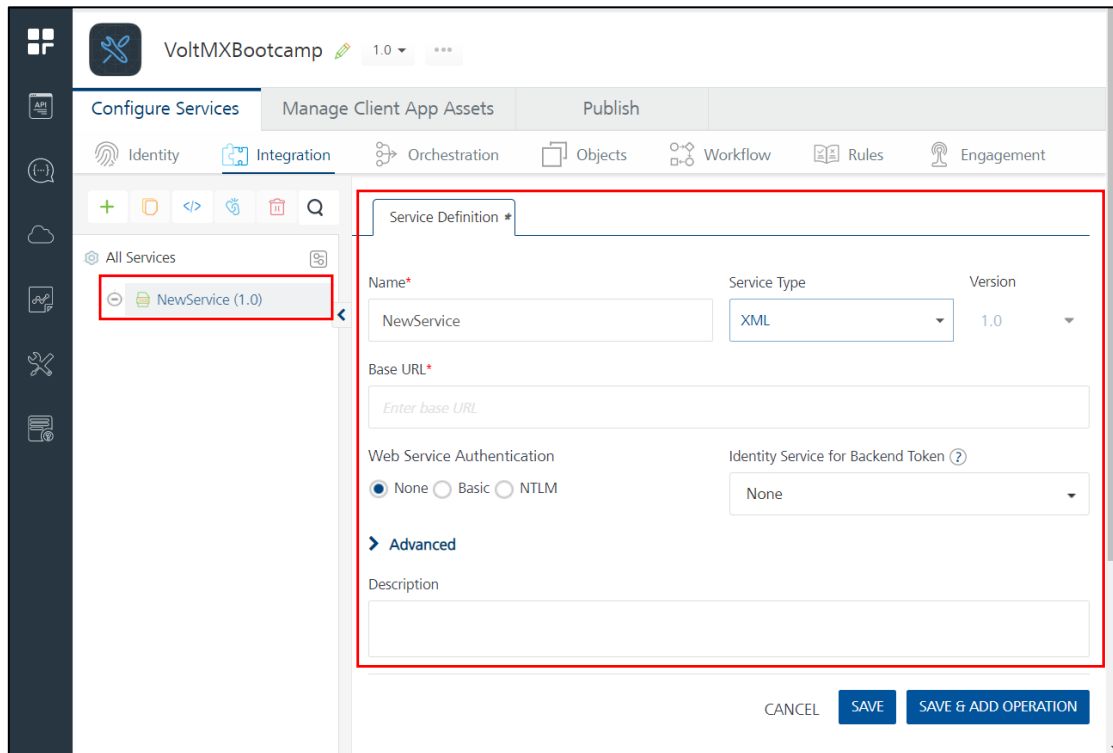
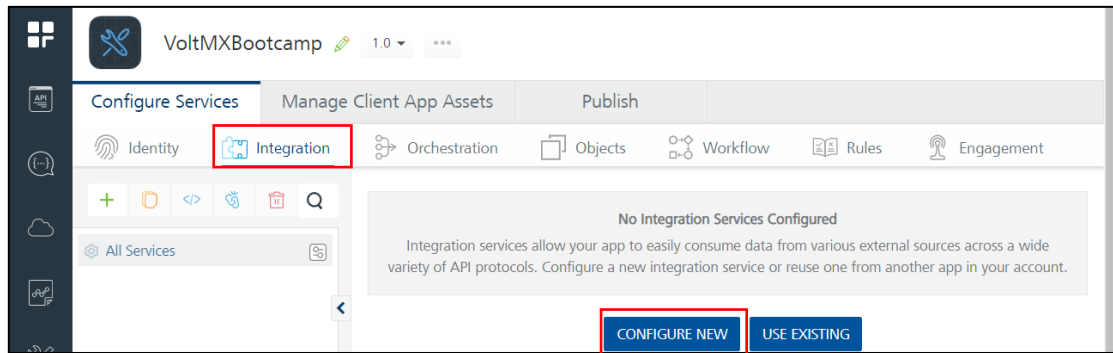
- Foundry コンソールにログインします。



- 前のレッスンで作成した Foundry アプリ (VoltMXBootcamp 等) を開きます。



- **Configure services > Integration** の下にある **CONFIGURE NEW** をクリックします。



- Integration サービスの名前を **NewYorkTimesNewsFeed** とします。
- Service Type** に **XML** を選択します。
- Base URL** の値として、**https://rss.nytimes.com/services/xml/rss/nyt/** を使用します。
- SAVE & ADD OPERATION** をクリックします。Integration サービスが作成されます。

VoltMXBootcamp 1.0

Configure Services | Manage Client App Assets | Publish

Identity | Integration | Orchestration | Objects | Workflow | Rules | Engagement

All Services

- NewService (1.0)

Service Definition *

Name*	Service Type	Version
NewYorkTimesNewsFeed	XML	1.0

Base URL*

https://rss.nytimes.com/services/xml/rss/nyt/

Web Service Authentication

☒ None ☐ Basic ☐ NTLM

Identity Service for Backend Token (?)

None

> Advanced

Description

CANCEL SAVE **SAVE & ADD OPERATION**

VoltMXBootcamp 1.0

Configure Services | Manage Client App Assets | Publish

Identity | Integration | Orchestration | Objects | Workflow | Rules | Engagement

All Services

- NewYorkTimesNewsF...

Service Definition | **Operations List** | NewOperation * %

Name*	Operation Security Level
NewOperation	Authenticated A

Target URL

https://rss.nytimes.com/services/xml/rss/nyt/ Enter Suffix

Target HTTP Method (?)

GET

> Advanced

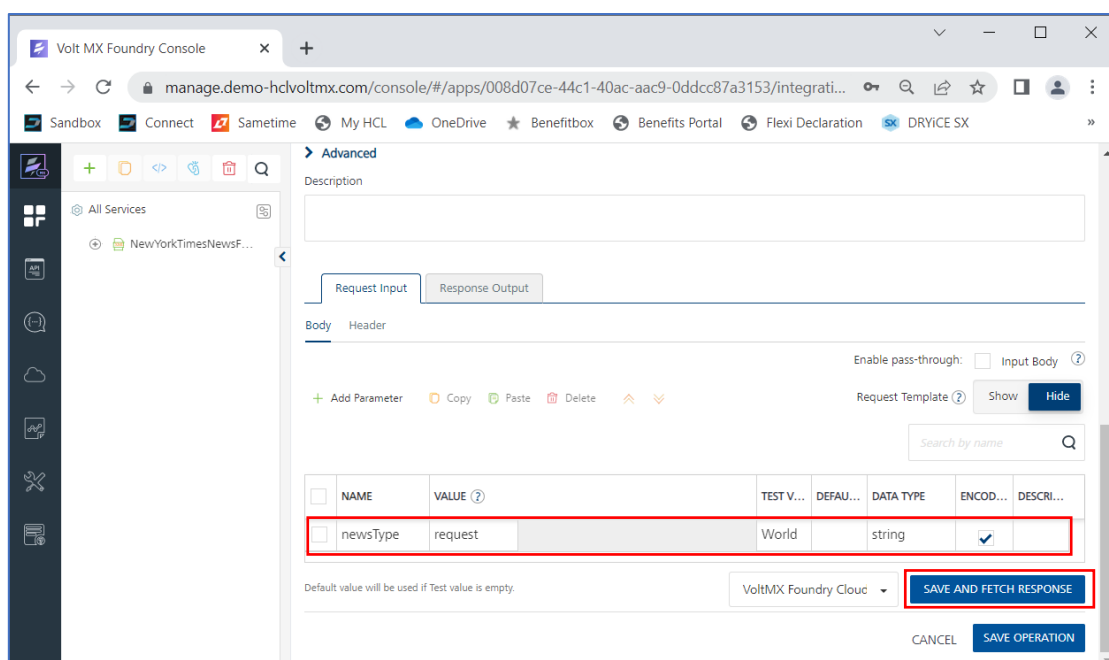
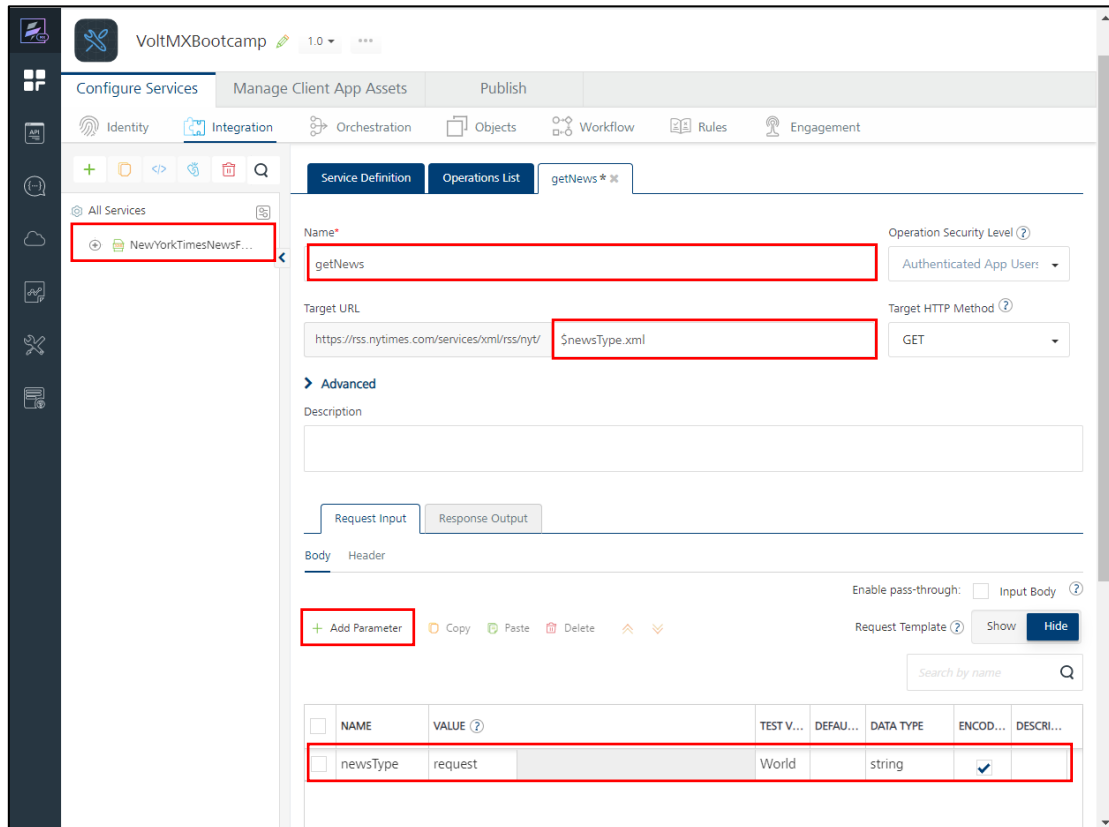
Description

Request Input | Response Output

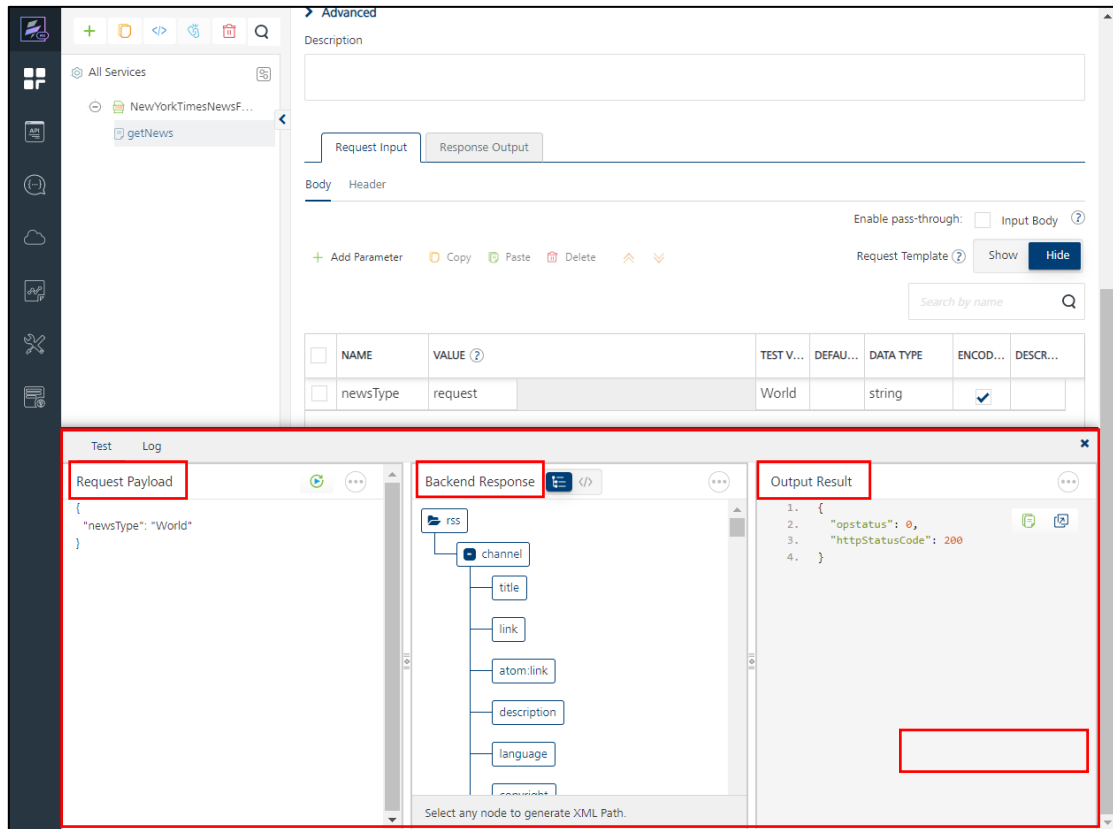
Body | Header

Enable pass-through: ☐ Input Body (?)

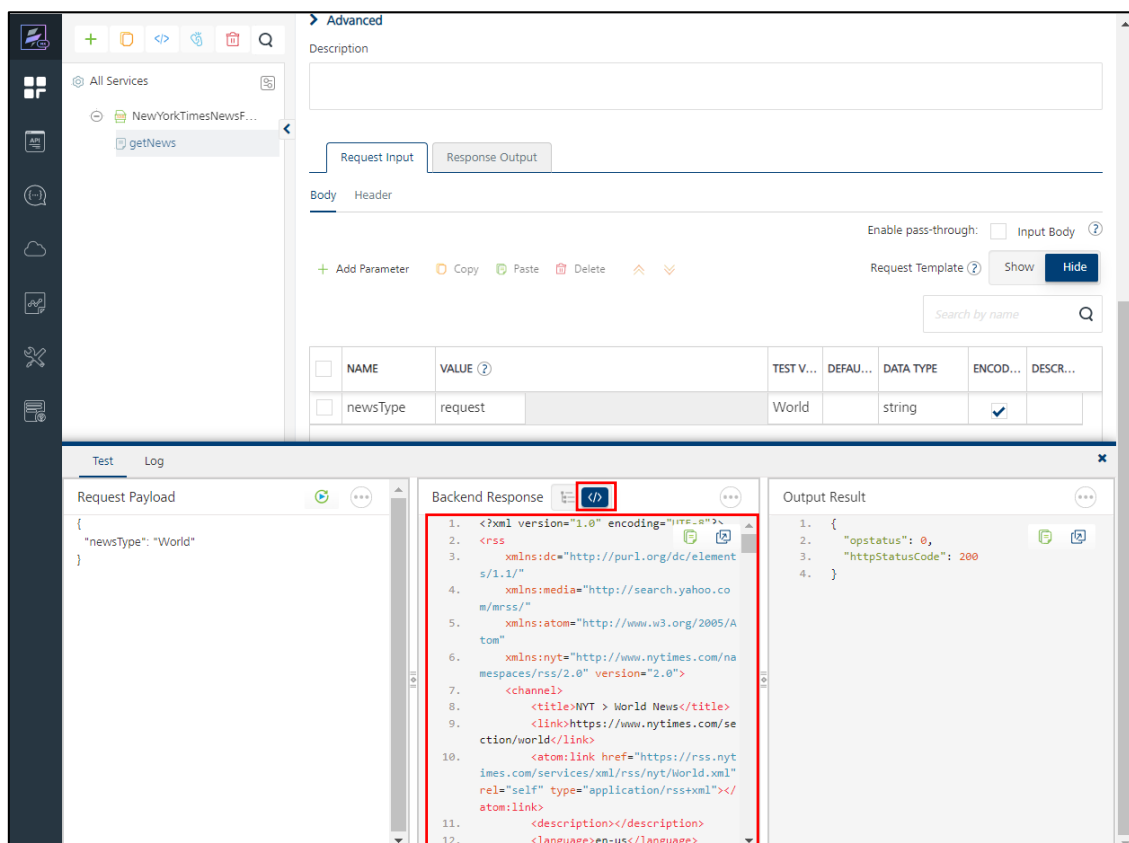
- 新しいオペレーションを設定します。 **NewOperation** タブで、詳細を設定します。
 - Name** の値を **getNews** にします。
 - Target URL** に **\$newsType.xml** というサフィックスを入力します。
- Request Input > Body** タブで、**Add Parameter** をクリックします。
 - NAME** の値を **newsType** とします。
 - TEST VALUE** の値を **World** にします。
- SAVE AND FETCH RESPONSE** をクリックします。



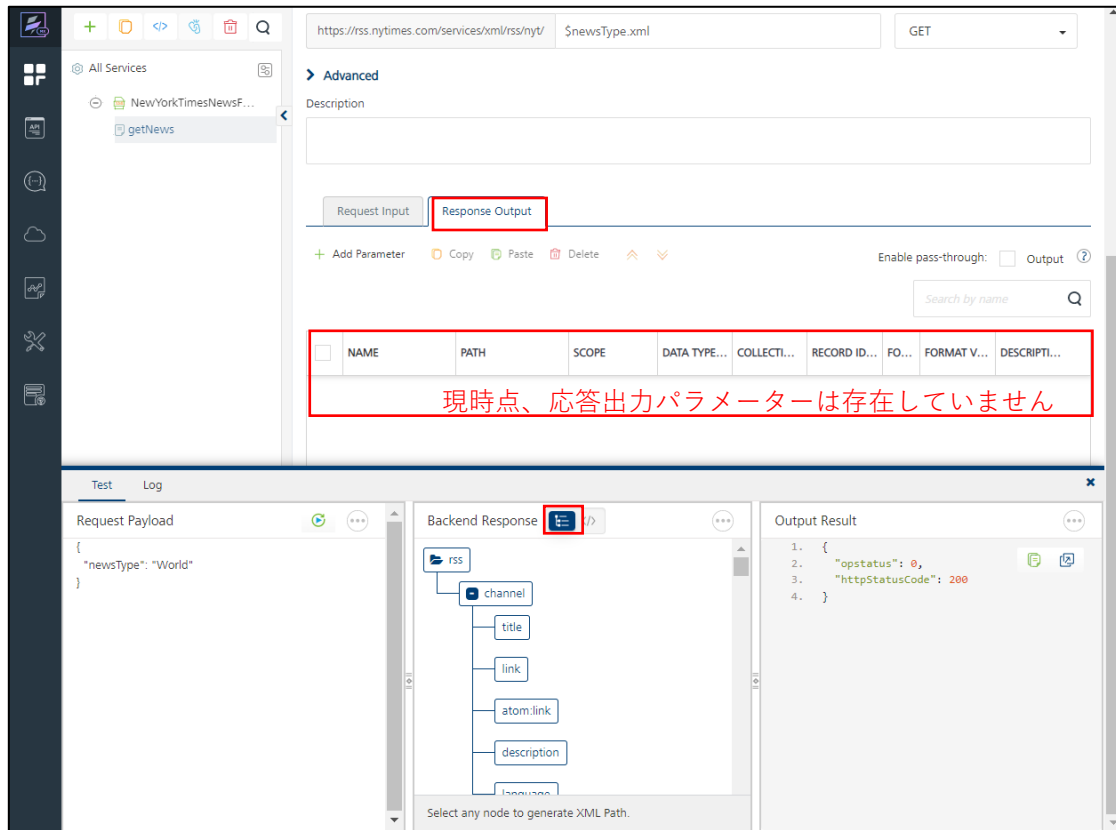
- Request Payload, Backend Response, Output Result をチェックします。



- Backend Response に、NewYorkTimes のニュースフィードから Foundry に送信されたレスポンスが表示されます。



- **Response Output** を設定します。
- **Backend Response** を使ってレスポンス出力を作成します。
- **Backend Response** で、
 - **channel** にカーソルを合わせ、**Create Response** をクリックします。Channel に対する新しいレスポンス出力が作成されます。



The screenshot shows the Volt MX Foundry Console interface. The top bar displays the URL `https://rss.nytimes.com/services/xml/rss/nyt/` and the endpoint `$newsType.xml`. The **Advanced** tab is selected, showing the **Response Output** section. The **Request Input** section shows a request payload: `{ "newsType": "World" }`. The **Backend Response** section shows a tree structure of the response, with the **channel** node highlighted. The **Output Result** section shows the JSON output: `{ "opstatus": 0, "httpStatusCode": 200 }`.

Request Input

```
{ "newsType": "World" }
```

Backend Response

- rss
 - channel- Create response
 - title
 - link
 - atom:link
 - description
 - language

Select any node to generate XML Path.

Output Result

```
1. {
2.   "opstatus": 0,
3.   "httpStatusCode": 200
4. }
```

The screenshot shows the Volt MX Foundry Console interface. The top bar displays the URL `https://rss.nytimes.com/services/xml/rss/nyt/` and the endpoint `$newsType.xml`. The **Advanced** tab is selected, showing the **Response Output** section. The **Request Input** section shows a request payload: `{ "newsType": "World" }`. The **Backend Response** section shows a tree structure of the response, with the **channel** node highlighted. The **Output Result** section shows the JSON output: `{ "opstatus": 0, "httpStatusCode": 200 }`.

Request Input

```
{ "newsType": "World" }
```

Backend Response

- rss
 - channel
 - title
 - link
 - atom:link
 - description
 - language

XPath: `//rss/channel`

Output Result

```
1. {
2.   "opstatus": 0,
3.   "httpStatusCode": 200
4. }
```

- item の下の **title** にカーソルを合わせ、 **Create response** をクリックします。Title に対する新しいレスポンス出力が作成されます。
- item の下にある **link** にカーソルを合わせて、 **Create response** をクリックします。Link に対する新しいレスポンス出力が作成されます。
- item の下にある **description** にカーソルを合わせて、 **Create response** をクリックします。description に対する新しいレスポンス出力が作成されます。
- item の下にある **pubDate** にカーソルを合わせて、 **Create response** をクリックします。pubDate に対する新しいレスポンス出力が作成されます。

The screenshot displays the HCL Volt MX IDE interface for configuring a REST client. The top section shows the URL `https://rss.nytimes.com/services/xml/rss/nyt/$newsType.xml` and the method `GET`. The **Advanced** tab is selected, and the **Response Output** section is highlighted with a red box. Below this, a table lists the response parameters:

NAME	PATH	SCOPE	DATA TYPE...	COLLECTI...	RECORD ID...	FO...	FORMAT V...	DESCRIPTI...
channel	/rss/channel	response	collection			No...		
title	item/title	response	string	channel		No...		
link	item/link	response	string	channel		No...		
description	item/description	response	string	channel		No...		
pubDate	item/pubDate	response	string	channel		No...		

The bottom section shows the **Test** tab with a **Request Payload** and a **Backend Response** tree view. The **Output Result** panel shows the response status and headers:

```

1. {
2.   "opstatus": 0,
3.   "httpStatusCode": 200
4. }
  
```

- **Add Parameter** をクリックします。
 - **NAME** の値を **media** とします。
 - **PATH** の値を **item/media:content/@url** とします。
 - **COLLECTION ID** の値を **channel** にします。

The screenshot displays the Swagger Editor interface for configuring an API endpoint. The top section shows the URL `https://rss.nytimes.com/services/xml/rss/nyt/$newsType.xml` and the HTTP method `GET`. The **Advanced** tab is selected, revealing a description field and a table of response parameters. The **Test** tab is active at the bottom, showing the request payload, the backend response structure, and the output result.

Request Configuration:

- URL: `https://rss.nytimes.com/services/xml/rss/nyt/$newsType.xml`
- Method: `GET`
- Response Type: `$newsType.xml`

Advanced Tab:

- Description: (Empty field)
- Request Input: (Tab)
- Response Output: (Tab)
- Buttons: `+ Add Parameter` (highlighted), `Copy`, `Paste`, `Delete`, `Enable pass-through` (checkbox), `Output` (checkbox), `Search by name` (input), `?` (help icon).

Response Parameters Table:

<input type="checkbox"/>	NAME	PATH	SCOPE	DATA TYPE...	COLLECTI...	RECORD ID...	FO...	FORMAT V...	DESCRIPTI...
<input type="checkbox"/>	channel	/rss/channel	response	collection			No...		
<input type="checkbox"/>	title	item/title	response	string	channel		No...		
<input type="checkbox"/>	link	item/link	response	string	channel		No...		
<input type="checkbox"/>	description	item/description	response	string	channel		No...		
<input type="checkbox"/>	pubDate	item/pubDate	response	string	channel		No...		

Test Tab:

- Request Payload:**

```
{
  "newsType": "World"
}
```
- Backend Response:**

```
{
  description: ...
  dc.creator: ...
  pubDate: ...
  media: {
    content: ...
  }
}
```
- Output Result:**

```
1. {
2.   "opstatus": 0,
3.   "httpStatusCode": 200
4. }
```

The screenshot displays the IBM Cloud Functions console interface for configuring a new function named 'getNews'. The function is triggered by an HTTP GET request to the URL 'https://rs.nytimes.com/services/xml/rss/nyt/\$newsType.xml'. The 'Advanced' tab is selected, showing a description field and a table of parameters. The 'Request Input' tab is also visible, showing a request payload with 'newsType' set to 'World'. The 'Backend Response' tab shows a tree view of the response structure, including 'description', 'dc:creator', and 'pubDate'. The 'Output Result' tab shows the JSON output of the function, including 'opstatus' and 'httpStatusCode'.

Function Configuration:

- Name:** getNews
- Trigger:** HTTP GET
- URL:** https://rs.nytimes.com/services/xml/rss/nyt/\$newsType.xml
- Description:** (Empty field)

Parameters Table:

NAME	PATH	SCOPE	DATA TYPE...	COLLECTI...	RECORD ID...	FO...	FORMAT V...	DESRIPTI...
channel	/rss/channel	response	collection			No...		
title	item/title	response	string	channel		No...		
link	item/link	response	string	channel		No...		
description	item/description	response	string	channel		No...		
pubDate	item/pubDate	response	string	channel		No...		
		response	string			No...		

Request Payload:

```
{
  "newsType": "World"
}
```

Backend Response:

```
{
  "description": "...",
  "dc:creator": "...",
  "pubDate": "..."
}
```

Output Result:

```
{
  "opstatus": 0,
  "httpStatusCode": 200
}
```

URL: `https://rss.nytimes.com/services/xml/rss/nyt/$newsType.xml` Method: `GET`

Advanced

Description

Request Input Response Output

+ Add Parameter Copy Paste Delete

Enable pass-through: ☐ Output ?

Search by name

<input type="checkbox"/>	NAME	PATH	SCOPE	DATA TYPE...	COLLECTI...	RECORD ID...	FO...	FORMAT V...	DESCRIPTI...
<input type="checkbox"/>	channel	/rss/channel	response	collection			No...		
<input type="checkbox"/>	title	item/title	response	string	channel		No...		
<input type="checkbox"/>	link	item/link	response	string	channel		No...		
<input type="checkbox"/>	description	item/description	response	string	channel		No...		
<input type="checkbox"/>	pubDate	item/pubDate	response	string	channel		No...		
<input type="checkbox"/>	media	item/media:con...	response	string	channel		No...		

Test Log

Request Payload

```
{
  "newsType": "World"
}
```

Backend Response

```

  description
  dc:creator
  pubDate
  XPath: //rss/channel/item/pubDate

```

Output Result

```

1. {
2.   "opstatus": 0,
3.   "httpStatusCode": 200
4. }

```

- Add Parameter をクリックします。
 - NAME の値を **newsTitle** とします。
 - PATH の値を **/rss/channel/title** とします。

URL: `https://rss.nytimes.com/services/xml/rss/nyt/$newsType.xml` Method: `GET`

Advanced

Description

Request Input Response Output

+ Add Parameter Copy Paste Delete

Enable pass-through: ☐ Output ?

Search by name

<input type="checkbox"/>	NAME	PATH	SCOPE	DATA TYPE...	COLLECTI...	RECORD ID...	FO...	FORMAT V...	DESCRIPTI...
<input type="checkbox"/>	channel	/rss/channel	response	collection			No...		
<input type="checkbox"/>	title	item/title	response	string	channel		No...		
<input type="checkbox"/>	link	item/link	response	string	channel		No...		
<input type="checkbox"/>	description	item/description	response	string	channel		No...		
<input type="checkbox"/>	pubDate	item/pubDate	response	string	channel		No...		
<input type="checkbox"/>	media	item/media:con...	response	string	channel		No...		
<input type="checkbox"/>	newsTitle	/rss/channel/title	response	string			No...		

Test Log

Request Payload

```
{
  "newsType": "World"
}
```

Backend Response

```

  description
  dc:creator
  pubDate
  XPath: //rss/channel/item/pubDate

```

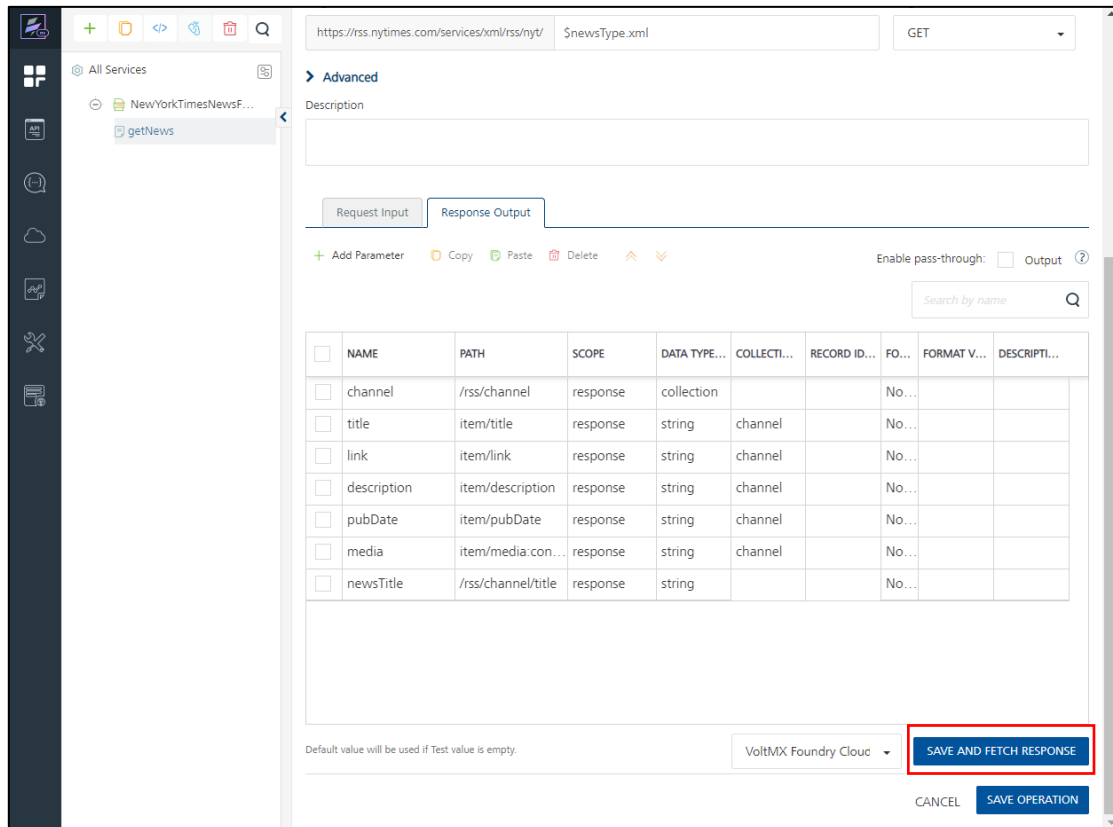
Output Result

```

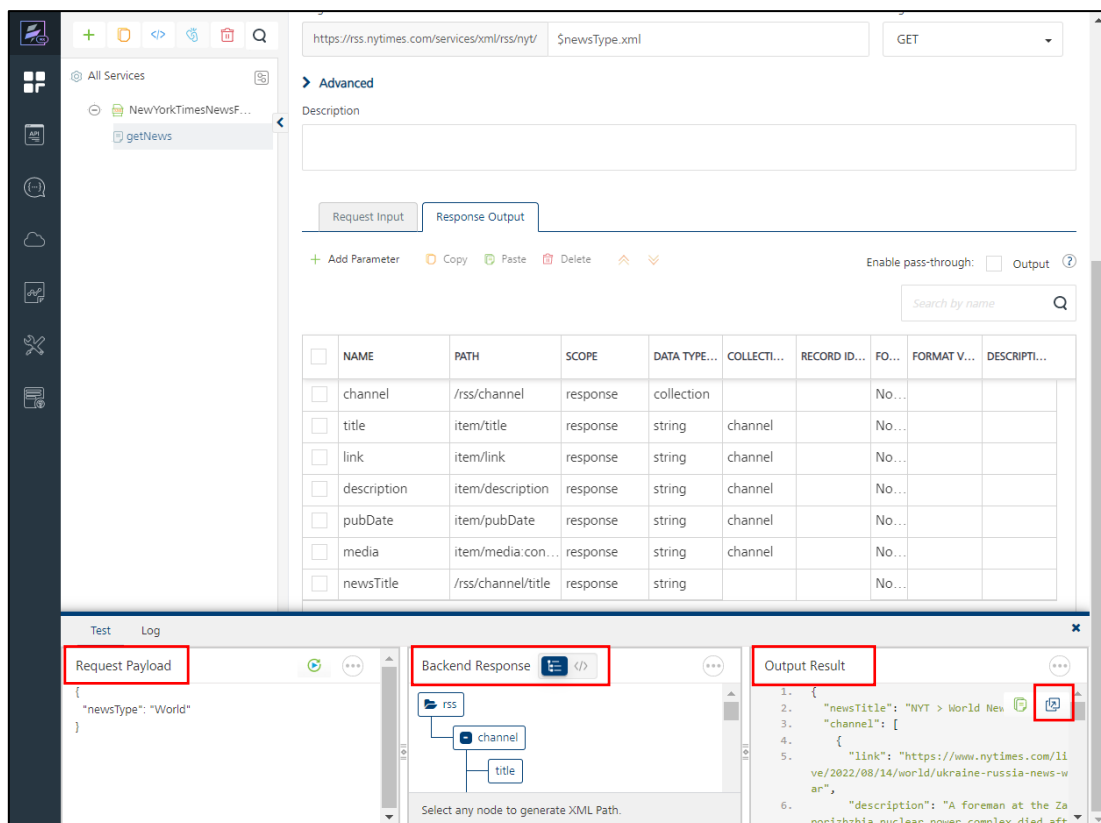
1. {
2.   "opstatus": 0,
3.   "httpStatusCode": 200
4. }

```

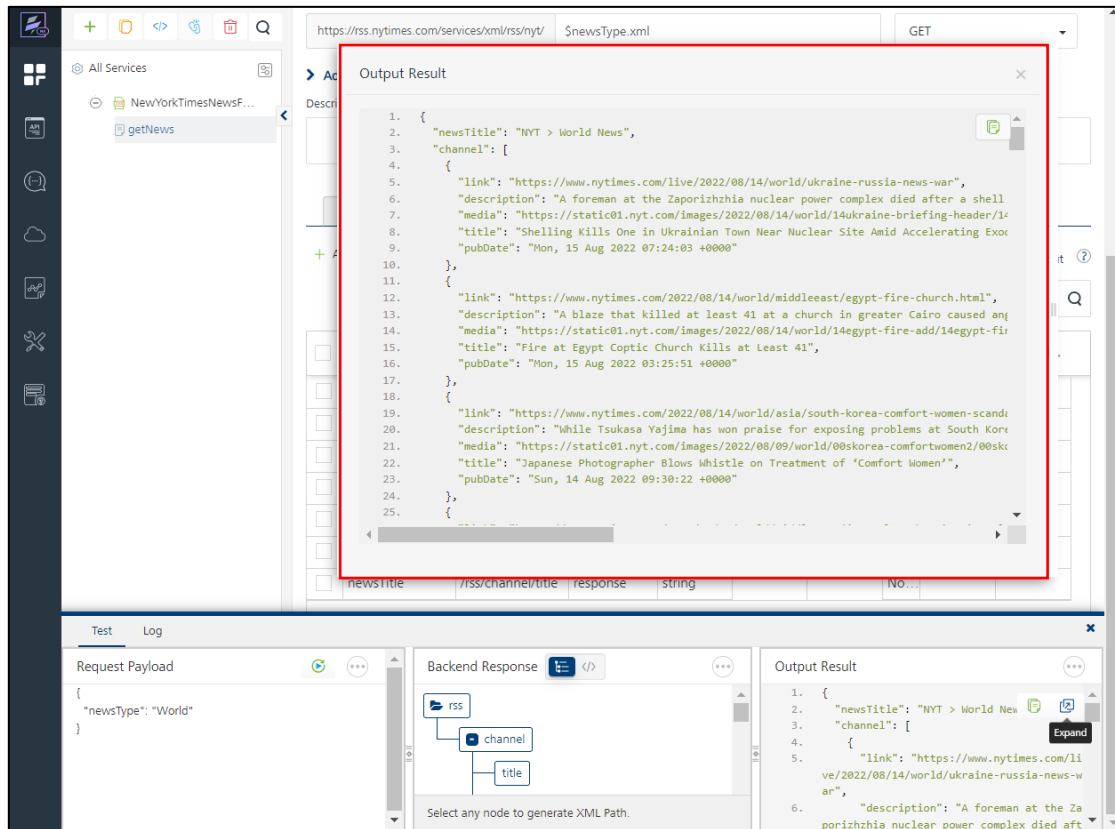
- **SAVE AND FETCH RESPONSE** をクリックします。



- **Request Payload, Backend Response, Output Result** を確認します。



- **Output Result** には、Foundry がフロントエンドアプリケーションに送信するレスポンスが表示されます。



おめでとうございます。あなたはこのレッスンの実習部分を完了しました。