



HCL Volt MX

Orchestration サービス

Student Guide



HCL Software Academy for HCL Digital Solutions

Creating a new generation of experts

もくじ

Orchestration サービス	3
前提条件	3
Findry で Integration サービスを構成して、場所の詳細が与えられた都市の名前を取得する	3
都市の天気の詳細を取得するための Integration サービスを Foundry で構成する	10
位置情報に基づいて天気の詳細を取得するための Foundry の Orchestration サービスの構成	18

Orchestration サービス

このレッスンでは、HCL Volt MX Foundry の Orchestration 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 Orchestration サービスを構成する方法

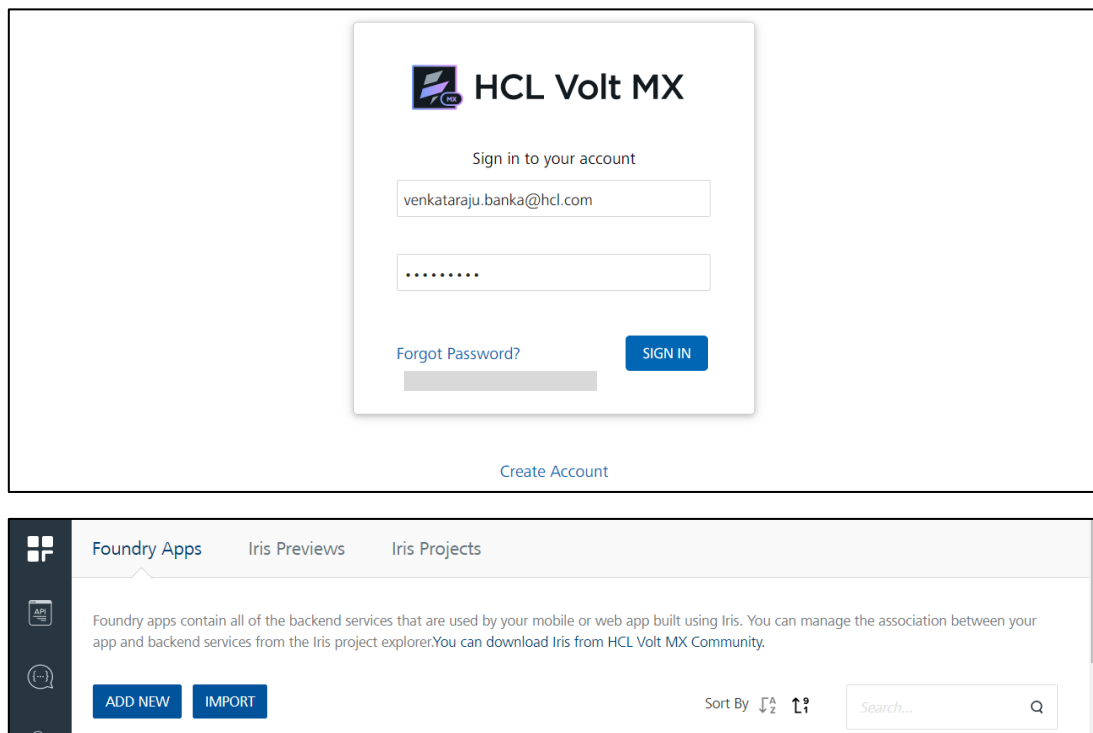
このドキュメントでは、このレッスンのハンズオン部分の詳細な手順を説明します。

前提条件

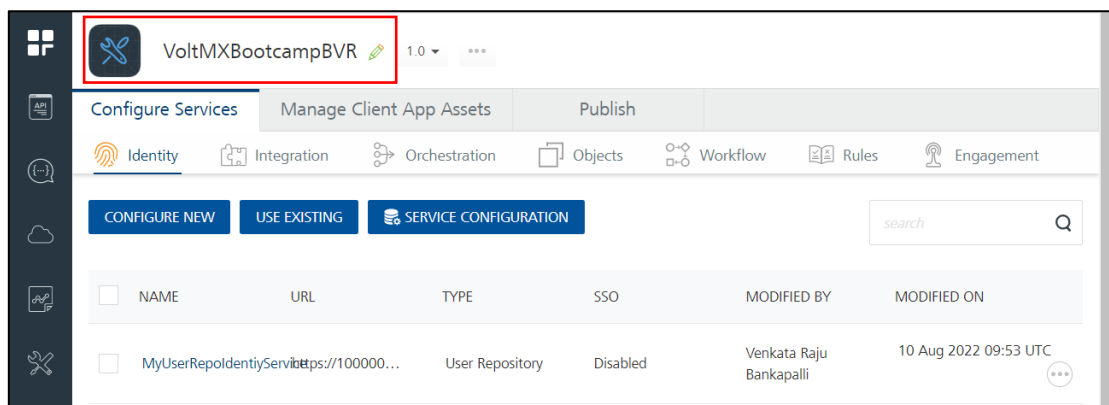
- HCL Volt MX Foundry コンソールにアクセスできる。
- 前のレッスンでハンズオンの手順を完了していること。

Foundry で Integration サービスを構成して、場所の詳細が与えられた都市の名前を取得する

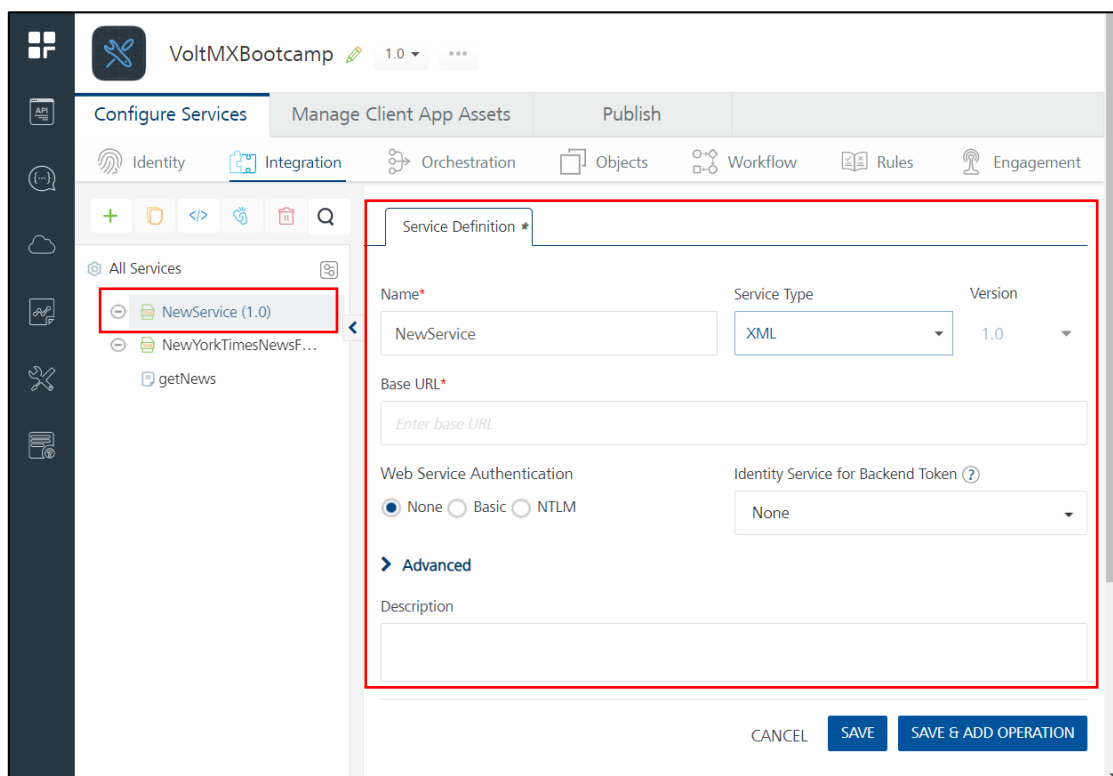
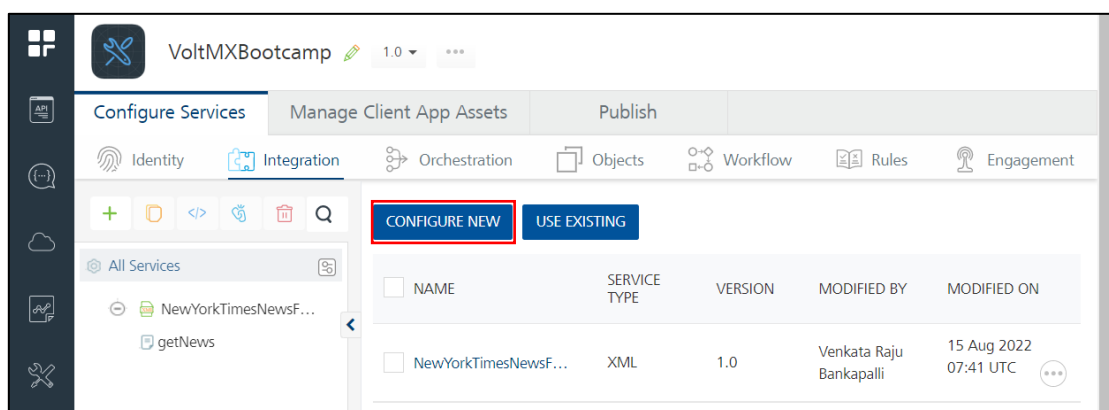
- Foundry コンソールにログインします。



- 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 **OpenstreetLocationsService** とします。
- Service Type** を **XML** と選択します。
- Base URL** の値として **http://nominatim.openstreetmap.org/** 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

Service Definition *

Name* OpenstreetLocationsService Service Type XML Version 1.0

Base URL* http://nominatim.openstreetmap.org/

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

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

All Services: OpenstreetLocationsS... NewYorkTimesNewsF... getNews

Name* NewOperation Operation Security Level Authenticated

Target URL http://nominatim.openstreetmap.org/ Enter Suffix Target HTTP Method GET

> Advanced

Description

Request Input | Response Output

Body Header

Enable pass-through: ☐ Input Body (?)

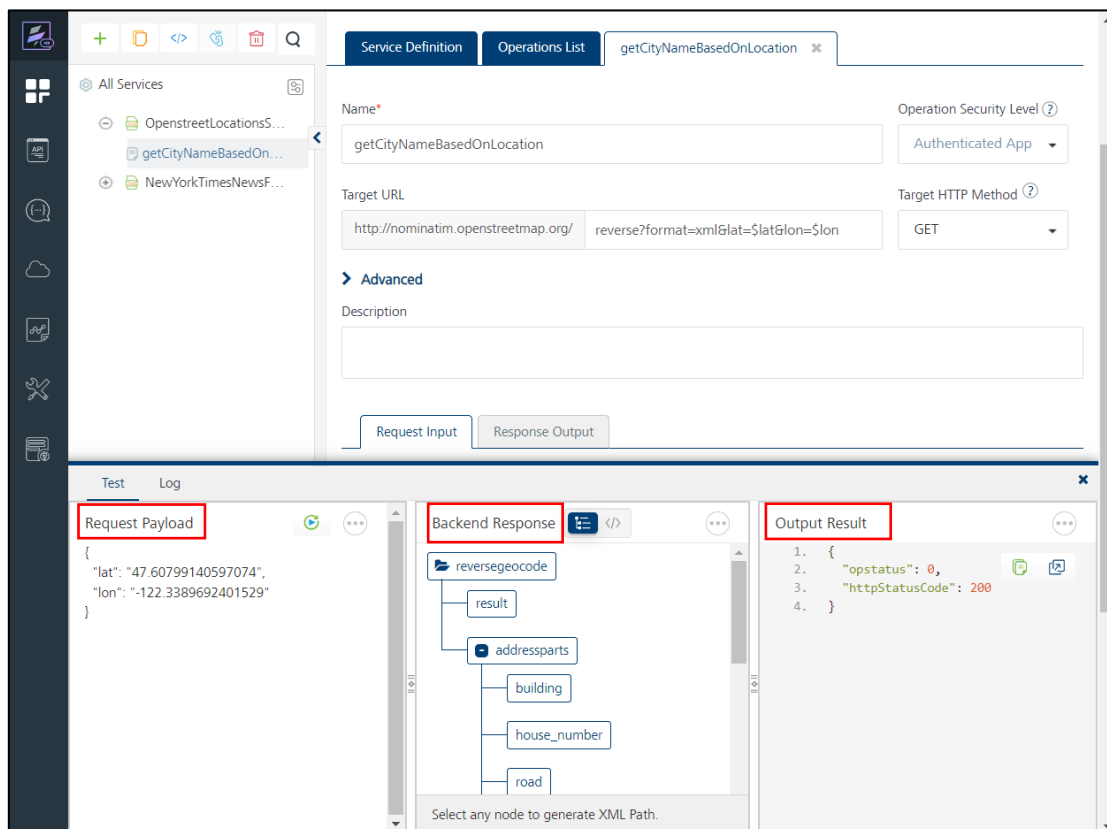
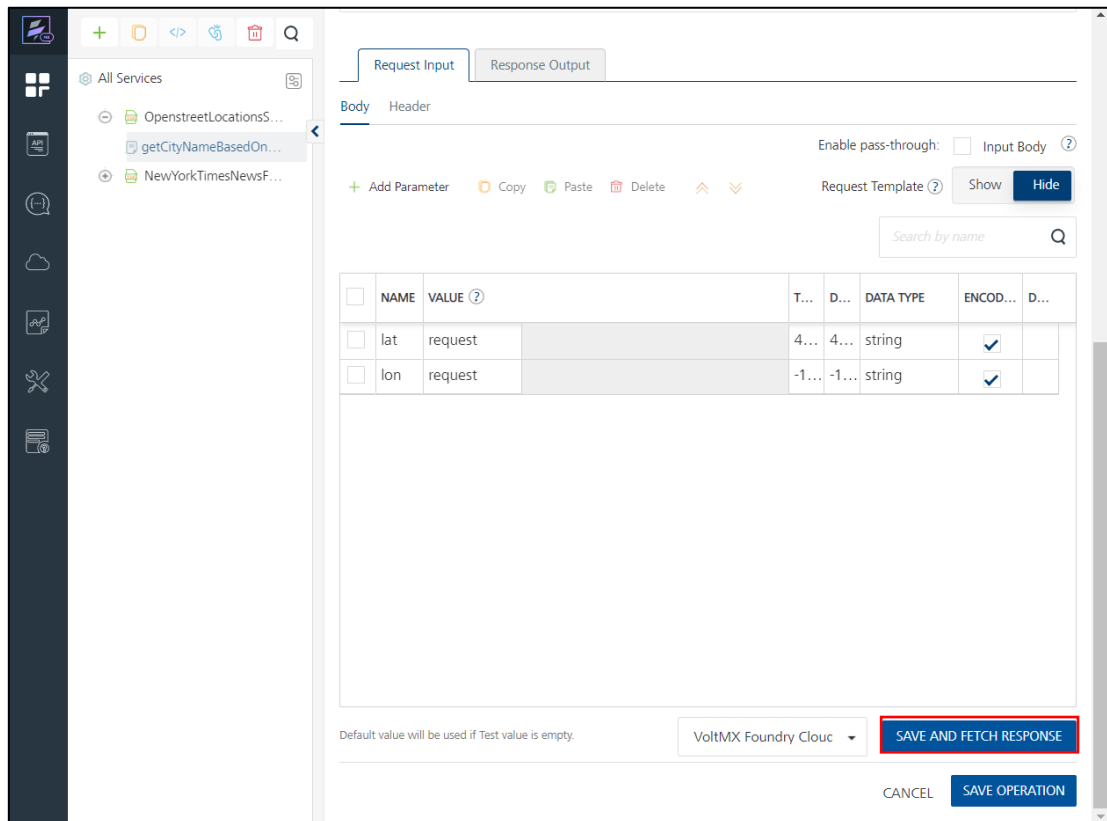
+ Add Parameter Copy Paste Delete Request Template (?) Show Hide

- **NewOperation** タブで、
 - **Name** の値を **getCityNameBasedOnLocation** とします。
 - **Target URL** にサフィックス **reverse?format=xml&lat=\$lat&lon=\$lon** 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 **lat** とします。
 - **TEST VALUE** と **DEFAULT VALUE** の値を **47.60799140597074** とします。
 - **Request Input > Body** タブで **Add Parameter** をクリックします。
 - **NAME** の値を **lon** とします。
 - **TEST VALUE** と **DEFAULT VALUE** の値を **-122.3389692401529** とします。

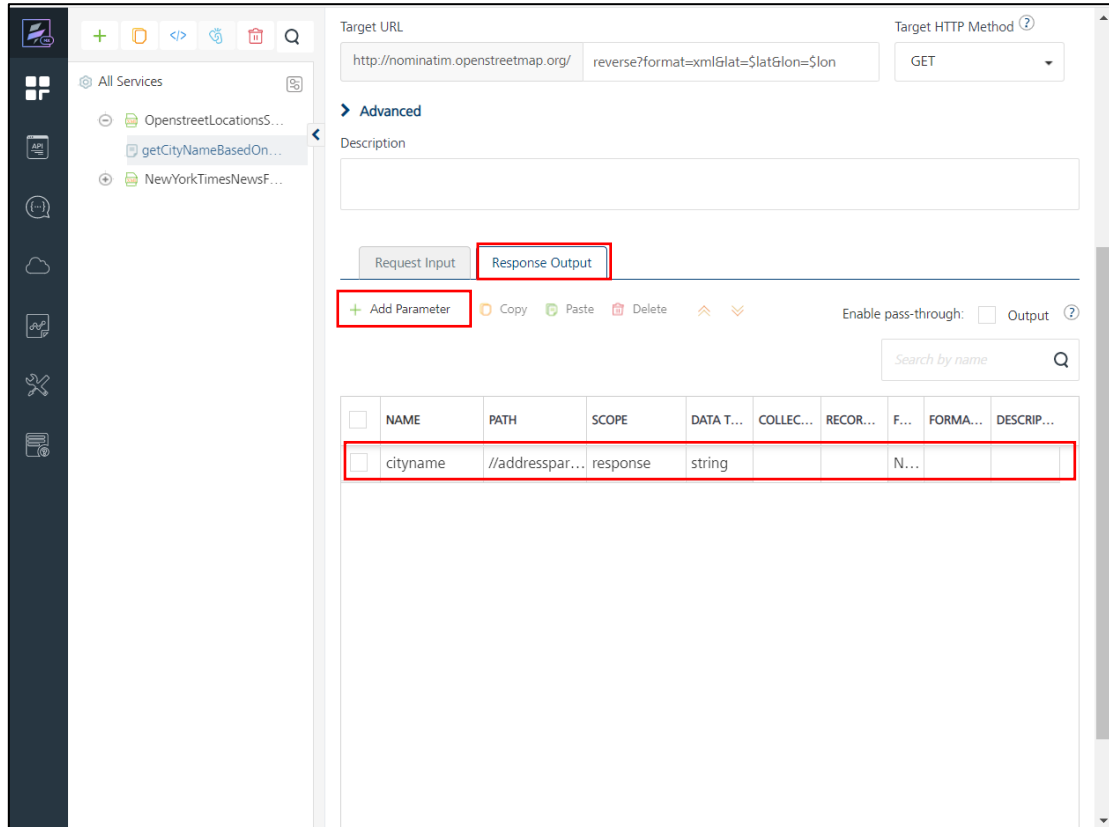
The screenshot shows the HCL Volt MX IDE interface for configuring a new operation. The 'Name' field is set to 'getCityNameBasedOnLocation'. The 'Target URL' is 'http://nominatim.openstreetmap.org/reverse?format=xml&lat=\$lat&lon=\$lon'. The 'Request Input' tab is active, and the 'Body' sub-tab is selected. The 'Add Parameter' button is highlighted. Below, a table lists parameters:

	NAME	VALUE	T...	D...	DATA TYPE	ENCOD...	D...
<input type="checkbox"/>	lat	request	4...	4...	string	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	lon	request	-1...	-1...	string	<input checked="" type="checkbox"/>	

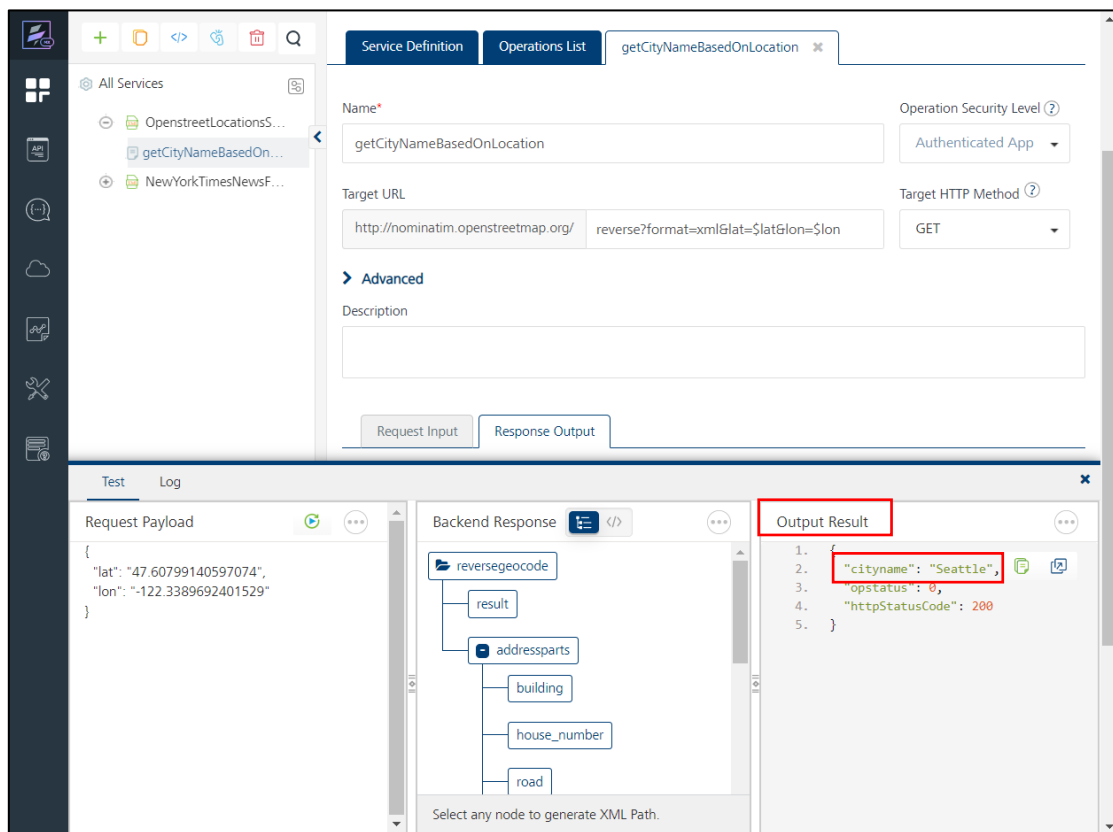
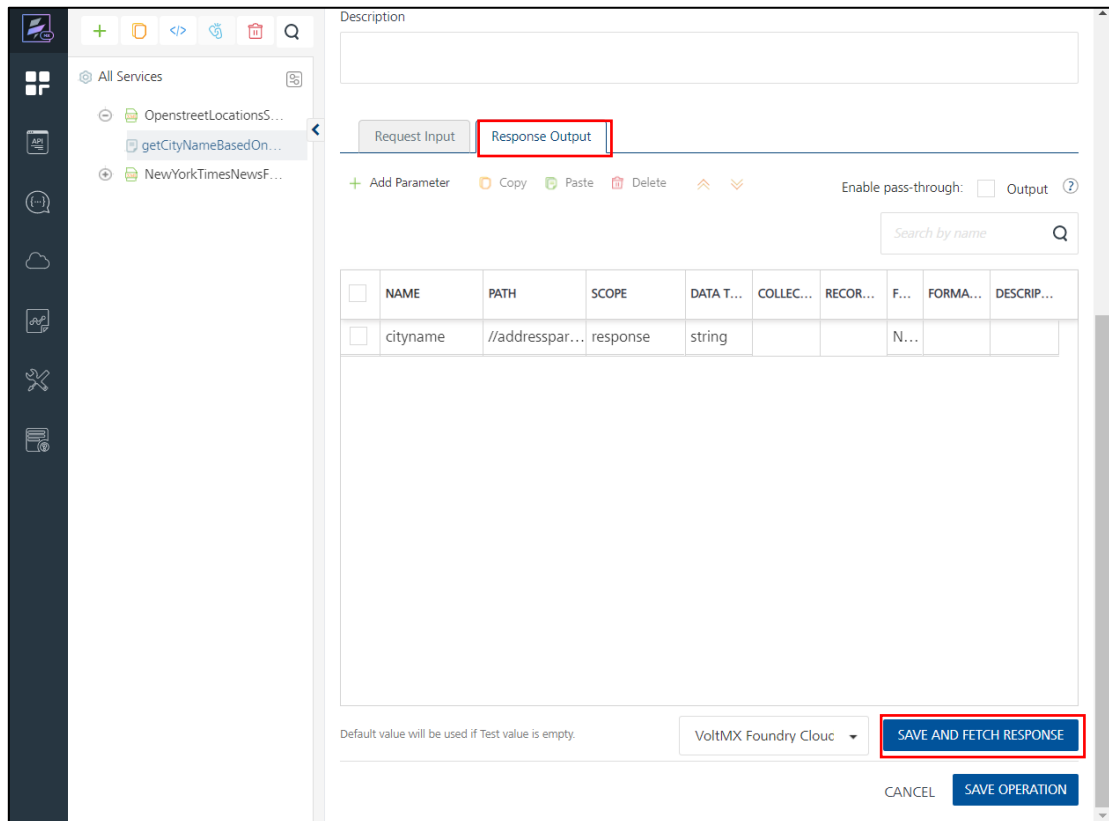
- **SAVE AND FETCH RESPONSE** をクリックします。
- **Request Payload** , **Backend Response** , **Output Result** を確認します。
- **Backend Response** では、OpenStreet から Foundry に送信されたレスポンスが表示されます。



- **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。
- **Add Parameter** をクリックします。
- **NAME** の値を **cityname** とします。
- **PATH** の値を **//addressparts/city** とします。

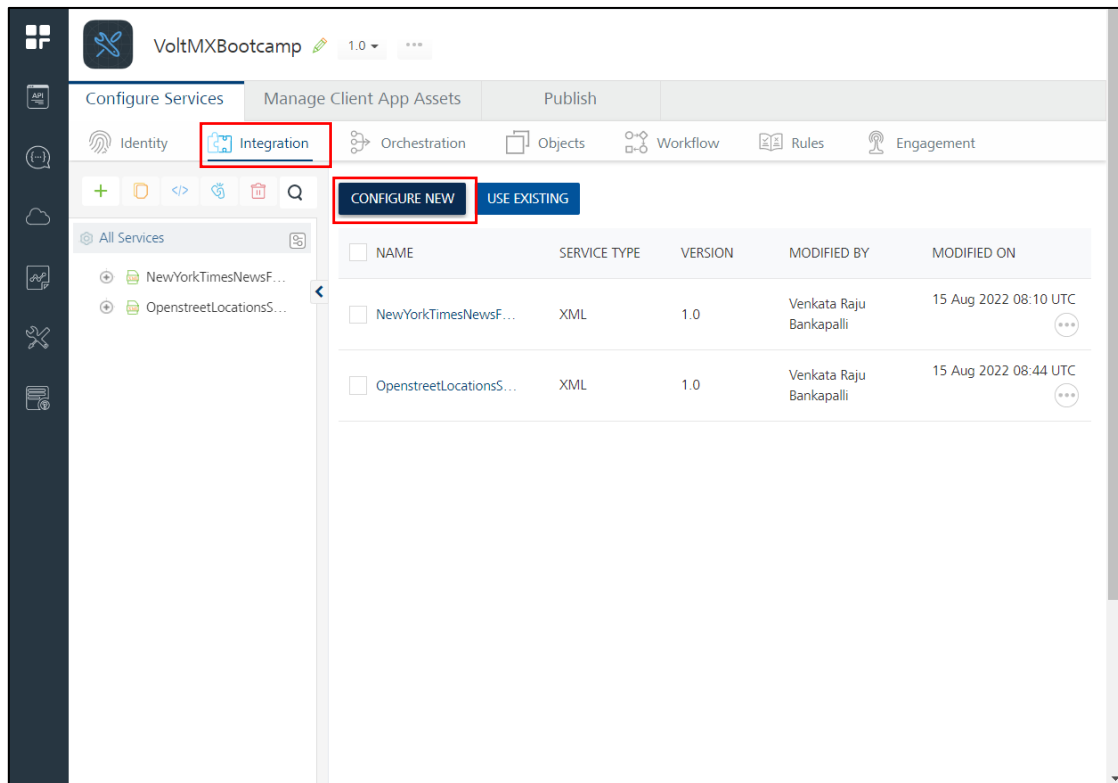


- **SAVE AND FETCH RESPONSE** をクリックします。
- **Request Payload** , **Backend Response** , **Output Result** を確認します。
- **Output Result** では、Foundry がフロントエンドアプリケーションに送信するレスポンスが表示されます。



都市の天気の詳細を取得するための Integration サービスを Foundry で構成する

- **Configure services > Integration** の下にある **CONFIGURE NEW** をクリックします。
- **Integration Service** の名前を **OpenweatherServices** とします。
- **Service Type** を **JSON** にします。
- **Base URL** の値として **https://api.openweathermap.org/data/2.5/** 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)
- NewYorkTimesNewsF...
- OpenstreetLocationsS...

Service Definition

Name* OpenweatherServices Service Type JSON Path Expression JSON Pat Version 1.0

Base URL* https://api.openweathermap.org/data/2.5/

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

- OpenweatherServices ...
- NewYorkTimesNewsF...
- OpenstreetLocationsS...

NewOperation

Name* NewOperation Operation Security Level Authenticated App

Target URL: https://api.openweathermap.org/data/2.5/ Enter Suffix Target HTTP Method GET

Advanced

Description

Request Input Response Output

Body Header

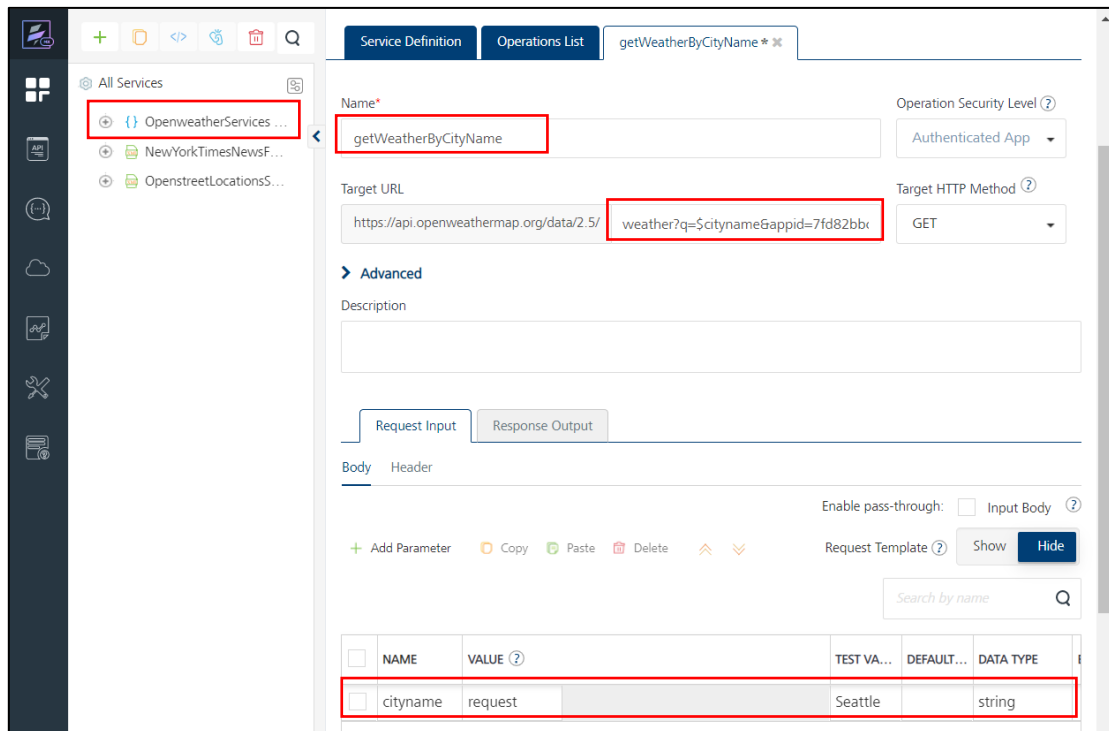
Enable pass-through: ☐ Input Body

+ Add Parameter Copy Paste Delete

Request Template Show Hide

Search by name

- **NewOperation** タブで、
 - **Name** の値を **getWeatherByCityName** とします。
 - **Target URL** に、サフィックスとして **weather?q=\$cityname&appid=7fd82bbcdf7fca658e257db4c6577f3** 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 **cityname** とします。
- **TEST VALUE** の値を **Seattle** とします。
- **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** では、Openweathermap から Foundry に送信されたレスポンスが表示されます。

The screenshot displays the HCL Volt MX Foundry interface for configuring and testing a service named 'getWeatherByCityName'.

Service Definition:

- Name:** getWeatherByCityName
- Operation Security Level:** Authenticated App
- Target URL:** https://api.openweathermap.org/data/2.5/weather?q=\$cityname&appid=7fd82bbx
- Target HTTP Method:** GET
- Description:** (Empty text area)

Advanced:

- Description:** (Empty text area)

Test Log:

- Request Payload:** { "cityname": "Seattle" }
- Backend Response:** A tree view showing the response structure: visibility, timezone, main (temp, temp_min, humidity, pressure, feels_like).
- Output Result:** 1. { 2. "opstatus": 0, 3. "httpStatusCode": 200 4. }

- **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、歯車のアイコン (Add All to Response Output) をクリックします。

現時点は応答出力はありません

	NAME	PATH	SCOPE	DATA T...	COLLEC...	RECOR...	F...	FORMA...	DESCRIP...
<input type="checkbox"/>	visibility	\$.visibility	response	number				N...	
<input type="checkbox"/>	timezone	\$.timezone	response	number				N...	
<input type="checkbox"/>	main	\$.main	response	record				N...	
<input type="checkbox"/>	temp	temp	response	number		main		N...	
<input type="checkbox"/>	temp_min	temp_min	response	number		main		N...	
<input type="checkbox"/>	humidity	humidity	response	number		main		N...	

- **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。

Request Payload

```
{
  "cityname": "Seattle"
}
```

Backend Response

```
visibility
timezone
main
  temp
  temp_min
  humidity
```

Output Result

```
1. {
2.   "visibility": 10000,
3.   "timezone": -25200,
4.   "main": {
5.     "temp": 288.59,
6.     "temp_min": 285.4,
7.     "humidity": 81,
8.     "pressure": 1018,
9.     "feels_like": 288.3,
10.    "temp_max": 291.34
11.  },
12.  "clouds": {
13.    "all": 0
14.  },
15.  "sys": {
```

Request Payload

```
{
  "cityname": "Seattle"
}
```

Backend Response

```
main
  temp
  temp_min
  humidity
```

Output Result

```
1. {
2.   "visibility": 10000,
3.   "timezone": -25200,
4.   "main": {
5.     "temp": 288.59,
6.     "temp_min": 285.4,
7.     "humidity": 81,
8.     "pressure": 1018,
9.     "feels_like": 288.3,
10.    "temp_max": 291.34
11.  },
12.  "clouds": {
13.    "all": 0
14.  },
15.  "sys": {
16.    "country": "US",
17.    "sunrise": 1660568724,
18.    "sunset": 1660620145,
19.    "id": 2041694,
20.    "type": 2
21.  },
22.  "dt": 1660574016,
23.  "coord": {
24.    "lon": -122.3321,
25.    "lat": 47.6062
26.  },
27.  "weather": [
28.    {
29.      "id": 800,
30.      "main": "Clear",
31.      "description": "clear sky",
32.      "icon": "01d"
33.    }
34.  ]
35.}
```

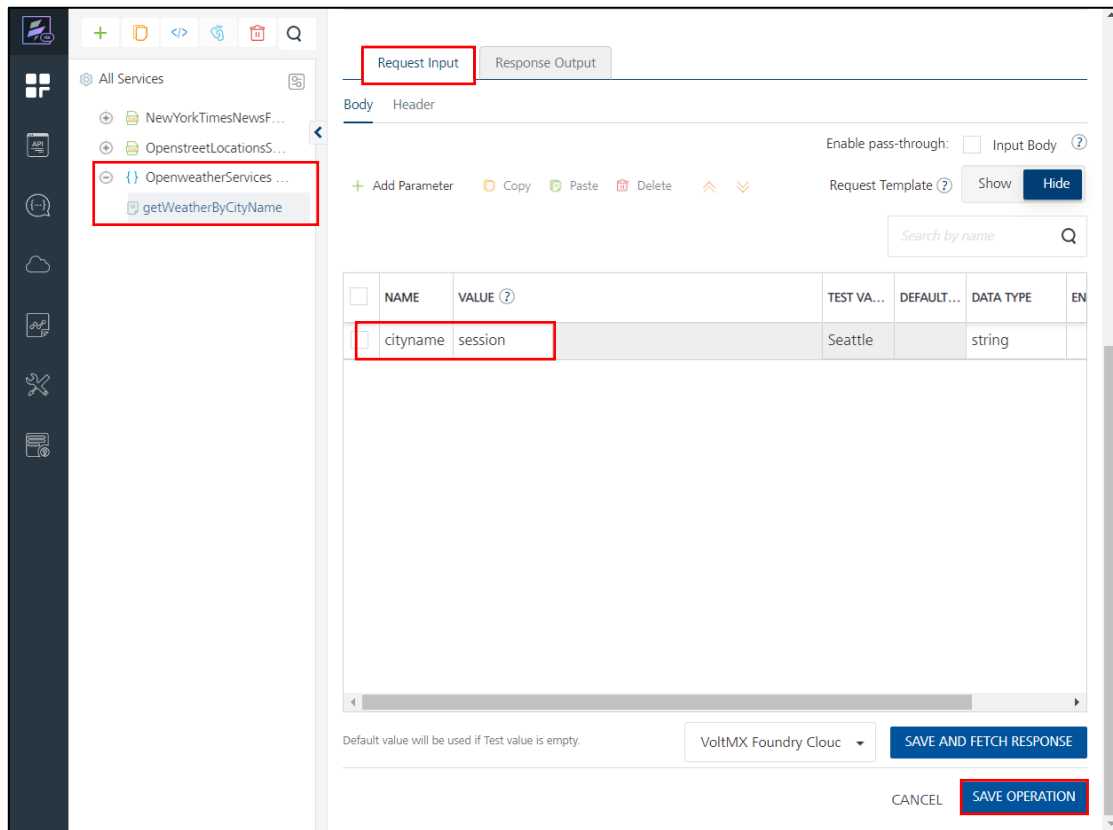
- **OpenstreetLocationsService > getCityNameBasedOnLocation Integration** サービスオペレーションを開いてください。
- レスpons出力で、**cityname** パラメータの **SCOPE** の値を **session** にします。 **SAVE OPERATION** をクリックします。

The screenshot shows the HCL Volt MX Foundry Cloud interface. On the left, a sidebar lists services, with 'OpenstreetLocationsService' and its operation 'getCityNameBasedOnLocation' highlighted by a red box. The main panel is titled 'Advanced' and shows the 'Response Output' configuration. A table lists the output parameters:

	NAME	PATH	SCOPE	DATA T...	COLLEC...	RECOR...	F...	FORMA...	DESCRIP...
<input type="checkbox"/>	cityname	//addresspar...	session	string			N...		

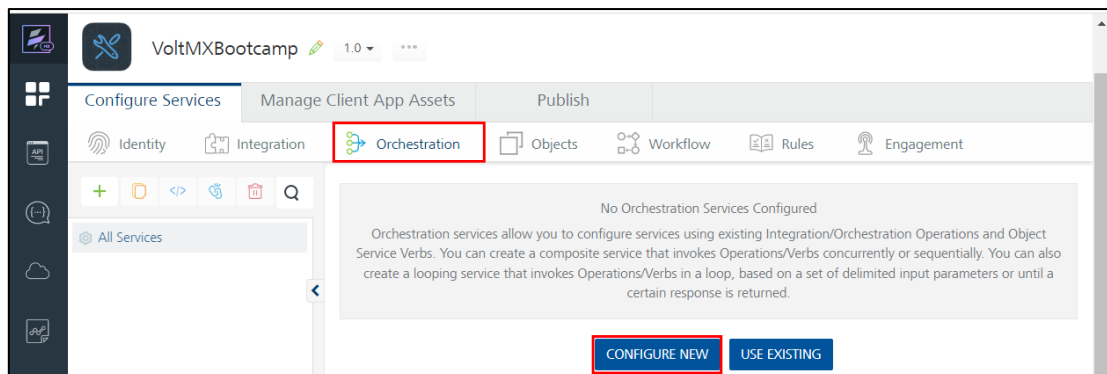
The 'cityname' parameter and its 'session' scope are highlighted with red boxes. At the bottom right, there is a 'SAVE AND FETCH RESPONSE' button.

- **OpenweatherServices > getWeatherByCityName Integration** サービス操作を開きます。
- リクエスト入力で、**cityname** パラメータの値を **VALUE** にして、**session** に設定します。**SAVE OPERATION** をクリックします。

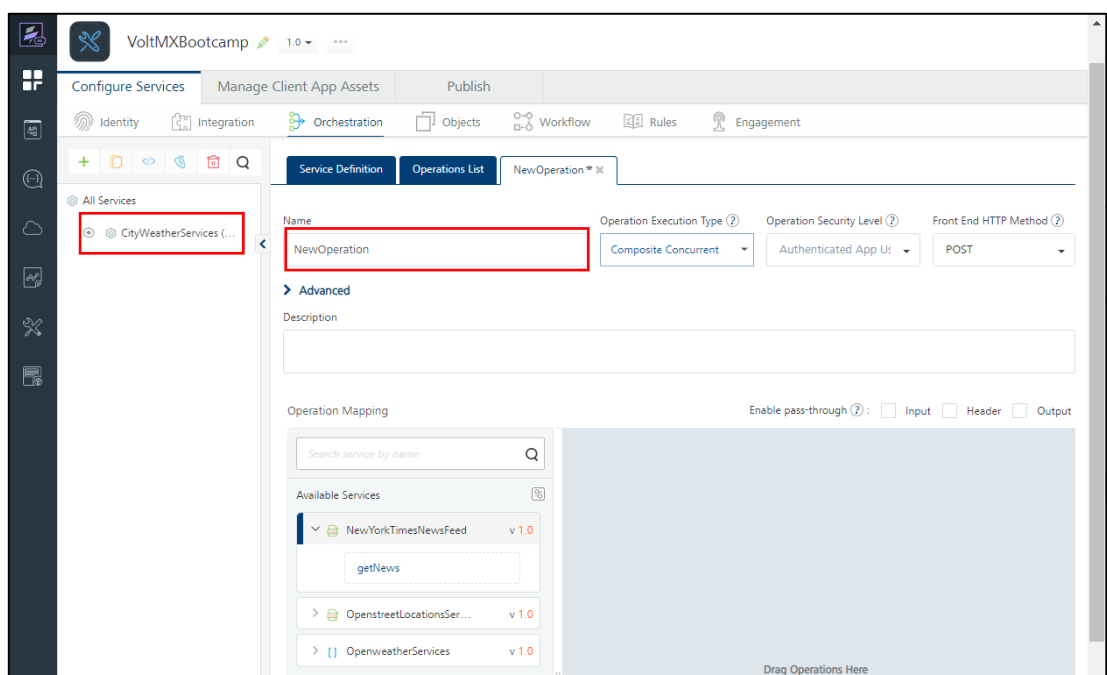
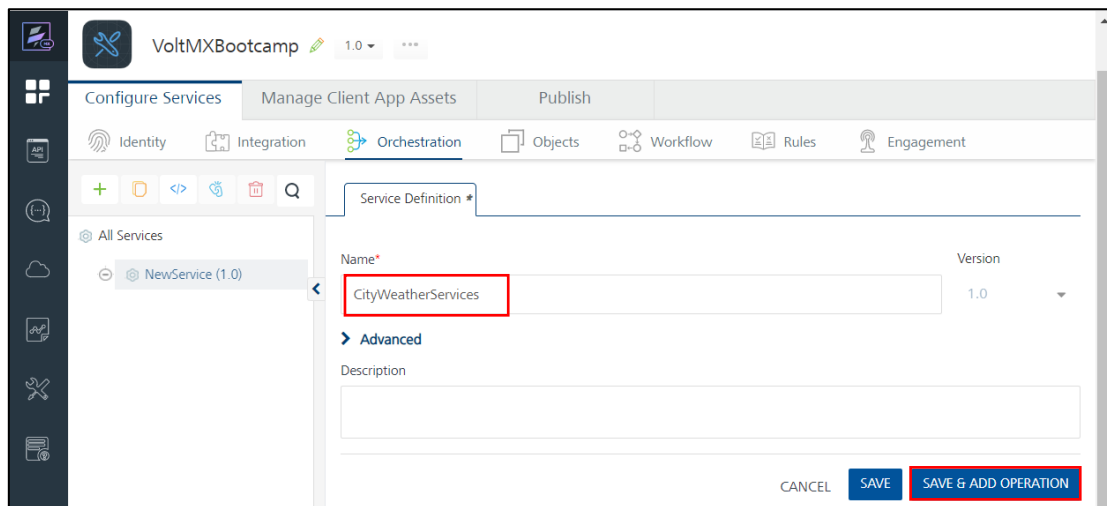


位置情報に基づいて天気の詳細を取得するための Foundry の Orchestration サービスの構成

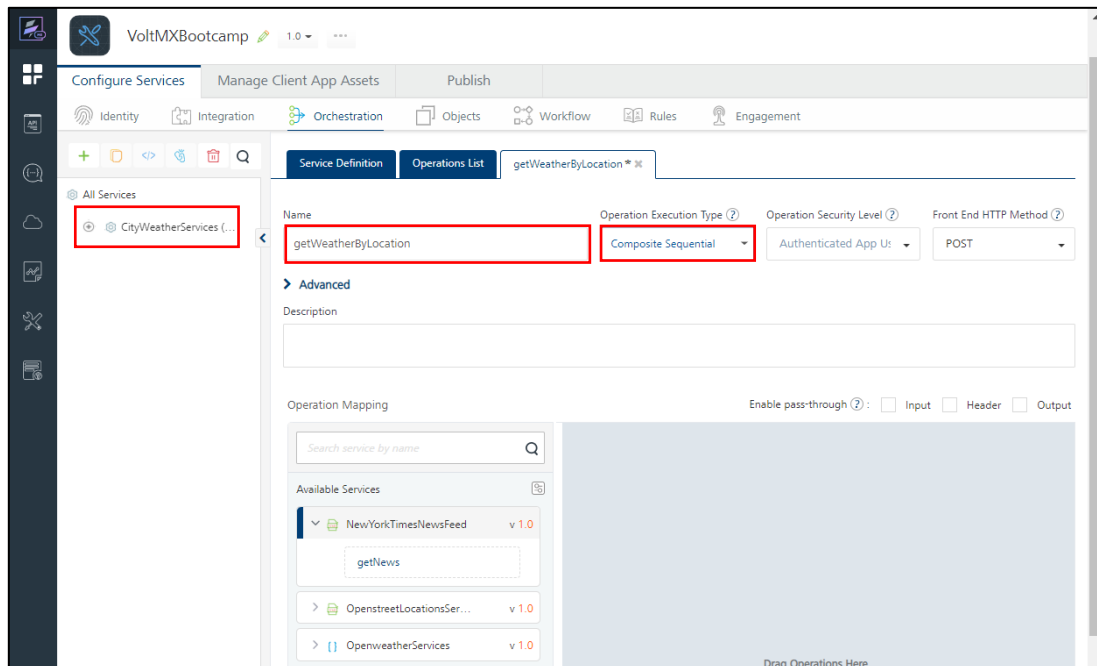
- **Configure services > Orchestration** の下にある **CONFIGURE NEW** をクリックします。



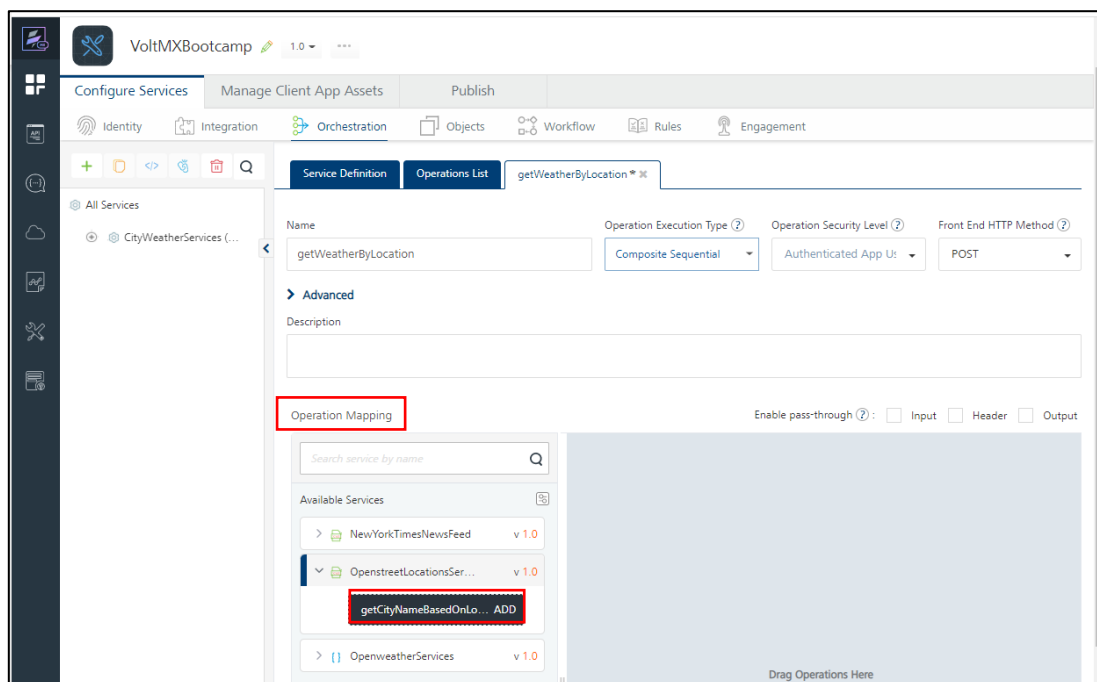
- **Name** を **CityWeatherServices** とします。 **SAVE & ADD OPERATION** をクリックします。 Orchestration サービスが作成されます。

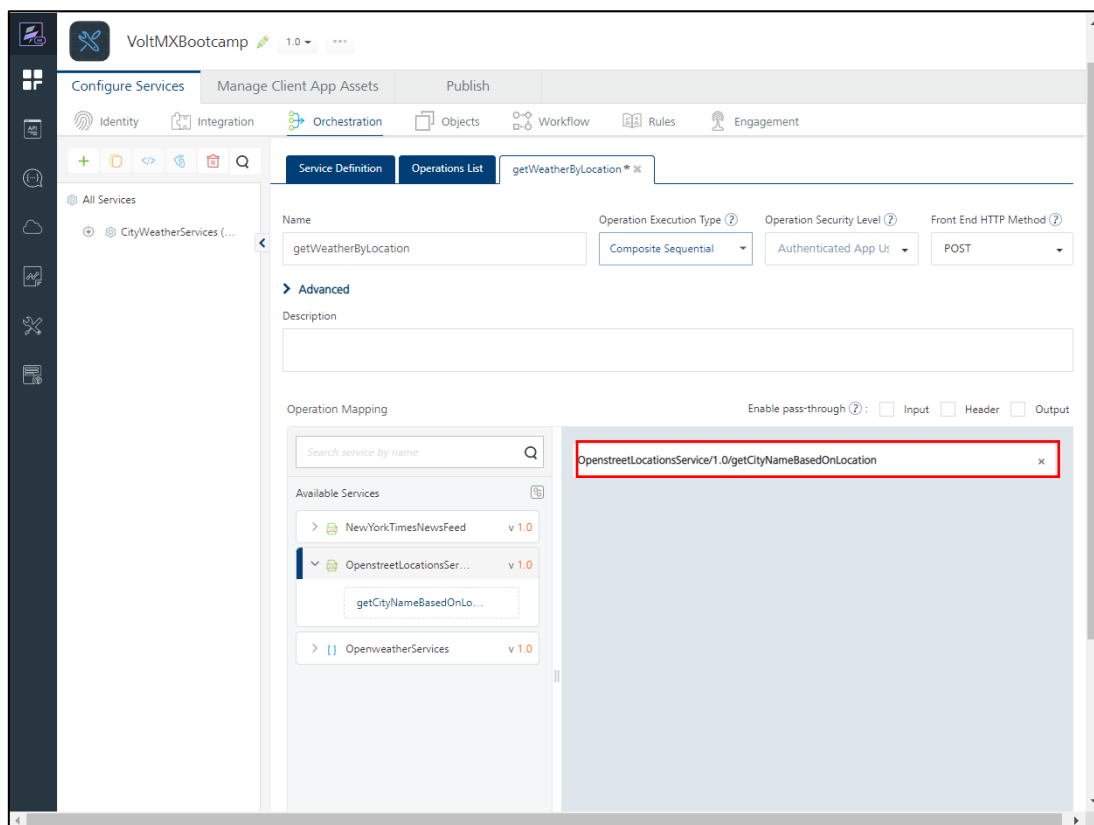


- **NewOperation** タブで、
 - **Name** の値を **getWeatherByLocation** とします。
 - **Operation Execution Type** の値を **Composite Sequential** に変更します。

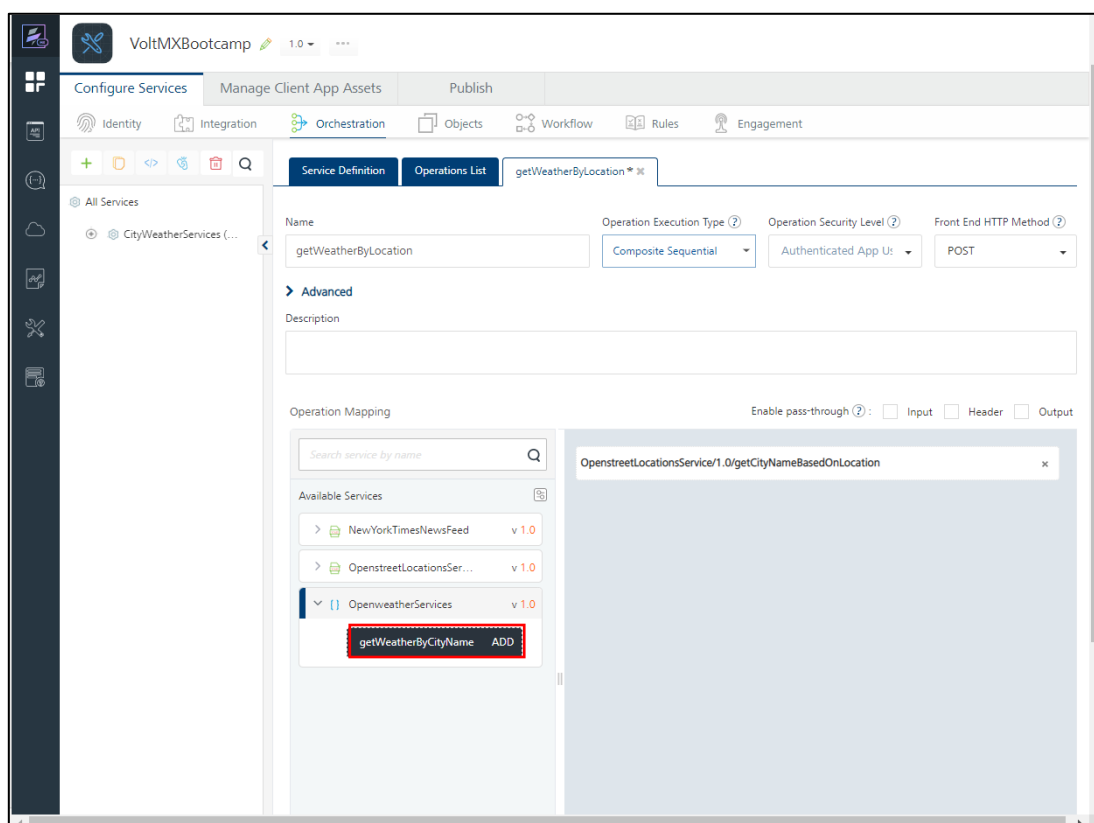


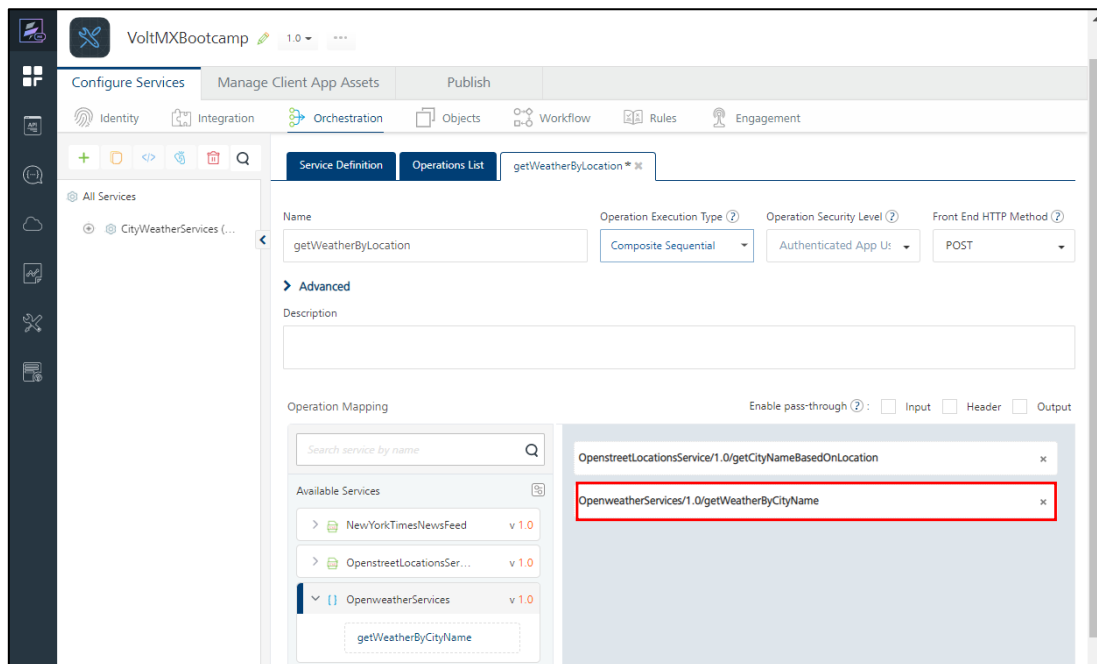
- Operation Mapping で、**OpenstreetLocationsService** > **getCityNameBasedOnLocation** サービス操作にカーソルを合わせて、**ADD** をクリックします。右側に追加されたことを確認します。



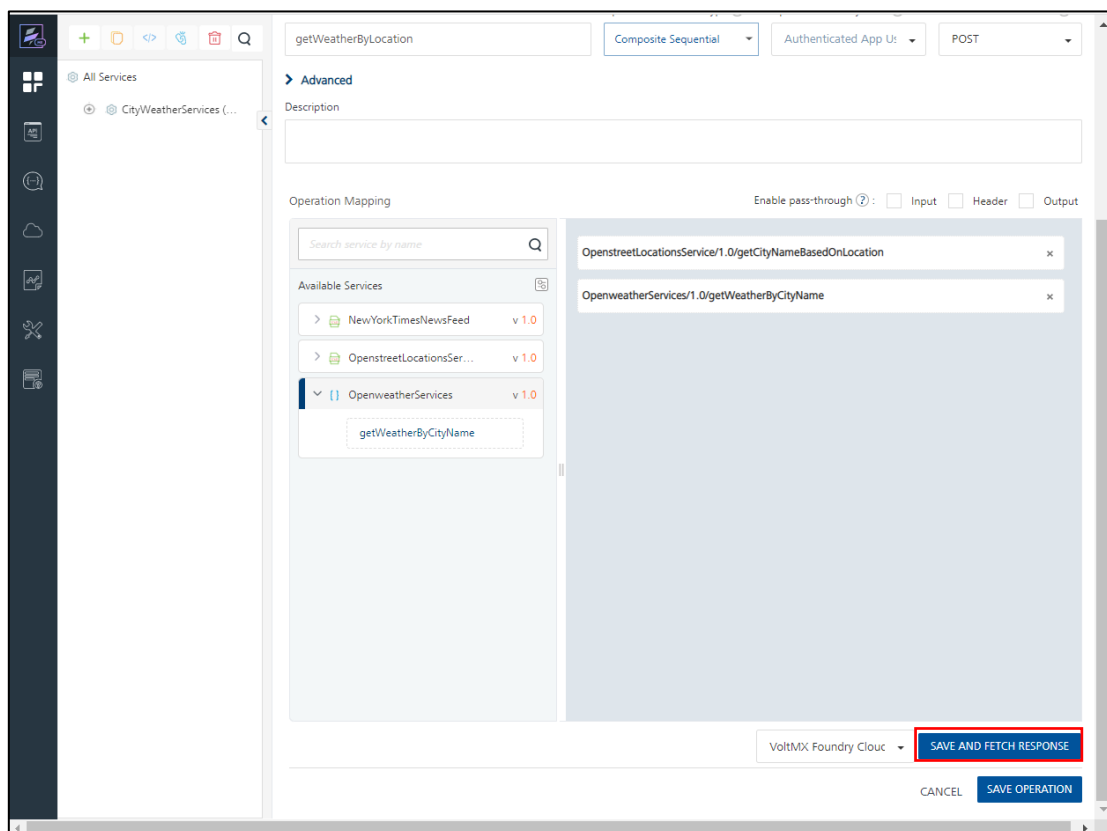


- **OpenweatherServices - getWeatherByCityName** サービスオペレーションにカーソルを合わせて、**ADD** をクリックします。右側に追加されたことを確認します。

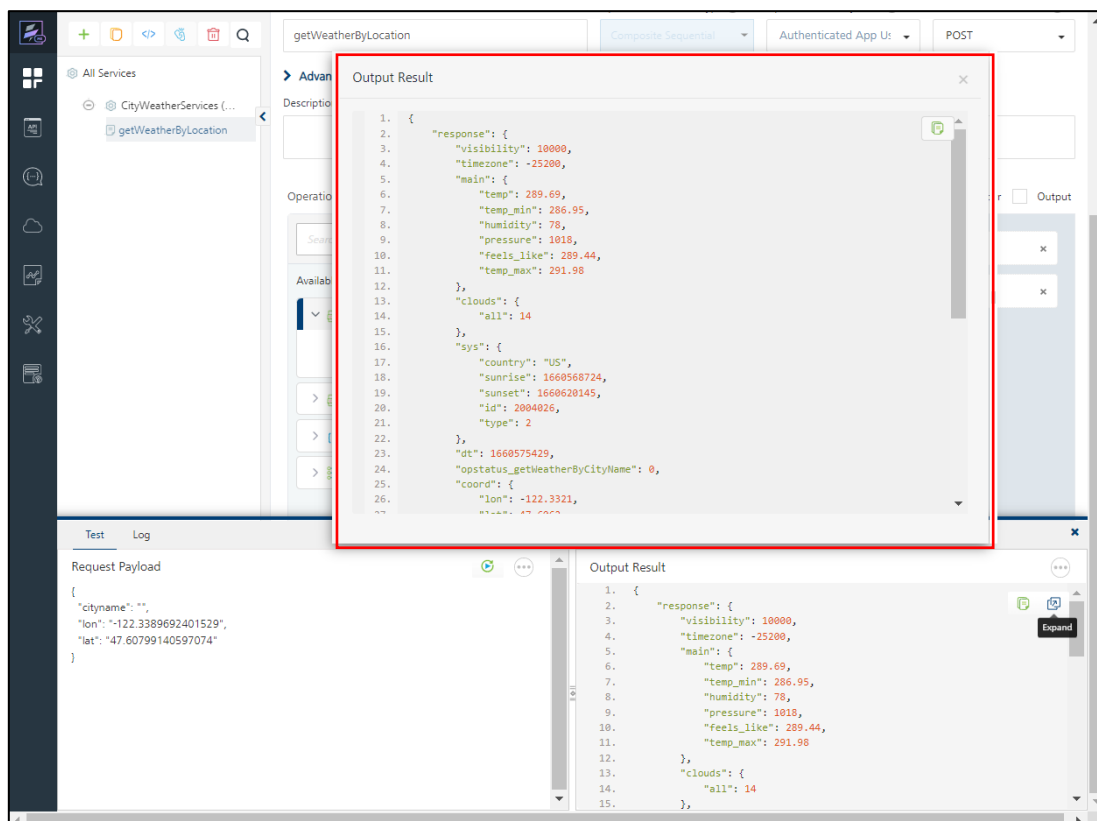
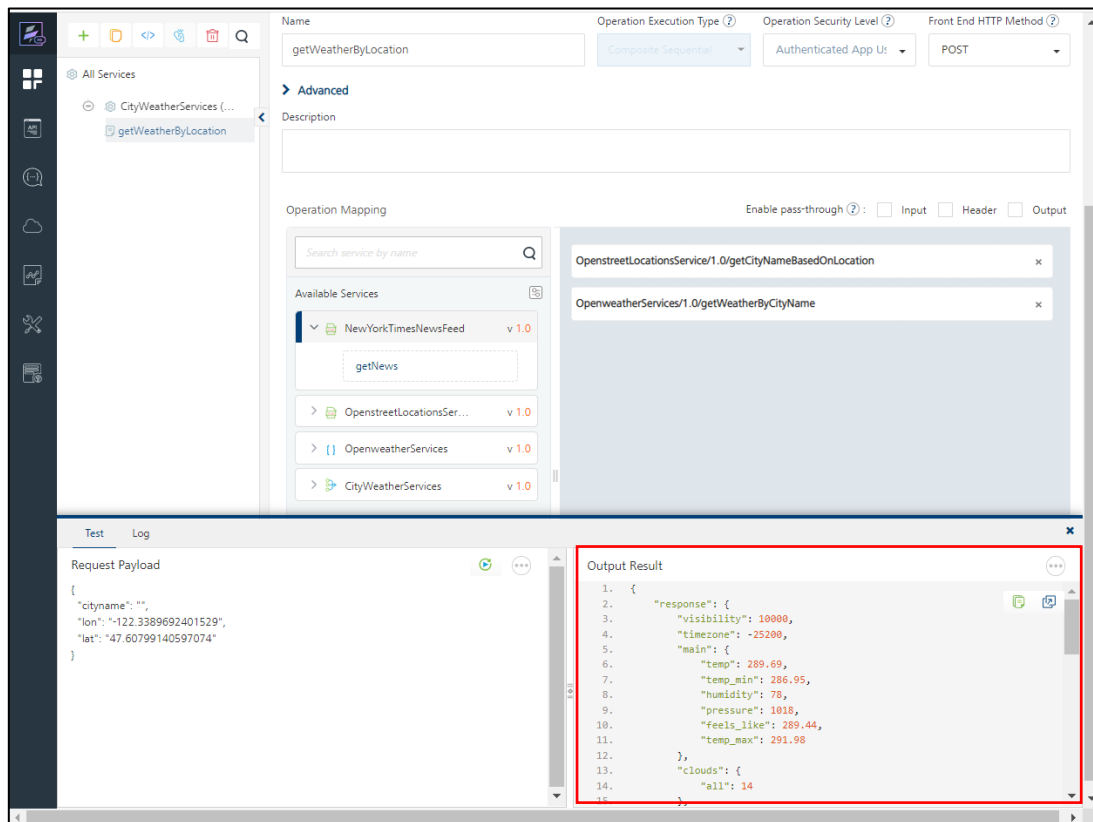




- **SAVE AND FETCH RESPONSE** をクリックします。

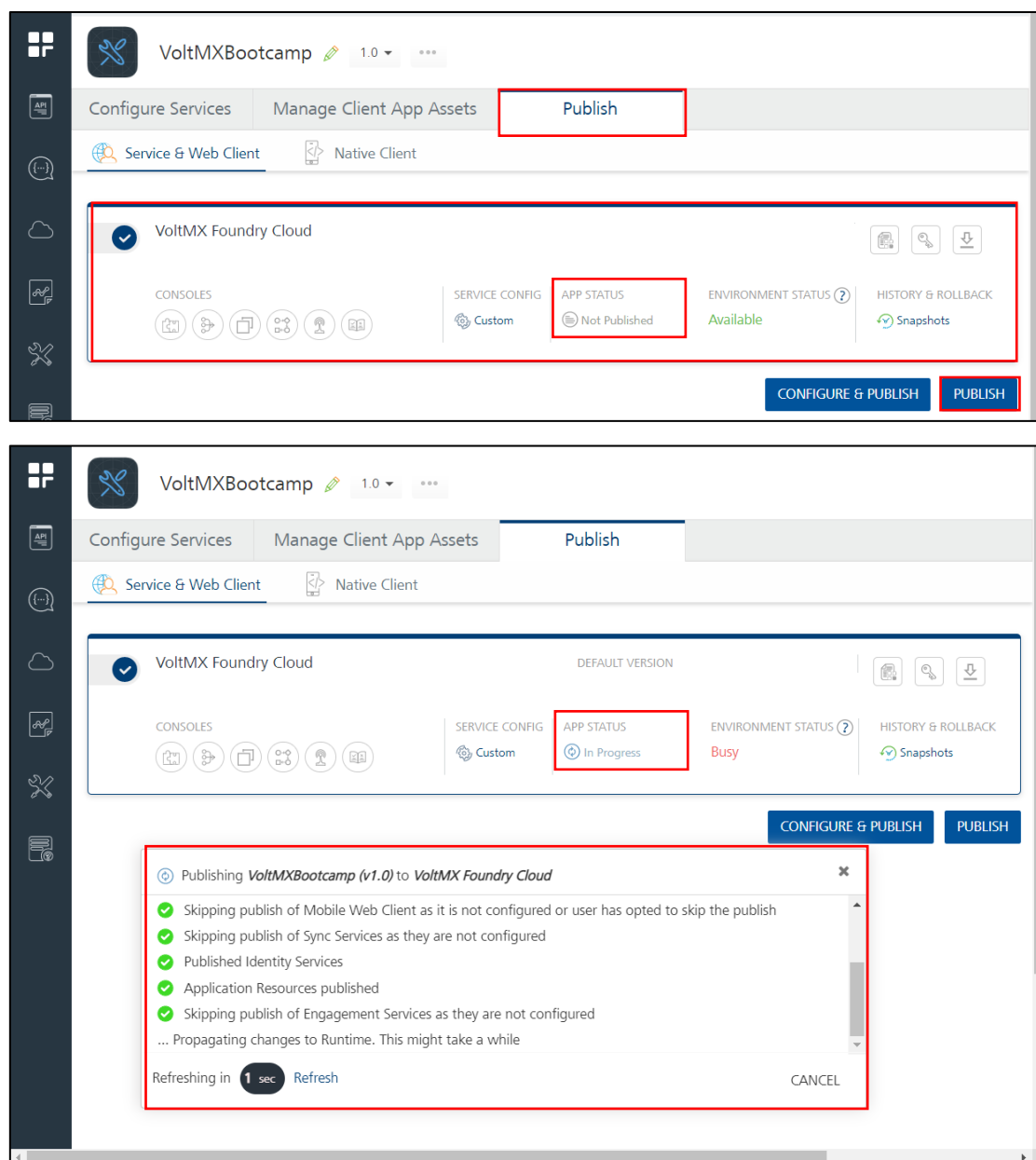


- **Request Payload** と **Output Result** を確認します。
- **Output Result** には、Foundry がフロントエンドアプリケーションに送信するレスポンスが表示されます。これには **getCityNameBasedOnLocation** と **getWeatherByCityName** 操作からのレスポンスが含まれます。

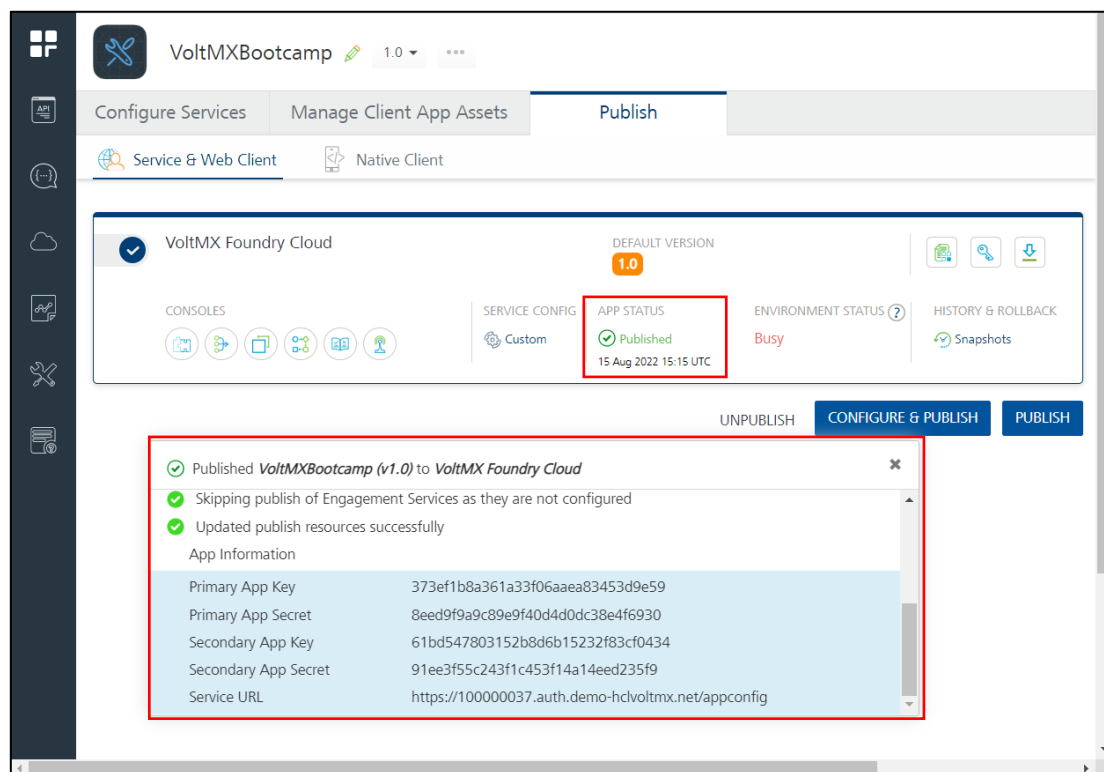


- **Publish** タブで、環境を選択し、**PUBLISH** をクリックします。Foundry アプリが公開されるのを待ちます。

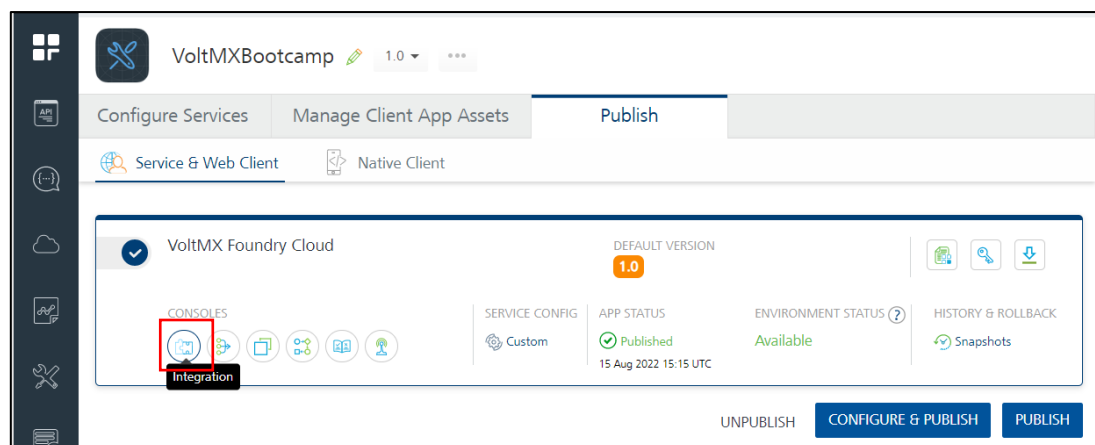
注：環境名は、**Foundry** の構成によって異なる場合があります。以下のスクリーンショットでは、**VoltMX Foundry Cloud** が、私たちの **Foundry** の構成で使用した環境名です。



- 公開に成功すると、Primary App Key、Primary App Secret などの詳細が表示されます。



- ランタイムコンソールを開き、サービスをテストします。CONSOLES の下にある最初の円をクリックします。



- ランタイムコンソールが別タブで開かれます。**OpenstreetLocationsService** のサービス名を検索します。

App Services

- Web Apps
- Monitoring
- Integration Services**
- Object Services
- Orchestration Services
- Workflow Services
- Rules Services
- Jobs
- Health Check
- Logs

Integration Services

Test any version of integration service and associated operations published to this environment. [More info](#)

OpenstreetLocationsService

Service Name	Service Type	Versions	Operations	Throttling
OpenstreetLocationsService	XML	1.0	Select Operation	{-}

<< < 1 > >>

- **OpenstreetLocationsService** で動作をテストします。

App Services

- Web Apps
- Monitoring
- Integration Services**
- Object Services
- Orchestration Services
- Workflow Services
- Rules Services
- Jobs
- Health Check
- Logs

Integration Services

Test any version of integration service and associated operations published to this environment. [More info](#)

OpenstreetLocationsService

Service Name	Service Type	Versions	Operations	Throttling
OpenstreetLocationsService	XML	1.0	Select Operation	{-}

getCityNameBasedOnLocation

< 1 > >>

App Services

- Web Apps
- Monitoring
- Integration Services**
- Object Services
- Orchestration Services
- Workflow Services
- Rules Services
- Jobs
- Health Check
- Logs
- Settings
- Reports
- Downloads

Integration Services

OpenstreetLocationsService > getCityNameBasedOnLocation

Request Input Response Output

Server URL

POST https://volt-mx-academy.demo-hclvoltmx.net:443/services/OpenstreetLocationsService/getCityNameBasedOnLocation

Target Backend URL

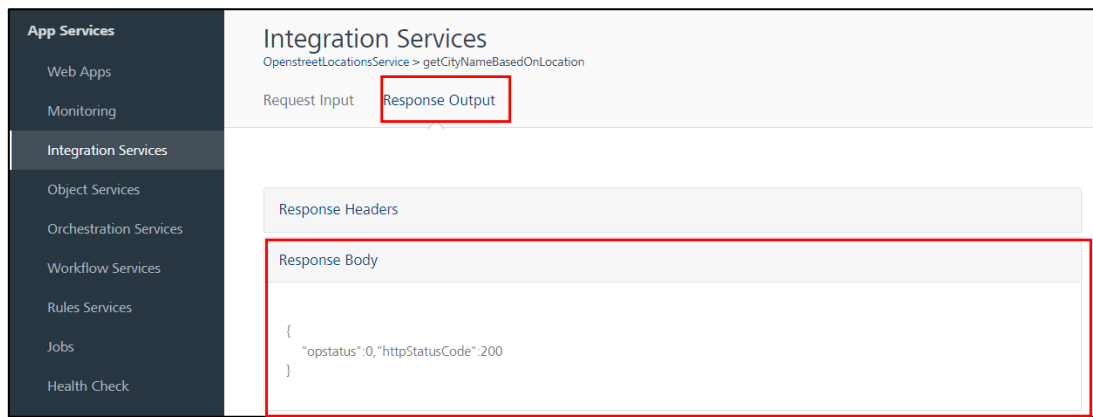
http://nominatim.openstreetmap.org/reverse?format=xml&lat=\$lat&lon=\$lon

Body Header

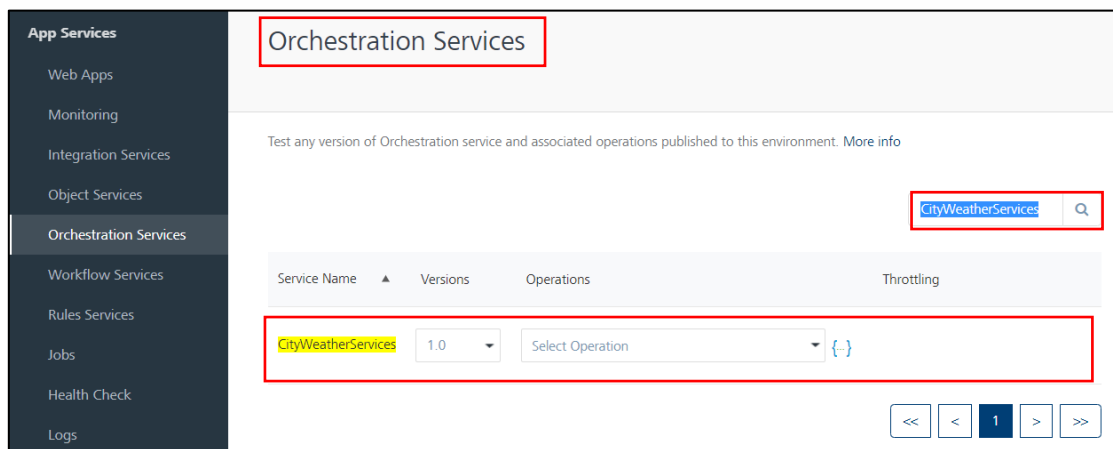
Param Name	Param Value
lat	47.60799140597074
lon	-122.3389692401529

Get Response

VERSION : V9 SP2 FP1 HF9



- **Orchestration Service** の **CityWeatherServices** をテストします。
- **Orchestration services** で **CityWeatherServices** を検索します。



- **getWeatherByLocation** オペレーションを開きます。 **Get Response** をクリックします。



App Services

- Web Apps
- Monitoring
- Integration Services
- Object Services
- Orchestration Services**
- Workflow Services
- Rules Services
- Jobs
- Health Check
- Logs
- Settings
- Reports
- Downloads

Orchestration Services
CityWeatherServices > getWeatherByLocation (CompositeConnector)

Request Input Response Output

POST https://volt-mx-academy.demo-hclvoltmx.net:443/services/CityWeatherServices/getWeatherByLocation

Body Header

Param Name	Param Value	Service(s) Using This Parameter
lon	-122.3389692401529	getCityNameBasedOnLocation
cityname	Value retrieved from session	getWeatherByCityName
lat	47.60799140597074	getCityNameBasedOnLocation

Get Response

VERSION : V9 SP2 FP1 HF9

- オーケストレーションされた両方のサービスからの応答が含まれていることを確認します。

App Services

- Web Apps
- Monitoring
- Integration Services
- Object Services
- Orchestration Services**
- Workflow Services
- Rules Services
- Jobs
- Health Check
- Logs
- Settings
- Reports
- Downloads

Orchestration Services
CityWeatherServices > getWeatherByLocation (CompositeConnector)

Request Input Response Output

Response Headers

Response Body

```
{
  "visibility":10000,"timezone":-25200,"main":{
    "temp":290.37,"temp_min":287.07,"humidity":.77,"pressure":1018,"feels_like":290.16,"temp_max":293.25
  },
  "clouds":{
    "all":0
  },
  "sys":{
    "country":"US",
    "sunrise":1660568724,"sunset":1660620145,"id":2004026,"type":2
  },
  "dt":1660577223,"opstatus_getWeatherByCityName":0,"coord":{
    "lon":-122.3321,"lat":47.6062
  },
  "name":"Seattle",
  "weather":[
    {
      "icon":"01d",
      "description":"clear sky"
    }
  ]
}
```

VERSION : V9 SP2 FP1 HF9

おめでとうございます。このレッスンのハンズオンが完了しました。