



HCL Volt MX

Foundry アプリケーションの 開発

データベースとオブジェクトサービス

データベースアプリケーションの作成

- このモジュールは、HVMX-FOU-301 と HVMX-FOU-302 Foundry 301 と 302 のレッスンの延長線上にあるものです。
- これらのレッスンでは、簡単なXMLとJSONサービスを呼び出す方法を学びました。
- また、プリプロセッサとポストプロセッサについても学びました。
- では、他のWebサービスの代替としてデータベースを使用するアプリケーションを作成しましょう。
- さらに、データベース統合サービスを使用して、オブジェクトサービスを作成します。

リレーショナルデータベースアダプター






















- Volt MX Foundry データベースアダプターを使用すると、独自のデータベースをエンドポイントとして接続できます。
- Volt MX Foundry Consoleでデータベースアダプターを設定すると、テーブルのデータに対して作成、読み込み、更新、削除(CRUDおよびバイナリCRUD)操作を実行し、ストアドプロシージャ、関数、およびビューを呼び出すことができるようになります。

Volt MX Foundry Databaseアダプターの利点

- Volt MX Foundry Database Adapter を使用する利点は以下のとおりです。
 - 管理者は、指定されたデータベースに接続できます。
 - 管理者は、CRUD操作を使用してデータベースを管理できます。
 - すべての主要なデータ型に対応しています。
 - Volt MXは、読み取り操作用に\$filter、\$orderby、\$top、\$skip、\$select、\$expand、\$batchsize、および\$batchidの8つのODATAパラメータをサポートしています。
 - Volt MXは、Delete操作のためのODATAパラメーター\$filterをサポートしています。
 - Volt MXは、ODATAメソッド:substringofおよびindexofをサポートしています。
 - 複数のテーブルに対してJoinのような単一の操作を行う場合、カスタムSQLクエリを実行できます。
- 詳しくはドキュメントを参照してください。

データベース統合サービス

- PostgreSQL、MySQL、SQL Server、Oracleのリレーショナルデータベースをサポートしています。

Technology Adapters		Business Adapters		
 XML		 MuleSoft	 IBM MQ	 Red Hat PAM
 SOAP		 AWS API Gateway	 SAP JCO	
 JSON		 Relational Database	 Email	
 Java		 MongoDB	 AWS S3	
 JavaScript		 RAML	 Cloudian	
 API Proxy		 OpenAPI (Swagger)	 Foundry Admin	
 Mock Data		 Salesforce		

- XML/JSON統合サービスとは異なり、Foundryは入出力パラメータを自動的に読み込みます。

データベース接続のプロパティ

- 統合サービスを作成するには、次のものがが必要です。

- Name (統合サービスの名前)
- データベース
- データベース接続URL
- ユーザーID
- パスワード
- 必要に応じて、オプションのパラメータを指定します。

- 注: データベースを選択すると、URL接続文字列のフォーマットが表示されます。

Enter or Select JDBC Class

PostgreSQL

MySQL

SQL Server

Oracle

Database Connection URL* ?

syntax : jdbc:<type of Database>://<ip_address>:

<port>Examples :

jdbc:oracle:thin:@10.115.192.141:1521:sid

jdbc:mysql://10.115.192.141:3036

jdbc:sqlserver://10.115.192.141:1433

jdbc:postgresql://10.115.192.141:5432

- ? アイコン を選択して追加のヘルプを参照できます。
- テスト接続を使用して、値を検証します。
- 操作を追加する前に、保存することを忘れないでください。

TEST CONNECTION

Connection Successful

All Services

myStoreDBSvc (1.0)

Service Definition

Operations List

Name*

myStoreDBSvc

Service Type

Relational Database

Version

1.0

Connection Parameters

Database*

com.mysql.jdbc.Driver

Database Connection URL* ?

jdbc:mysql://localhost:3306/mystore

User ID ?

volt

Password ?

.....

Advanced settings ?

Max Pool Size ?

Connection Timeout (ms) ?

hclvoltmx

TEST CONNECTION

Authentication

☒ Use Existing Identity Provider

None

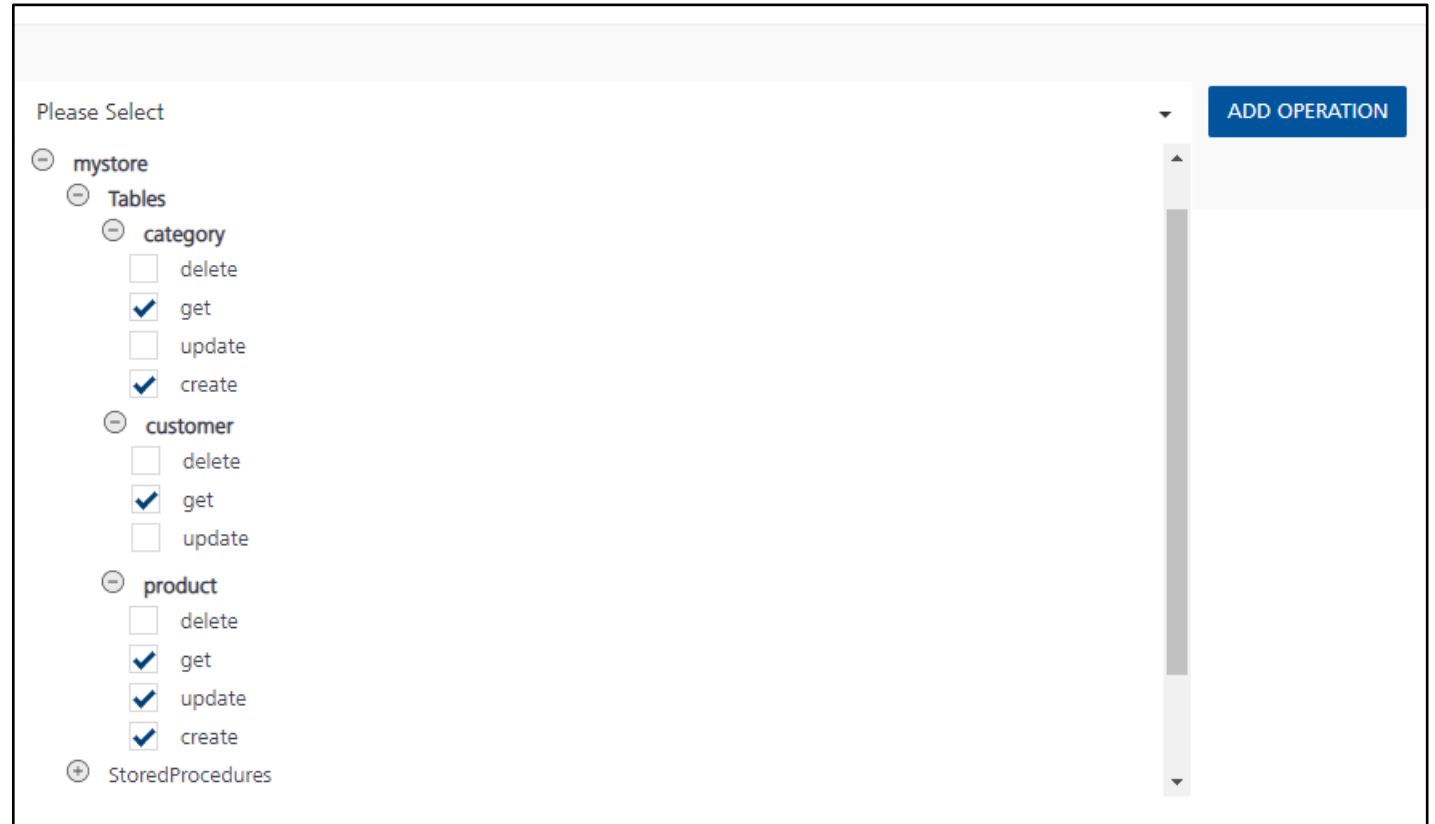
☐ Use proxy from settings

> Advanced

Description

データベース操作








- オペレーションを追加するために、Operations List タブを選択します（または、ServicesタブのSave and Add Operation ボタン）。
- ドロップダウンを展開します。



- 必要なテーブルとCRUD操作を選択し、操作を追加します。

選択した操作

- サンプルアプリケーションで使用する操作は次のとおりです。
 - Category: Get/Create
 - Customer: Get
 - Product: Get/Create/Update

Configured Operations				<input type="text" value="search"/>	
<input type="checkbox"/>	NAME ▼	MODIFIED BY	MODIFIED ON		
<input type="checkbox"/>	mystore_category_create	volt mx	11 Mar 2022 00:40 UTC		
<input type="checkbox"/>	mystore_category_get	volt mx	11 Mar 2022 00:40 UTC		
<input type="checkbox"/>	mystore_customer_get	volt mx	11 Mar 2022 00:42 UTC		
<input type="checkbox"/>	mystore_product_create	volt mx	11 Mar 2022 00:47 UTC		
<input type="checkbox"/>	mystore_product_get	volt mx	11 Mar 2022 00:40 UTC		
<input type="checkbox"/>	mystore_product_update	volt mx	11 Mar 2022 00:40 UTC		

データベースの操作

- ご覧のように、Foundryはデータベースのプロパティを読み取ることができ、すでに出力パラメータを定義しています。
- 取得操作のSAVEとFETCH RESPONSEをクリックすると、結果が表示されます。

The screenshot displays the Foundry interface for configuring and testing a database service. On the left, a sidebar lists services under 'All Services', including 'StoreDBSvc (1.0)' and 'myStoreDBSvc (1.0)'. The 'myStoreDBSvc (1.0)' service is selected, showing its configuration in the main panel. The 'Mapped To' field is set to 'mystore.product' and the 'Action' is 'get'. The 'Advanced' tab is active, showing the 'Response Output' section. Below this, a table lists the response parameters:

	NAME	PATH	SCOPE	DATA TYPE	COLLECTION ID	RECORD ID	DESCRIPTION
<input type="checkbox"/>	product		response	collection			
<input type="checkbox"/>	product_id		response	number	product		
<input type="checkbox"/>	product_desc		response	string	product		
<input type="checkbox"/>	product_image		response	string	product		

Below the table, the 'Test' tab is active, showing the 'Request Payload' as an empty object `{}`. The 'Output Result' section displays the JSON response from the test:

```
1. {
2.   "product": [
3.     {
4.       "product_desc": "Lego Star Wars 1000 pc set",
5.       "quantity": "12",
6.       "category_id": "1",
7.       "price": "129.99",
8.       "product_image": "https://i.picsum.photos/id/11/50/50",
9.       "product_id": "1"
10.    },
11.  ]
}
```

データベースとOData

- データベース接続では、OData を使用して結果をフィルタリングします。

For example (sample employee table), shown below:

Command Name	Test value for the command	Result
\$filter	emp_Id ge 30	Filters and displays data in the table based on age of employees who are older than 30.
\$orderby	emp_Age	Arranges data in the table based on employees' age.
\$top	5	Displays top five records in the table.
\$skip	5	Displays all records in the table except top five records.

For example (sample configuration for ODATA commands), shown below:

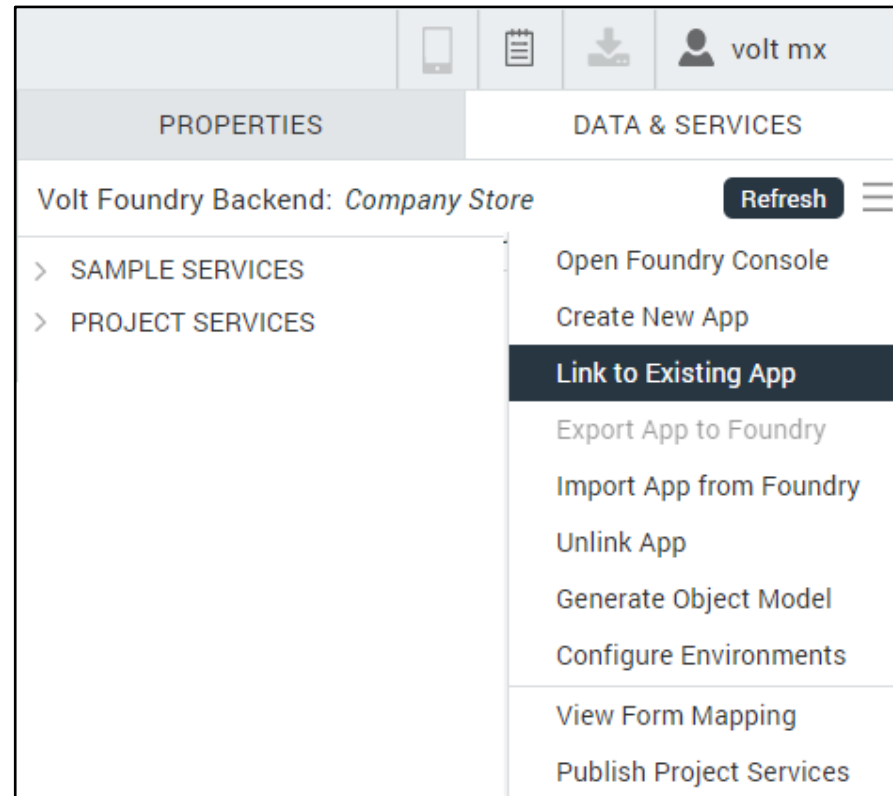
<input type="checkbox"/>	NAME	VALUE ?	TEST VALUE
<input type="checkbox"/>	\$filter	request	emp_Id ge 30
<input type="checkbox"/>	\$orderby	request	emp_Age
<input type="checkbox"/>	\$top	request	5

演習 - Company Store (ステップ 1 - データベース)

- Foundryアプリを作成する
- 統合データベースサービスを作成する
- 6つのオペレーションを作成する
 - Category: get/update
 - Product: get/create/update
 - Customer: get
- テストと公開

Iris でのデータベース統合サービスの呼び出し

- Iris でデータベース統合サービスを呼び出すのは、XML/JSON/Swagger サービスを呼び出すのと同じです。
- まず、Iris アプリケーションが Foundry アプリケーションに接続されていることを確認する必要があります。



コードでデータベース統合サービスを呼び出す

- 他の統合サービスと同様に、Foundry SDK を使用して、Iris アプリケーションからサービスを呼び出せます。
- 詳細はドキュメントに記載されています。
 - https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Foundry/voltmx_foundry_user_guide/Content/Relational_Database_Adapter.htm

オペレーションシグネチャの呼び出し

- ドキュメントを確認すると、この呼び出しのシグネチャは次のようになっています。
- `integrationSvc.invokeOperation(operationName, headers, params, successCallback, failureCallback, options)`
- これを詳しく見てみましょう。

データベース統合サービス - コードでの使用

```
1.  var serviceName = "integration_service_name";
2.  // Get an instance of SDK
3.  var client = voltmx.sdk.getCurrentInstance();
4.  var integrationSvc = client.getIntegrationService(serviceName);
5.  var operationName = "operation_name";
6.  var params = {
7.      "custom-input-key1": "custom-input-value1"
8.  };
9.  var headers = {
10.      "custom-header-key1": "custom-header-value1"
11.  };
12.  var options = {
13.      "httpRequestOptions": {
14.          "timeoutIntervalForRequest": 60,
15.          "timeoutIntervalForResource": 600
16.      }
17.  };
18.  integrationSvc.invokeOperation(operationName, headers, params, function(response) {
19.      voltmx.print("Integration Service Response is: " + JSON.stringify(response));
20.  }, function(error) {
21.      voltmx.print("Integration Service Failure:" + JSON.stringify(error));
22.  }, options);
```

データベース統合サービス - getCategory

```
1.  getCategoryes : function ()
2.      {
3.          var serviceName = "StoreDBSvc";
4.          // Get an instance of SDK
5.          var client = voltmx.sdk.getCurrentInstance();
6.          var integrationSvc = client.getIntegrationService(serviceName);
7.          var operationName = "mystore_category_get";
8.          var params = {};
9.          var headers = {};
10.         var options = { "httpRequestOptions": { "timeoutIntervalForRequest": 60, "timeoutIntervalForResource": 600 } };
11.         integrationSvc.invokeOperation(operationName, headers, params, successCallback.bind(this), errorCallback.bind(this),
options);
12.         function successCallback (response)
13.             {
14.                 this.view.segCategory.widgetDataMap = {lblCatName : "categoryname", lblCatDesc:"categorydesc"};
15.                 this.view.segCategory.setData(response.category);
16.             }
17.         function errorCallback (error)
18.             {
19.                 alert("Integration Service Failure:" + JSON.stringify(error));
20.             }
21.     }
```

パラメーターの送信

- 先ほど見たように、invoke 操作のパラメータのひとつは "params" オブジェクトです。
- params オブジェクトは、操作に必要なパラメータのキーと値のコレクションで、キーはカラム、値はカラムに挿入または更新する内容です。
- これらは Foundry が作成した入力パラメータに対応します。

gets のパラメータ

- 前述のとおり、params オブジェクトは Foundry が作成した入力パラメータに対応します。
- これは、OData パラメータも同様に可能です。

- 例えば、以下のようになります。

```
var params = {  
    // "custom-input-key1": "custom-input-value1"  
    "$filter": "'catid' eq '" + rowData.catid + "'"  
};
```

- これは、catid = 選択されたカテゴリーID というフィルタを作成します。

作成時のパラメータ

- 主キーカラムを持つテーブルへの挿入では、他のすべてのカラムに値を指定する必要があります。
- 例えば、product_insertサービスのパラメータには、以下のようなものがあります。

```
var params = {  
  "catid": this.view.lstCategory.selectedKey,  
  "desc": this.view.txtDesc.text,  
  "amount": this.view.txtAmount.text,  
  "quantity" : this.view.txtQty.text  
};
```

- 注: 挿入に成功すると、主キーと params オブジェクトのすべてのカラムが返されます。

更新のためのパラメータ

- 更新のために、データベースが更新する行を知ることができるように、主キーカラムを提供する必要があります。

- 例:

```
var params = {  
    "productid": this.view.segProducts.selectedRowItems[0].productid,  
    "cost": this.view.segProducts.selectedRowItems[0].price,  
    "quantity": this.view.txtQuantity.text  
};
```

- 注:更新されない場合は、すべての列を指定する必要はありません。Foundryはパラメータの一部である列のみを返します。

演習 - 会社の店舗

- ウェブアプリを作成する
- Category テーブルのデータ構造を見る
- 簡単なUIを設計する。
 - セグメントまたはリストボックスにカテゴリーのリストを表示する。
 - 新しいカテゴリーを追加する
- 完了したら、Product テーブルのデータ構造を確認する。
- 簡単なUIを設計する。
 - 選択したカテゴリーに関連する商品のリストをセグメントボックスまたはリストボックスに表示する。
 - 新しい商品の追加と更新(選択されたカテゴリーを使用する)
- 前の画面のサンプルコードをガイドラインとして使用してください。
- カテゴリーセグメントの行をクリックすると、そのカテゴリーに関連する商品が読み込まれます。

HCL

www.hcltech.com

\$10 BILLION | 159,000+ IDEAPRENEURS | 50 COUNTRIES