



目次

- 1. 改訂情報
- 2. はじめに
 - 2.1. 本書の目的
 - 2.2. 対象読者
 - 2.3. サンプルコードについて
 - 2.4. 本書の構成
- 3. 概要
 - 3.1. OData 連携とは
 - 3.2. OData 連携の全体像と、本チュートリアルガイドの説明範囲
 - 3.3. チュートリアルの準備
- 4. OData(V2) サービスのエンティティ操作
 - 4.1. チュートリアル
 - 4.1.1. ゴールの説明
 - 4.1.2. 準備
 - 4.1.3. エンティティの登録
 - 4.1.3.1. OData(V2)定義の作成
 - 4.1.3.2. フロー定義の作成
 - 4.1.3.3. 動作確認
 - 4.1.4. エンティティの取得
 - 4.1.4.1. OData(V2)定義の作成
 - 4.1.4.2. フロー定義の作成
 - 4.1.4.3. 動作確認
 - 4.1.5. シングルエンティティの取得
 - 4.1.5.1. OData(V2)定義の作成
 - 4.1.5.2. フロー定義の作成
 - 4.1.5.3. 動作確認
 - 4.1.6. エンティティの更新
 - 4.1.6.1. OData(V2)定義の作成
 - 4.1.6.2. フロー定義の作成
 - 4.1.6.3. 動作確認
 - 4.1.7. エンティティの削除
 - 4.1.7.1. OData(V2)定義の作成
 - 4.1.7.2. フロー定義の作成
 - 4.1.7.3. 動作確認
 - 4.1.8. まとめ
- 5. OData(V4) サービスのエンティティ操作
 - 5.1. チュートリアル
 - 5.1.1. ゴールの説明
 - 5.1.2. 準備
 - 5.1.3. エンティティの登録
 - 5.1.3.1. OData(V4)定義の作成
 - 5.1.3.2. フロー定義の作成
 - 5.1.3.3. 動作確認
 - 5.1.4. エンティティの取得
 - 5.1.4.1. OData(V4)定義の作成
 - 5.1.4.2. フロー定義の作成
 - 5.1.4.3. 動作確認
 - 5.1.5. シングルエンティティの取得

- 5.1.5.1. OData(V4)定義の作成
- 5.1.5.2. フロー定義の作成
- 5.1.5.3. 動作確認
- 5.1.6. エンティティの更新
 - 5.1.6.1. OData(V4)定義の作成
 - 5.1.6.2. フロー定義の作成
 - 5.1.6.3. 動作確認
- 5.1.7. エンティティの削除
 - 5.1.7.1. OData(V4)定義の作成
 - 5.1.7.2. フロー定義の作成
 - 5.1.7.3. 動作確認
- 5.1.8. まとめ

改訂情報

変更年月日	変更内容
-------	------

2016-12-01	初版
------------	----

本書の目的

本書は、OData 連携を利用してビジネスロジックの開発を始める開発者のみなさまの支援を目的としたドキュメントです。

対象読者

本書では次の開発者を対象としています。

- OData 連携による開発の一連の流れを知りたい。
- OData 連携を利用してビジネスロジックを開発したい。

なお、本書では次の内容を理解していることが必須となります。

- IM-LogicDesignerを理解している。
- OData を理解している。

また、SAP HANA との OData 連携を行う場合、次のドキュメントを読了し、SAP HANA の設定が完了していることも必須となります。

- [OData for SAP HANA セットアップガイド](#)

サンプルコードについて

本書に掲載されているサンプルコードは可読性を重視しており、性能面や保守性といった観点において必ずしも適切な実装ではありません。

開発においてサンプルコードを参考にされる場合には、上記について十分に注意してください。

本書の構成

本書は次の構成となっています。

- [概要](#)
本書、および、OData 連携の概要について説明します。
- [OData\(V2\) サービスのエンティティ操作](#)
OData(V2) サービスのエンティティを操作する方法を説明します。
- [OData\(V4\) サービスのエンティティ操作](#)
OData(V4) サービスのエンティティを操作する方法を説明します。

概要

- OData 連携とは
- OData 連携の全体像と、本チュートリアルガイドの説明範囲
- チュートリアルの準備

OData 連携とは

OData 連携とは、intra-mart Accel Platform と OData サービスを連携させる機能です。

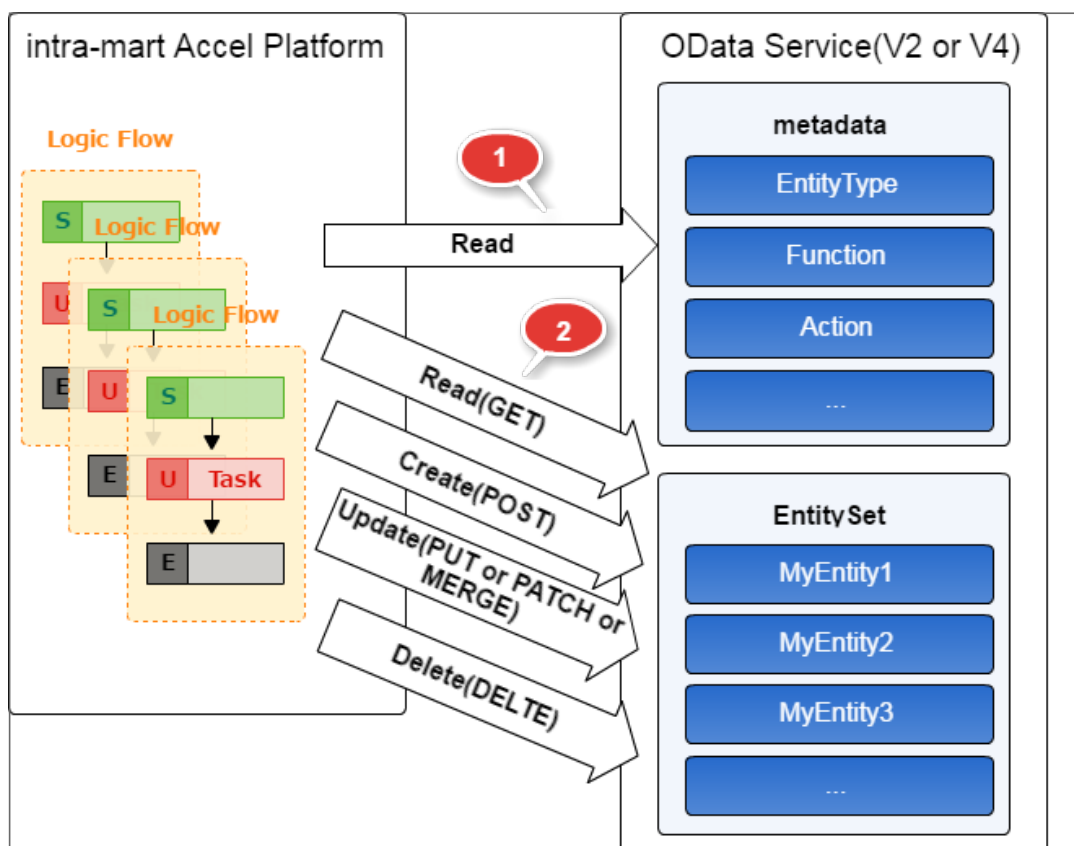
OData 連携の特徴は以下の通りです。

- OData(V2)連携
 - バージョン2 の OData サービスへのエンティティ作成、取得、更新、削除を行えます。
- OData(V4)連携
 - バージョン4 の OData サービスへのエンティティ作成、取得、更新、削除を行えます。

このチュートリアルガイドでは上記概要の順に、実際の操作方法や実行までの手順を開発者の皆様へ紹介します。

OData 連携の全体像と、本チュートリアルガイドの説明範囲

OData 連携の全体像と、各機能に対する本チュートリアルガイドの説明範囲を以下に示します。



本書では吹き出しの番号に対応する次のようなチュートリアルを用意しています。

1. ユーザ定義の作成
 - OData サービスのエンティティを操作するユーザ定義の作成
2. エンティティ作成、取得、更新、削除
 - リクエストの発行による OData サービスへのエンティティ作成、取得、更新、削除

IM-LogicDesigner に関しては本書では特に取り上げません。IM-LogicDesigner の詳細は「[IM-LogicDesigner仕様書](#)」を参照してください。

チュートリアルの準備

このチュートリアルガイドに書かれているチュートリアルを行うには以下の準備が必要です。

- IM-LogicDesignerのフローカテゴリ、ユーザ定義カテゴリが少なくとも一つは作成済みであること。
 - チュートリアルの中では SampleCategory というカテゴリを使用します。

これらの準備が行われていることを確認してください。

また、intra-mart Accel Platform にはテナント管理者またはLogicDesigner管理者でログインしてください。

- チュートリアル
 - ゴールの説明
 - 準備
 - エンティティの登録
 - エンティティの取得
 - シングルエンティティの取得
 - エンティティの更新
 - エンティティの削除
 - まとめ

チュートリアル

このチュートリアルが終了すると、次のような事ができます。

- OData(V2) サービスのエンティティを操作する。
 - 登録
 - 参照
 - 更新
 - 削除

ゴールの説明

このチュートリアルでは、以下で公開されている OData(V2) リファレンスサービスを利用し、製品エンティティの操作を行います。



コラム

[https://services.odata.org/V2/\(S\(im_odata\)\)/OData/OData.svc](https://services.odata.org/V2/(S(im_odata))/OData/OData.svc)

1. エンティティの登録
 - エンティティを登録するユーザ定義タスクを作成します。
 - 動作確認のためのフロー定義を作成します。
 - 動作確認を行います。
2. エンティティの取得
 - エンティティを取得するユーザ定義タスクを作成します。
 - 動作確認のためのフロー定義を作成します。
 - 動作確認を行います。
3. シングルエンティティの取得
 - シングルエンティティを取得するユーザ定義タスクを作成します。
 - 動作確認のためのフロー定義を作成します。
 - 動作確認を行います。
4. エンティティの更新
 - エンティティを更新するユーザ定義タスクを作成します。
 - 動作確認のためのフロー定義を作成します。
 - 動作確認を行います。
5. エンティティの削除

- エンティティを削除するユーザ定義タスクを作成します。
- 動作確認のためのフロー定義を作成します。
- 動作確認を行います。

このような順番で OData(V2) サービスのエンティティの操作方法や作業の流れを説明します。

準備

チュートリアルの準備に書かれている準備が行われていることを確認してください。

エンティティの登録

IM-LogicDesignerのユーザ定義タスクを作成します。作成するのは「OData(V2)定義」です。

後述の「メソッド」を「POST」として作成することで OData サービスのエンティティの登録を行うようにします。

OData(V2)定義の作成

1. intra-mart Accel Platform にテナント管理者またはLogicDesigner管理者でログインします。
2. 「サイトマップ」→「LogicDesigner」→「OData(V2)定義新規作成」を開きます。
3. 「サービスURL」に「[https://services.odata.org/V2/\(S\(im_odata\)\)/OData/OData.svc](https://services.odata.org/V2/(S(im_odata))/OData/OData.svc)」を設定します。

OData(V2)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザーカテゴリ *

検索 新規作成

ユーザーカテゴリID *

ユーザーカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

返却値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

OData(V2)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="X"/>

メソッド * GET シングルエンティティ

クエリオプション

クエリオプション名 *	クエリオプション値 *	削除
-------------	-------------	----

エンティティセット名 *

入力返却値取得 *

i コラム

本チュートリアルでは必要ありませんが、利用する OData サービス に認証がかけられている場合、必要に応じて「認証種別」を「BASIC認証」または「OAuth認証」に設定し、認証情報を設定してください。

i コラム

OData サービスが独自のリクエストヘッダで認証を行う場合、以下のようにリクエストヘッダを追加して認証情報を設定してください。

リクエストヘッダ		+ 追加	削除
ヘッダ名 *	ヘッダ値 *		
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0		✕
Authorization	my custom authorization		✕

- リクエストヘッダ名に独自のリクエストヘッダ名を設定（上記画像では「Authorization」を設定）
- リクエストヘッダ値に認証情報を設定（上記画像では「my custom authorization」を設定）

i コラム

OData サービスが独自のリクエストヘッダで認証を行い、かつフロー実行時にリクエストヘッダを設定する必要がある場合、次のようにして認証情報を設定してください。

入力値

boolean + string + integer + date + object + 追加

JSON入力

配列型にする キー名を変更 型を変更

削除 全削除

- body <object> *
 - ID <integer>
 - Name <string>
 - Description <string>
 - ReleaseDate <calendar> *
 - DiscontinuedDate <calendar>
 - Rating <integer> *
 - Price <bigdecimal> *
 - authorization <string>

返却値

boolean + string + integer + date + object + 追加

JSON入力

配列型にする キー名を変更 型を変更

削除 全削除

- data <object>
 - ID <integer>
 - Name <string>
 - Description <string>
 - ReleaseDate <calendar>
 - DiscontinuedDate <calendar>
 - Rating <integer>
 - Price <bigdecimal>

OData(V2)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ		+ 追加	削除
ヘッダ名 *	ヘッダ値 *		
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0		✕
Authorization	\${authorization}		✕

- 入力値に独自のプロパティを追加（上記の画像例では「authorization」）
- リクエストヘッダ値に「\${authorization}」を設定
 - 「\${authorization}」を設定することで、実行時に入力値「authorization」プロパティに渡された値を設定できます

4. 「エンティティセット名」の横にあるリロードアイコンをクリックします。

OData(V2)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザーカテゴリ *

検索 新規作成

ユーザーカテゴリID *

ユーザーカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

返却値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

OData(V2)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="X"/>

メソッド * GET シングルエンティティ

クエリオプション

クエリオプション名 *	クエリオプション値 *	削除

エンティティセット名 *

入力返却値取得 *

5. 「エンティティセット名」に「Products」を選択します。

OData(V2)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザーカテゴリ * 検索

ユーザーカテゴリID *

ユーザーカテゴリ名

ソート番号 *

アイコン 選択

入力値

boolean 配下に配置する

配列型にする

返却値

boolean 配下に配置する

配列型にする

OData(V2)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="X"/>

メソッド * GET シングルエンティティ

クエリオプション

クエリオプション名 *	クエリオプション値 *	削除
-------------	-------------	----

エンティティセット名 * Products

入力返却値取得 *

6. 「メソッド」に「POST」を選択します。

OData(V2)定義編集 *

ユーザ定義一覧

ユーザ定義共通設定

ユーザ定義ID *

バージョン * 1

ユーザ定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザカテゴリ *

検索 新規作成

ユーザカテゴリID *

ユーザカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean + 追加 配下に配置する

配列型にする

返却値

+ string + integer + date + object

boolean + 追加 配下に配置する

配列型にする

OData(V2)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

+ 追加 <input type="button" value=""/>		
ヘッダ名 *	ヘッダ値 *	削除 <input type="button" value=""/>
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="X"/>

メソッド *

エンティティセット名 *

入力返却値取得 *

7. 「入力/返却値取得」の横にある「取得」ボタンをクリックします。

OData(V2)定義編集 *

ユーザ定義一覧

ユーザ定義共通設定

ユーザ定義ID *

バージョン * 1

ユーザ定義名 *	標準 *	<input type="text"/>
	日本語	<input type="text"/>
	英語	<input type="text"/>
	中国語 (中華人民共和国)	<input type="text"/>

ユーザカテゴリ * 検索

ユーザカテゴリID *

ユーザカテゴリ名

ソート番号 *

アイコン

入力値

boolean 配下に配置する

配列型にする

- ▾ body <object> *
 - ID <integer>
 - Name <string>
 - Description <string>
 - ReleaseDate <calendar> *
 - DiscontinuedDate <calendar>
 - Rating <integer> *
 - Price <bigdecimal> *

返却値

boolean 配下に配置する

配列型にする

- ▾ data <object>
 - ID <integer>
 - Name <string>
 - Description <string>
 - ReleaseDate <calendar>
 - DiscontinuedDate <calendar>
 - Rating <integer>
 - Price <bigdecimal>

OData(V2)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ	<input type="button" value="+ 追加"/>		
ヘッダ名 *	ヘッダ値 *		<input type="button" value="削除"/>
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0		<input type="button" value="X"/>

メソッド *

エンティティセット名 *

入力返却値取得 *

i コラム

「入力値」ツリーの「body」配下にあるプロパティが登録するデータ、「返却値」ツリーの「data」配下にあるプロパティが、登録されたデータを表します。

8. その他の項目は以下のように入力してください。

ユーザ定義ID ODataV2Post

ユーザ定義名	ODataV2Post
--------	-------------

ユーザカテゴリID	SampleCategory
-----------	----------------

9. 「登録」 ボタンをクリックします。
 10. 確認ダイアログで「OK」 ボタンをクリックします。
- これで「OData(V2)定義」の作成は終了です。

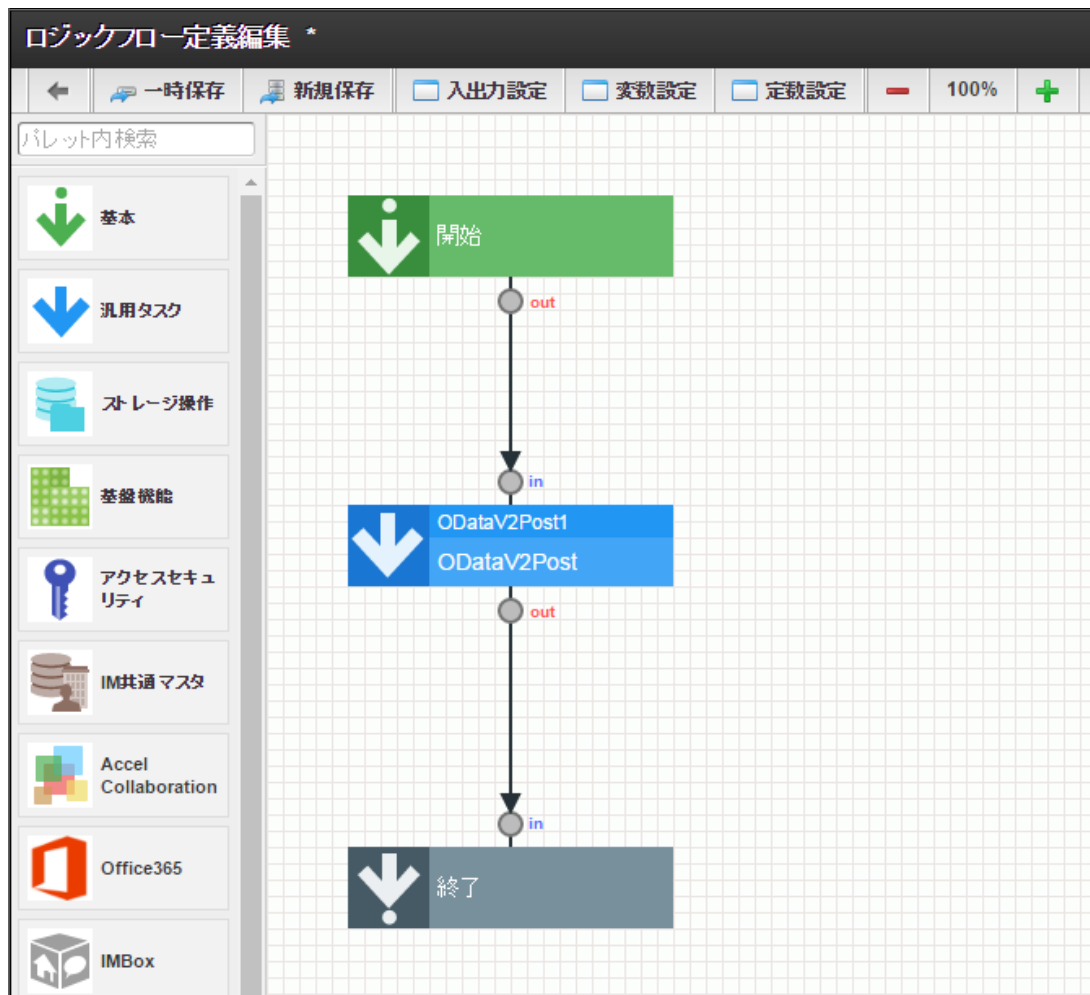
フロー定義の作成

続いてフロー定義の作成を行います。

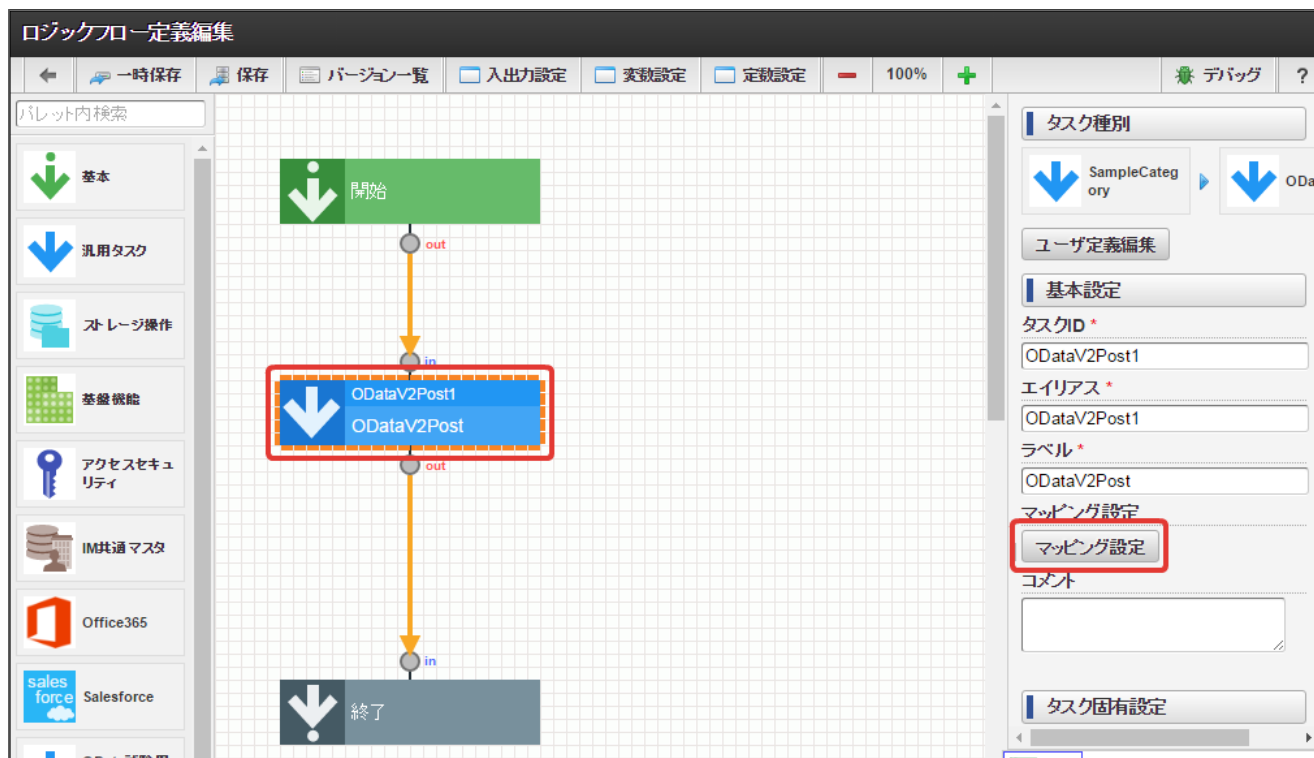
11. 「サイトマップ」 → 「LogicDesigner」 → 「フロー定義」を開きます。
12. 「新規作成」リンクをクリックし、「ロジックフロー定義編集」画面を表示します。
13. 「入出力設定」リンクをクリックし、「入力」に次のような項目を追加し、「決定」ボタンをクリックします。

キー名	型
ID	integer
Name	string
Description	string
ReleaseDate	calendar
DiscontinuedDate	calendar
Rating	integer
Price	bigdecimal

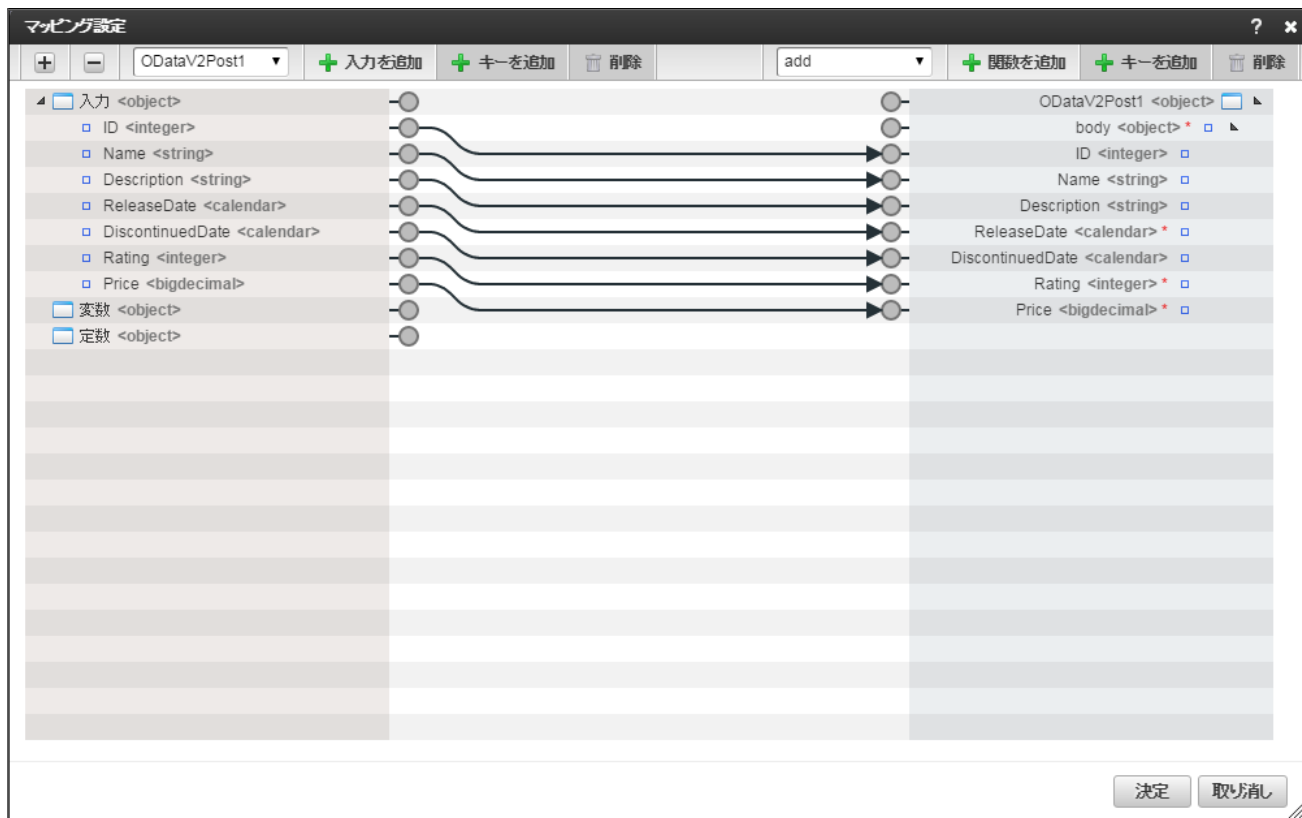
14. ロジックフロー定義編集画面のデザイナーに、先ほど作成した「ODataV2Post」タスクを追加し、「開始」「終了」と接続します。



15. 「ODataV2Post」タスクをクリックし、「マッピング設定」ボタンをクリックします。



16. 「入力」の各項目と、「ODataV2Post」の各項目を紐づけ、「決定」ボタンをクリックします。



17. 「新規保存」リンクをクリックし、新規保存ダイアログを表示します。ダイアログに次のような内容を入力し「決定」ボタンをクリックします。

フロー定義ID	ODataV2Post
フロー定義名 標準	ODataV2Post
フローカテゴリID	SampleCategory

18. 「確認」ダイアログで「決定」ボタンをクリックします。

動作確認

最後に動作確認を行います。IM-LogicDesigner のデバッグ機能を使ってフローを実行し、OData サービスにエンティティが作成されることを確認します。

19. 「ODataV2Post」フローの「編集」アイコンをクリックします。
20. 「ロジックフロー定義編集」画面の「デバッグ」リンクをクリックします。
21. 「ロジックフロー定義のデバッグ」画面の「実行」リンクをクリックします。
22. 「デバッグの実行」ダイアログが表示されるので、次のように入力し「実行」ボタンをクリックし、「確認」ダイアログで「決定」ボタンをクリックします。

ID	9
Name	テスト(Name)
Description	テスト(Description)
ReleaseDate	2016-12-01T10:00:00Z
DiscontinuedDate	2017-12-01T10:00:00Z
Rating	100
Price	1000

23. これで、OData サービスにエンティティが作成されました。

本当に作成されたかどうかを確認します。

24. ブラウザから「[https://services.odata.org/V2/\(S\(im_odata\)\)/OData/OData.svc/Products](https://services.odata.org/V2/(S(im_odata))/OData/OData.svc/Products)」にアクセスします。

25. フローのデバッグ画面で入力した値でエンティティが作成されていることを確認できます。

```

</content>
</entry>
<entry>
  <id>http://services.odata.org/(S(im_odata))/V2/OData/OData.svc/Products(9)</id>
  <title type="text">テスト(Name)</title>
  <summary type="text">テスト(Description)</summary>
  <updated>2016-11-21T12:33:19Z</updated>
  <author>
    <name/>
  </author>
  <link rel="edit" title="Product" href="Products(9)"/>
  <link rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/Category" type="application/atom+xml;type=entry" title="Category" href="Products(9)/Category"/>
  <link rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/Supplier" type="application/atom+xml;type=entry" title="Supplier" href="Products(9)/Supplier"/>
  <category term="ODataDemo.Product" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <content type="application/xml">
    <m:properties>
      <d:ID m:type="Edm.Int32">9</d:ID>
      <d:ReleaseDate m:type="Edm.DateTime">2016-12-01T10:00:00Z</d:ReleaseDate>
      <d:DiscontinuedDate m:type="Edm.DateTime">2017-12-01T10:00:00Z</d:DiscontinuedDate>
      <d:Rating m:type="Edm.Int32">100</d:Rating>
      <d:Price m:type="Edm.Decimal">1000</d:Price>
    </m:properties>
  </content>
</entry>
</feed>

```

エンティティの取得

次に、エンティティを参照する IM-LogicDesigner のユーザ定義タスクを作成します。作成するのは「OData(V2)定義」です。

「メソッド」を「GET」として作成することで OData サービスのエンティティの取得を行うようにします。

OData(V2)定義の作成

エンティティの登録の 1 から 5 までの手順を繰り返してください。

1. 「メソッド」に「GET」を選択します。
2. 「クエリオプション」の横にある「追加」アイコンをクリックし、以下のクエリオプションを追加します。
 - 「クエリオプション名」を「\$orderby」
 - 「クエリオプション値」を「ID」

コラム

「\$orderby」に「ID」を指定することで、IDでソートされた結果を取得する事が出来ます。クエリオプションの詳細については以下のドキュメントを参照してください。

- [System Query Options](#)

3. 「入力/返却値取得」の横にある「取得」ボタンをクリックします。

OData(V2)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザーカテゴリ *

検索 新規作成

ユーザーカテゴリID *

ユーザーカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

返却値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

```

data <object[]>
  ID <integer>
  Name <string>
  Description <string>
  ReleaseDate <calendar>
  DiscontinuedDate <calendar>
  Rating <integer>
  Price <bigdecimal>
    
```

OData(V2)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="X"/>

メソッド * シングルエンティティ

クエリオプション

クエリオプション名 *	クエリオプション値 *	削除
\$orderby	ID	<input type="button" value="X"/>

エンティティセット名 *

入力返却値取得 *

i コラム

「返却値」ツリーの「data」配下にあるプロパティが取得するデータを表します。

i コラム

「返却値」ツリーの「data」配下にあるプロパティは、選択後「削除」をクリックすることで削除できます。不要なプロパティを削除することで、余計なプロパティの取得を防ぎ、実行時の負荷を軽減できます。

4. その他の項目は以下のように入力してください。

ユーザ定義ID	ODataV2Get
ユーザ定義名	ODataV2Get
ユーザカテゴリID	SampleCategory

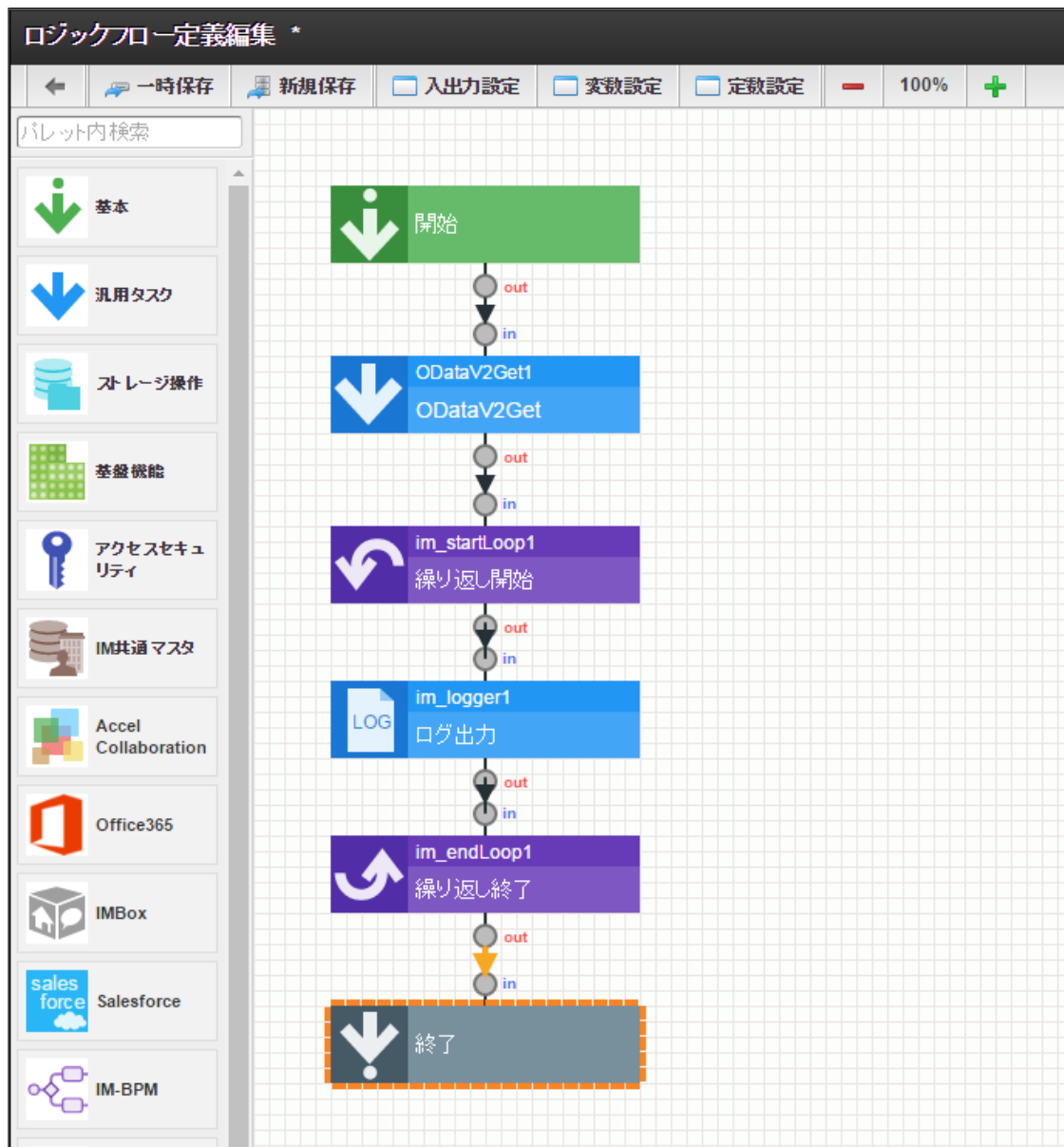
5. 「登録」 ボタンをクリックします。
6. 確認ダイアログで「OK」 ボタンをクリックします。

これで「OData(V2)定義」の作成は終了です。

フロー定義の作成

続いてフロー定義の作成を行います。

7. 「サイトマップ」 → 「LogicDesigner」 → 「フロー定義」を開きます。
8. 「新規作成」リンクをクリックし、「ロジックフロー定義編集」画面を表示します。
9. ロジックフロー定義編集画面のデザイナーに、先ほど作成した「ODataV2Get」タスクと、「繰り返し開始」「繰り返し終了」「ログ出力」タスクを追加し、「開始」「終了」と接続します。



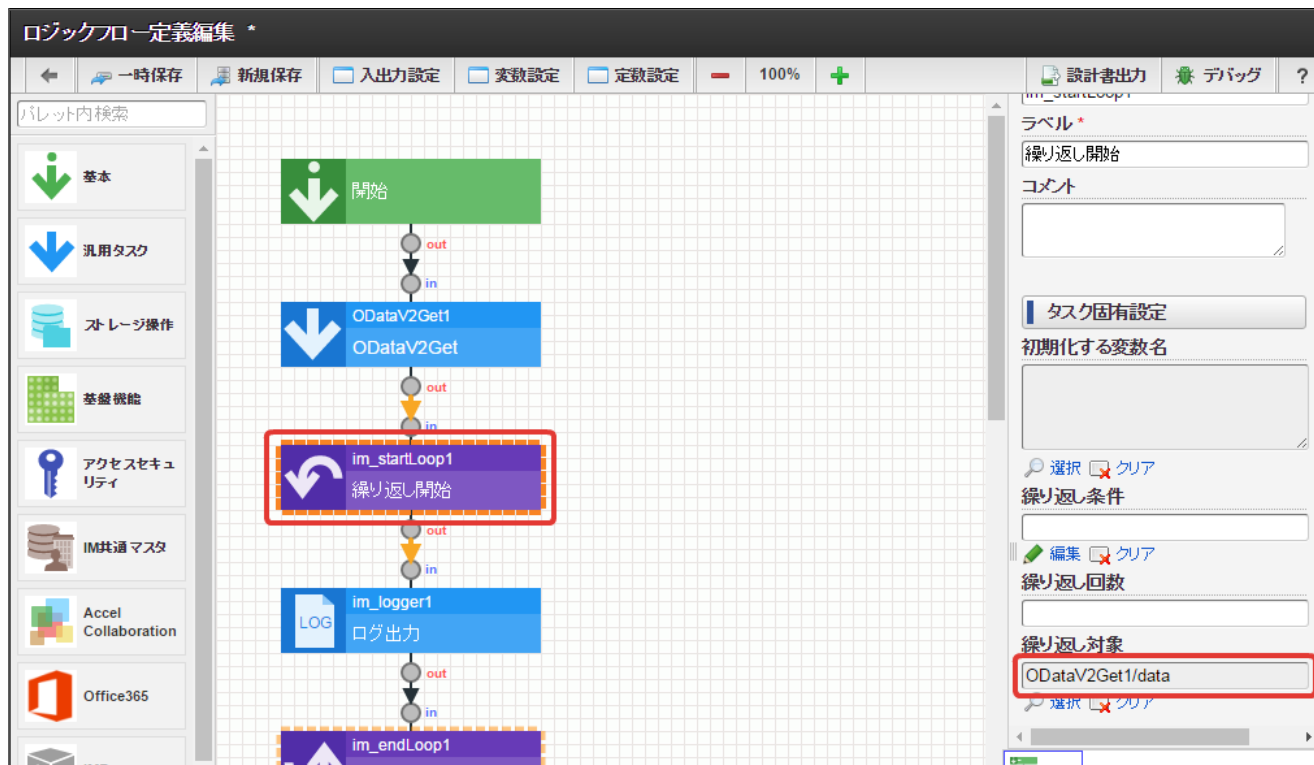
10. 「定数設定」リンクをクリックし、次のような項目を追加し、「決定」ボタンをクリックします。

- 「定数ID」を「eI」
- 「定数値」を以下

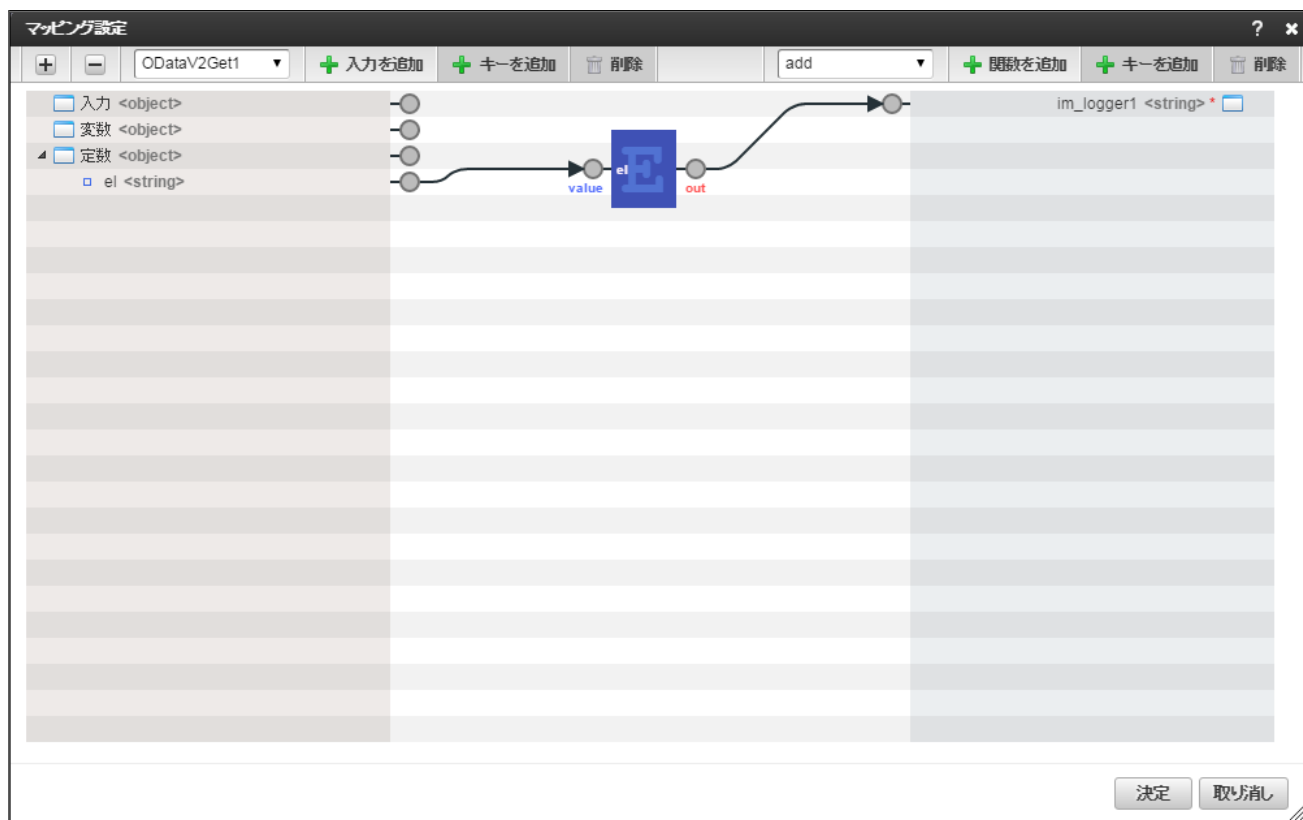
```

ID:${im_startLoop1.item.ID}
ReleaseDate:${im_startLoop1.item.ReleaseDate}
DiscontinuedDate:${im_startLoop1.item.DiscontinuedDate}
Rating:${im_startLoop1.item.Rating}
Price:${im_startLoop1.item.Price}
  
```

11. 「繰り返し開始」タスクをクリックし、「繰り返し対象」に「ODataV2Get1/data」を設定します。



12. 「ログ出力」タスクをクリックし、「マッピング設定」ボタンをクリックし、次のように各項目を紐づけ、「決定」ボタンをクリックします。



13. 「新規保存」リンクをクリックし、新規保存ダイアログを表示します。ダイアログに次のような内容を入力し「決定」ボタンをクリックします。

フロー定義ID ODataV2Get
 フロー定義名 標準 ODataV2Get
 フローカテゴリID SampleCategory

14. 「確認」ダイアログで「決定」ボタンをクリックします。

これでフロー定義の作成が終了です。

動作確認

最後に動作確認を行います。IM-LogicDesigner のデバッグ機能を使ってフローを実行し、OData サービスのデータが取得されることを確認します。

15. 「ODataV2Get」フローの「編集」アイコンをクリックします。
16. 「ロジックフロー定義編集」画面の「デバッグ」リンクをクリックします。
17. 「ロジックフロー定義のデバッグ」画面の「実行」リンクをクリックします。
18. 「デバッグの実行」ダイアログが表示されるので、「実行」ボタンをクリックし、「確認」ダイアログで「決定」ボタンをクリックします。
19. 以下のログが出力されます。

```

[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:0
ReleaseDate:java.util.GregorianCalendar[time=694224000000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,zo

DiscontinuedDate:
Rating:4
Price:2.5
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:1
ReleaseDate:java.util.GregorianCalendar[time=812505600000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,zo

DiscontinuedDate:
Rating:3
Price:3.5
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:2
ReleaseDate:java.util.GregorianCalendar[time=970358400000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,zo

DiscontinuedDate:
Rating:3
Price:20.9
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:3
ReleaseDate:java.util.GregorianCalendar[time=1128124800000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,z

DiscontinuedDate:java.util.GregorianCalendar[time=1159660800000,areFieldsSet=true,areAllFieldsSet=true,lenient=t

Rating:3
Price:19.9
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:4
ReleaseDate:java.util.GregorianCalendar[time=1041724800000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,z

DiscontinuedDate:
Rating:3
Price:22.99
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:5
ReleaseDate:java.util.GregorianCalendar[time=1154649600000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,z

DiscontinuedDate:
Rating:3
Price:22.8
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:6
ReleaseDate:java.util.GregorianCalendar[time=1162684800000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,z

DiscontinuedDate:
Rating:3
Price:18.8
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:7
ReleaseDate:java.util.GregorianCalendar[time=1163548800000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,z

DiscontinuedDate:
Rating:3
Price:35.88
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:8
ReleaseDate:java.util.GregorianCalendar[time=1210204800000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,z

DiscontinuedDate:
Rating:3
Price:1088.8
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:9
ReleaseDate:java.util.GregorianCalendar[time=1480586400000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,z

DiscontinuedDate:java.util.GregorianCalendar[time=1512122400000,areFieldsSet=true,areAllFieldsSet=true,lenient=t

Rating:100
Price:1000

```

20. ブラウザから「[https://services.odata.org/V2/\(S\(im_odata\)\)/OData/OData.svc/Products](https://services.odata.org/V2/(S(im_odata))/OData/OData.svc/Products)」にアクセスします。

ブラウザに表示されるデータと、出力されたログの内容が一致することが確認できます。

次に、シングルエンティティを参照する IM-LogicDesigner のユーザ定義タスクを作成します。作成するのは「OData(V2)定義」です。

「メソッド」を「GET」、「シングルエンティティ」を選択することで OData サービスのシングルエンティティの取得を行うようにします。

OData(V2)定義の作成

[エンティティの登録](#)の 1 から 5 までの手順を繰り返してください。

1. 「メソッド」に「GET」を選択します。
2. 「メソッド」の横にある「シングルエンティティ」を選択します。
3. 「入力/返却値取得」の横にある「取得」ボタンをクリックします。

OData(V2)定義編集 *

ユーザ定義一覧

ユーザ定義共通設定

ユーザ定義ID *

バージョン * 1

ユーザ定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザカテゴリ *

検索 新規作成

ユーザカテゴリID *

ユーザカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

- ▾ key <object> *
 - ID <integer> *

返却値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

- ▾ data <object>
 - ID <integer>
 - Name <string>
 - Description <string>
 - ReleaseDate <calendar>
 - DiscontinuedDate <calendar>
 - Rating <integer>
 - Price <bigdecimal>

OData(V2)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="削除"/>

メソッド * シングルエンティティ

クエリオプション

クエリオプション名 *	クエリオプション値 *	削除
-------------	-------------	----

エンティティセット名 *

入力返却値取得 *



コラム

「入力値」ツリーの「key」配下にあるプロパティが取得するシングルエンティティを表すキープロパティとして扱われます。

**コラム**

「返却値」ツリーの「data」配下にあるプロパティが取得するデータを表します。

**コラム**

「返却値」ツリーの「data」配下にあるプロパティは、選択後「削除」をクリックすることで削除できます。不要なプロパティを削除することで、余計なプロパティの取得を防ぎ、実行時の負荷を軽減できます。

4. その他の項目は以下のように入力してください。

ユーザ定義ID	ODataV2GetSingle
ユーザ定義名	ODataV2GetSingle
ユーザカテゴリID	SampleCategory

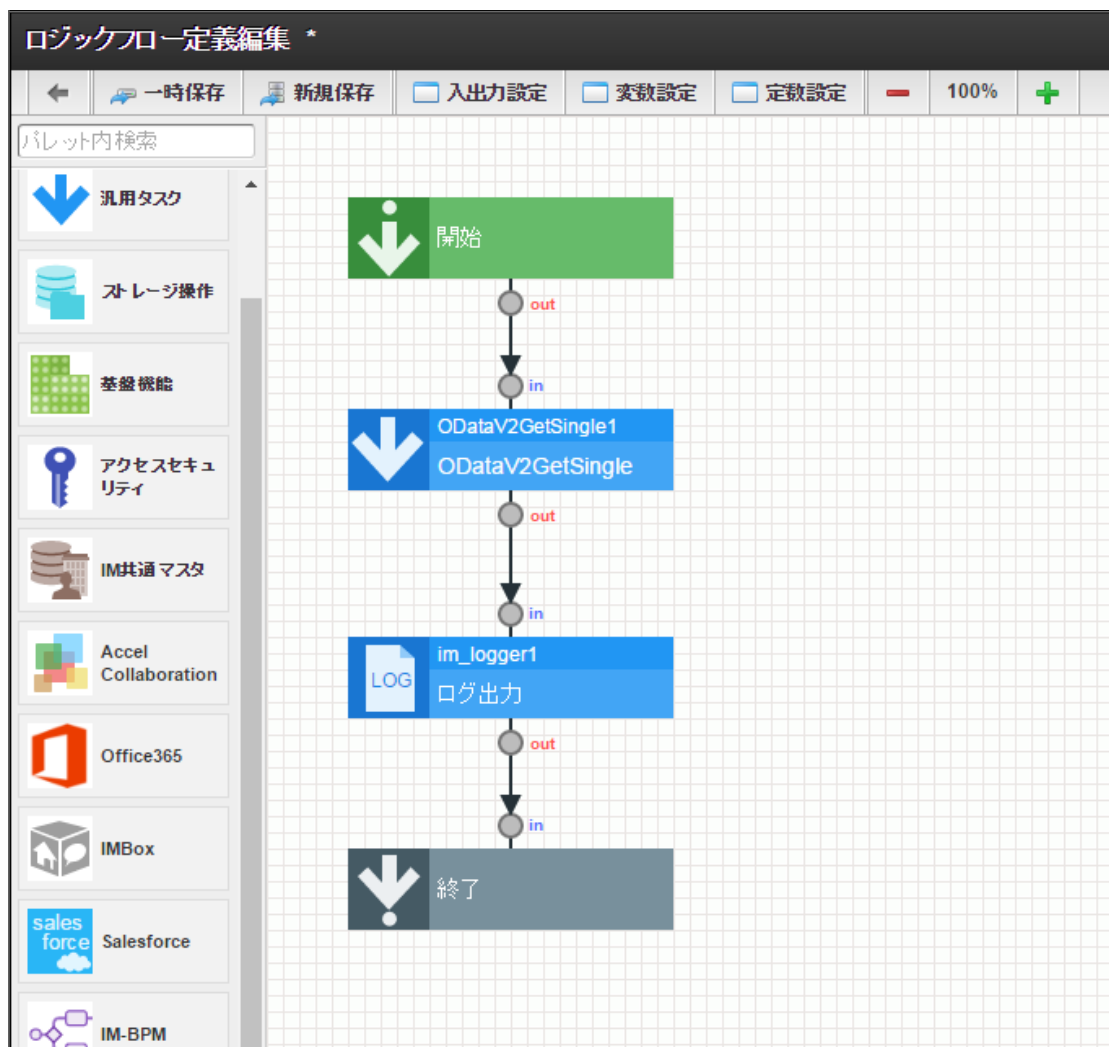
5. 「登録」ボタンをクリックします。
6. 確認ダイアログで「OK」ボタンをクリックします。

これで「OData(V2)定義」の作成は終了です。

フロー定義の作成

続いてフロー定義の作成を行います。

7. 「サイトマップ」→「LogicDesigner」→「フロー定義」を開きます。
8. 「新規作成」リンクをクリックし、「ロジックフロー定義編集」画面を表示します。
9. ロジックフロー定義編集画面のデザイナーに、先ほど作成した「ODataV2GetSingle」タスクと「ログ出力」タスクを追加し、「開始」「終了」と接続します。



10. 「入出力設定」リンクをクリックし、「入力」に次のような項目を追加し、「決定」ボタンをクリックします。

キー名	型
-----	---

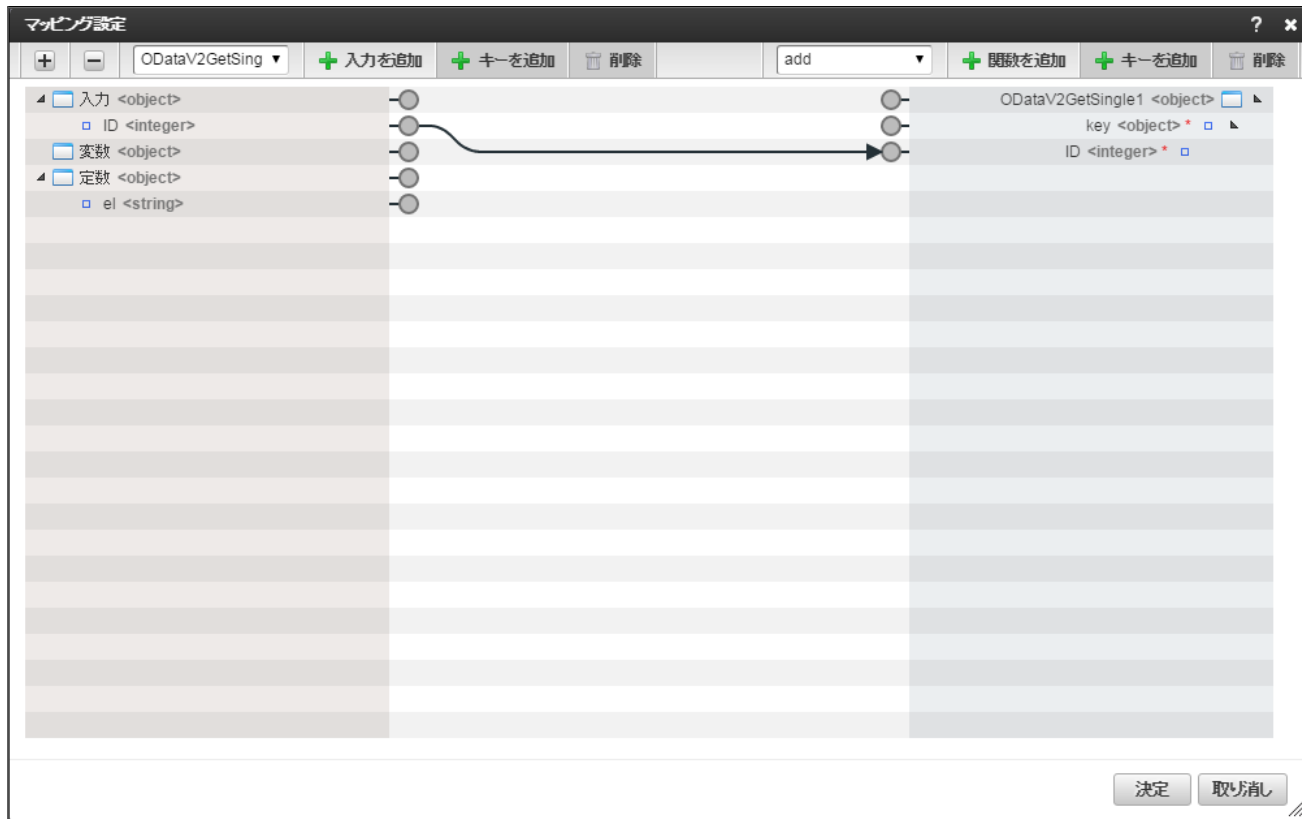
ID	integer
----	---------

11. 「定数設定」リンクをクリックし、次のような項目を追加し、「決定」ボタンをクリックします。

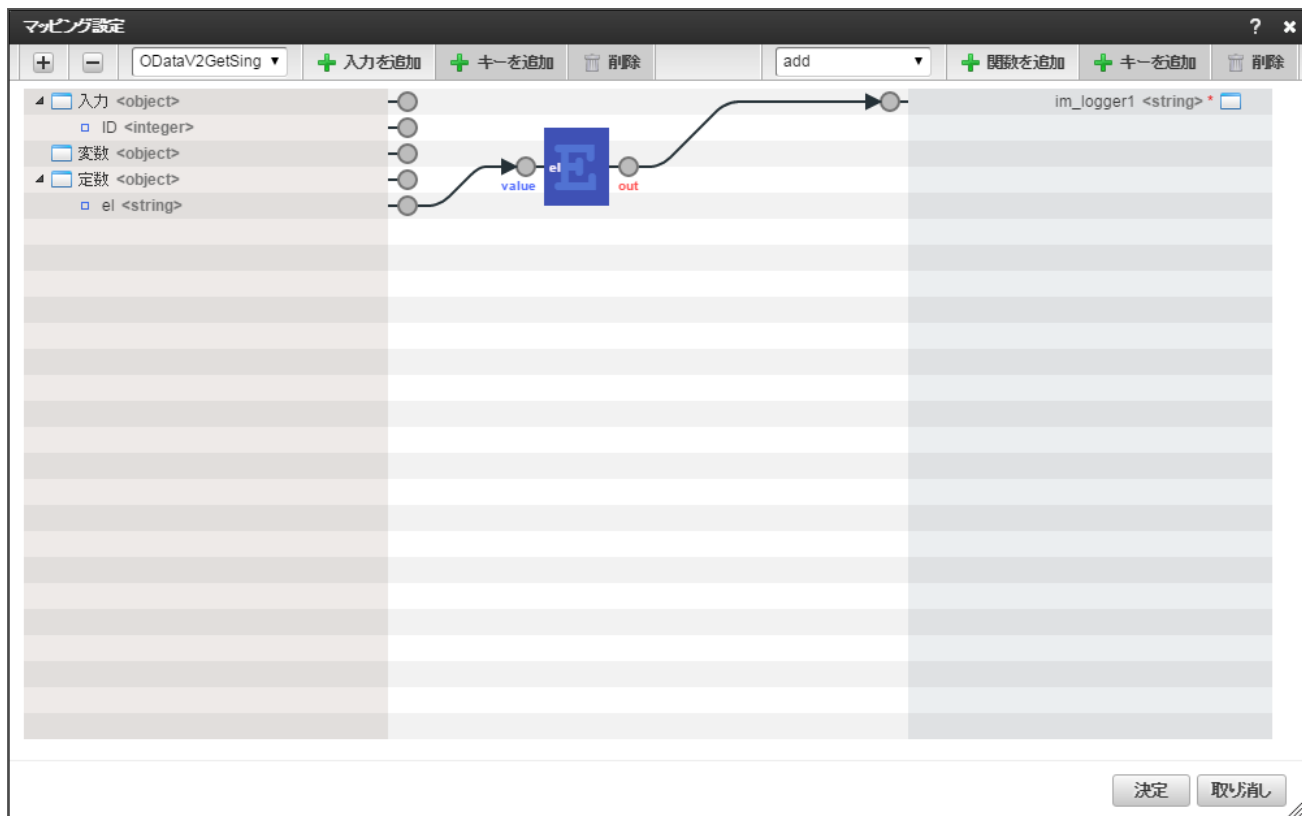
- 「定数ID」を「e1」
- 「定数値」を以下

```
ID:${ODataV2GetSingle1.data.ID}
ReleaseDate:${ODataV2GetSingle1.data.ReleaseDate}
DiscontinuedDate:${ODataV2GetSingle1.data.DiscontinuedDate}
Rating:${ODataV2GetSingle1.data.Rating}
Price:${ODataV2GetSingle1.data.Price}
```

12. 「ODataV2GetSingle」タスクをクリックし、「マッピング設定」ボタンをクリックし、次のように各項目を紐づけ、「決定」ボタンをクリックします。



13. 「ログ出力」タスクをクリックし、「マッピング設定」ボタンをクリックし、次のように各項目を紐づけ、「決定」ボタンをクリックします。



14. 「新規保存」リンクをクリックし、新規保存ダイアログを表示します。ダイアログに次のような内容を入力し「決定」ボタンをクリックします。

フロー定義ID	ODataV2GetSingle
フロー定義名 標準	ODataV2GetSingle
フローカテゴリID	SampleCategory

15. 「確認」ダイアログで「決定」ボタンをクリックします。

これでフロー定義の作成が終了です。

動作確認

最後に動作確認を行います。IM-LogicDesigner のデバッグ機能を使ってフローを実行し、OData サービスのデータが取得されることを確認します。

16. 「ODataV2GetSingle」フローの「編集」アイコンをクリックします。

17. 「ロジックフロー定義編集」画面の「デバッグ」リンクをクリックします。

18. 「ロジックフロー定義のデバッグ」画面の「実行」リンクをクリックします。

19. 「デバッグの実行」ダイアログが表示されるので、次のように入力し「実行」ボタンをクリックし、「確認」ダイアログで「決定」ボタンをクリックします。

ID 7

20. 以下のログが出力されます。

```
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:7
ReleaseDate:java.util.GregorianCalendar[time=1163548800000,areFieldsSet=true,areAllFieldsSet=true,lenient=true,z
DiscontinuedDate:
Rating:3
Price:35.88
```

21. ブラウザから「[https://services.odata.org/V2/\(S\(im_odata\)\)/OData/OData.svc/Products\(7\)](https://services.odata.org/V2/(S(im_odata))/OData/OData.svc/Products(7))」にアクセスします。

ブラウザに表示されるデータと、出力されたログの内容が一致することが確認できます。

エンティティの更新

次に、エンティティを更新する IM-LogicDesigner のユーザ定義タスクを作成します。作成するのは「OData(V2)定義」です。

「メソッド」を「PUT」または「MERGE」として作成することで OData サービスのエンティティの更新を行うようにします。

コラム

「PUT」と「MERGE」はどちらもエンティティの更新を行いますが、「PUT」は入力プロパティの更新を行い、入力プロパティに与えられなかったプロパティは初期値に更新されます。

これとは逆に「MERGE」では入力プロパティに与えられたプロパティの更新のみを行い、入力プロパティに与えられなかったプロパティは更新されません。

本チュートリアルでは「メソッド」に「MERGE」を指定してエンティティの更新を行います。

OData(V2)定義の作成

[エンティティの登録](#)の 1 から 5 までの手順を繰り返してください。

1. 「メソッド」に「MERGE」を選択します。
2. 「入力/返却値取得」の横にある「取得」ボタンをクリックします。

3. その他の項目は以下のように入力してください。

ユーザ定義ID	ODataV2Merge
ユーザ定義名	ODataV2Merge
ユーザカテゴリID	SampleCategory

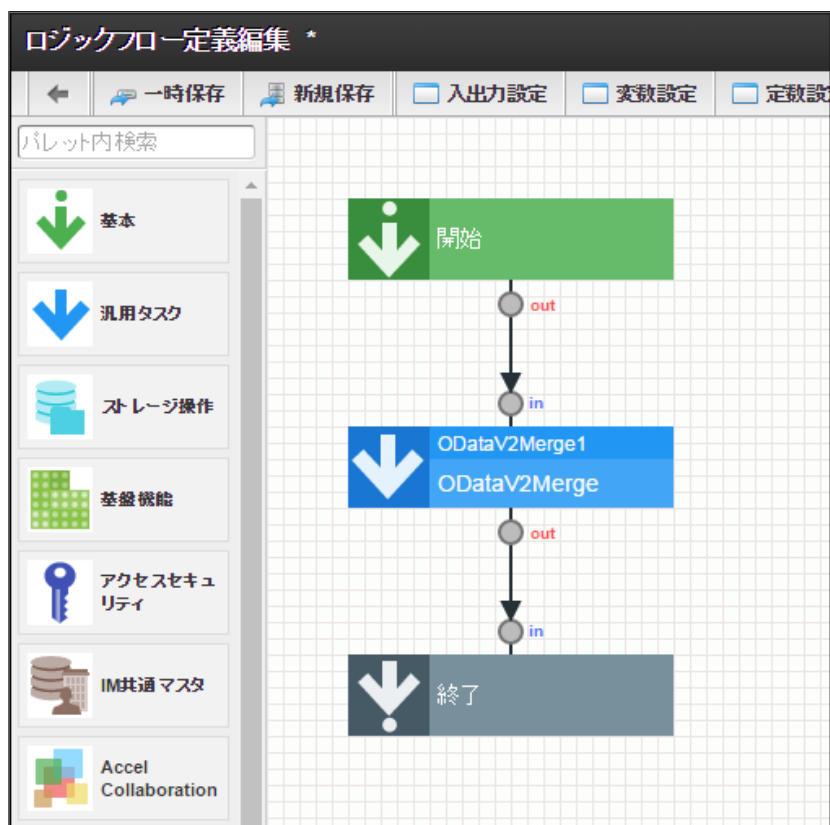
- 「登録」 ボタンをクリックします。
- 確認ダイアログで「OK」 ボタンをクリックします。

これで「OData(V2)定義」の作成は終了です。

フロー定義の作成

続いてフロー定義の作成を行います。

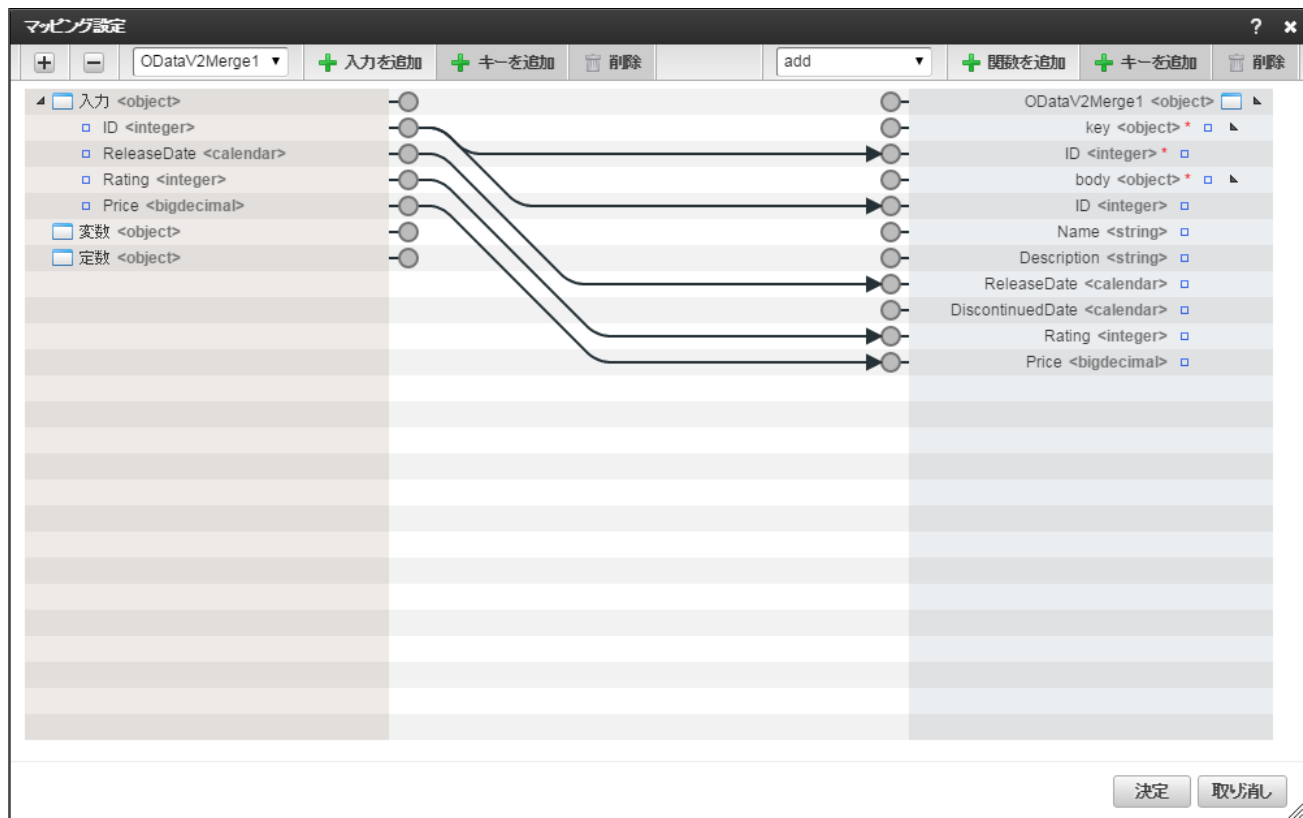
- 「サイトマップ」→「LogicDesigner」→「フロー定義」を開きます。
- 「新規作成」リンクをクリックし、「ロジックフロー定義編集」画面を表示します。
- ロジックフロー定義編集画面のデザイナーに、先ほど作成した「ODataV2Merge」タスクを追加し、「開始」「終了」と接続します。



- 「入出力設定」リンクをクリックし、「入力」に次のような項目を追加し、「決定」ボタンをクリックします。

キー名	型
ID	integer
ReleaseDate	calendar
Rating	integer
Price	bigdecimal

- 「ODataV2Merge」タスクをクリックし、「マッピング設定」ボタンをクリックし、次のように各項目を紐づけ、「決定」ボタンをクリックします。



- 「新規保存」リンクをクリックし、新規保存ダイアログを表示します。ダイアログに次のような内容を入力し「決定」ボタンをクリックします。

フロー定義ID	ODataV2Merge
フロー定義名 標準	ODataV2Merge
フローカテゴリID	SampleCategory

- 「確認」ダイアログで「決定」ボタンをクリックします。

これでフロー定義の作成が終了です。

動作確認

最後に動作確認を行います。IM-LogicDesigner のデバッグ機能を使ってフローを実行し、OData サービスのデータが更新されることを確認します。

- 「ODataV2Merge」フローの「編集」アイコンをクリックします。
- 「ロジックフロー定義編集」画面の「デバッグ」リンクをクリックします。
- 「ロジックフロー定義のデバッグ」画面の「実行」リンクをクリックします。
- 「デバッグの実行」ダイアログが表示されるので、次のように入力し「実行」ボタンをクリックし、「確認」ダイアログで「決定」ボタンをクリックします。

ID	3
ReleaseDate	2016-12-01T10:00:00Z
Rating	100
Price	1000

- これで、OData サービスのエントリティが更新されました。

本当に作成されたかどうかを確認します。

- ブラウザから「[https://services.odata.org/V2/\(S\(im_odata\)\)/OData/OData.svc/Products\(3\)](https://services.odata.org/V2/(S(im_odata))/OData/OData.svc/Products(3))」にアクセスします。

19. フローのデバッグ画面で入力した値でエンティティが更新されていることを確認できます。

```
<entry xml:base="http://services.odata.org/(S(in_odata))/V2/OData/OData.svc/" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microso
<id http://services.odata.org/(S(in_odata))/V2/OData/OData.svc/Products(3)/id>
<title type="text">Havina Cola</title>
<summary type="text">The Original Key Lime Cola</summary>
<updated>2016-11-22T02:52:04Z</updated>
<author>
  <name/>
</author>
<link rel="edit" title="Product" href="Products(3)"/>
<link rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/Category" type="application/atom+xml;type=entry" title="Category" href="Products(3)/Category"/>
<link rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/Supplier" type="application/atom+xml;type=entry" title="Supplier" href="Products(3)/Supplier"/>
<category term="ODataDemo.Product" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
<content type="application/xml">
  <m:properties>
    <d:ID m:type="Edm.Int32">3</d:ID>
    <d:ReleaseDate m:type="Edm.DateTime">2016-12-01T10:00:00Z</d:ReleaseDate>
    <d:DiscontinuedDate m:type="Edm.DateTime">2006-10-01T00:00:00</d:DiscontinuedDate>
    <d:Rating m:type="Edm.Int32">100</d:Rating>
    <d:Price m:type="Edm.Decimal">1000</d:Price>
  </m:properties>
</content>
</entry>
```



コラム

更新処理では、キープロパティを変更することはできません。



コラム

入力データとして与えられた「ReleaseDate」「Rating」「Price」プロパティが更新され、入力データに与えられなかった「DiscontinuedDate」プロパティは更新されていないことが確認できます。

メソッドに「PUT」を指定した場合、入力データとして与えられた「ReleaseDate」「Rating」「Price」プロパティは同様に更新されますが、入力データに与えられなかった「DiscontinuedDate」プロパティは初期値に更新されます。

エンティティの削除

最後に、エンティティを削除する IM-LogicDesigner のユーザ定義タスクを作成します。作成するのは「OData(V2)定義」です。

「メソッド」を「DELETE」として作成することで OData サービスのエンティティの削除を行うようにします。

OData(V2)定義の作成

エンティティの登録の 1 から 5 までの手順を繰り返してください。

1. 「メソッド」に「DELETE」を選択します。
2. 「入力/返却値取得」の横にある「取得」ボタンをクリックします。

OData(V2)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザーカテゴリ *

検索 新規作成

ユーザーカテゴリID *

ユーザーカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean + 追加 配下に配置する JSON入力

配列型にする キー名を変更 型を変更

削除 全削除

- key <object> *
 - ID <integer> *

返却値

+ string + integer + date + object

boolean + 追加 配下に配置する JSON入力

配列型にする キー名を変更 型を変更

削除 全削除

OData(V2)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ + 追加

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button"/>

メソッド * **DELETE**

エンティティセット名 * Products

入力返却値取得 * **取得**



コラム

「入力値」ツリーの「key」配下にあるプロパティが削除するエンティティを表すキープロパティとして扱われます。

3. その他の項目は以下のように入力してください。

ユーザー定義ID ODataV2Delete

ユーザ定義名 ODataV2Delete

ユーザカテゴリID SampleCategory

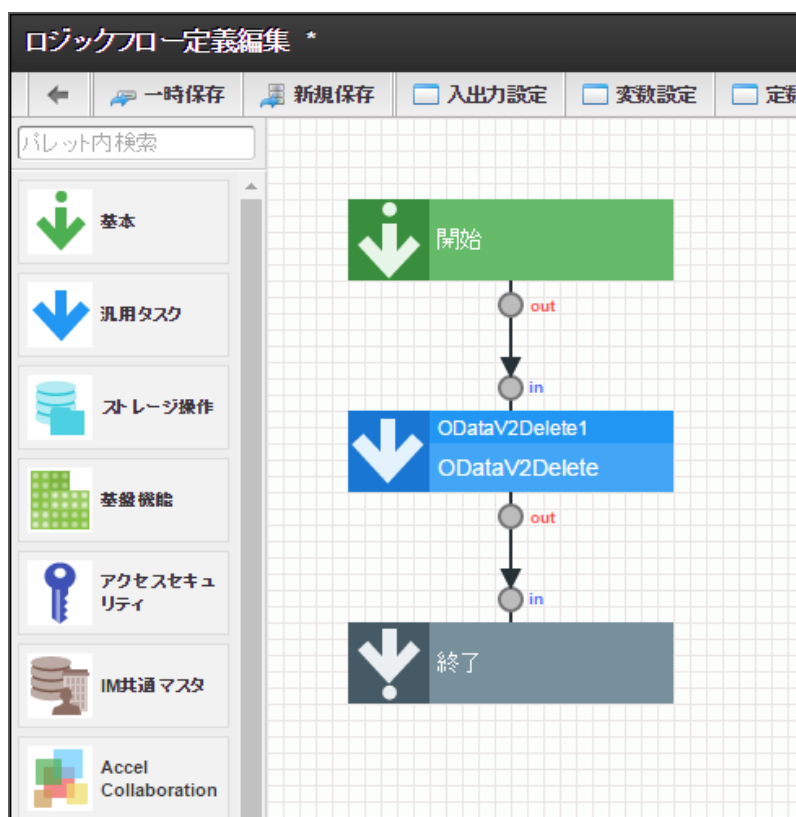
4. 「登録」 ボタンをクリックします。
5. 確認ダイアログで「OK」 ボタンをクリックします。

これで「OData(V2)定義」の作成は終了です。

フロー定義の作成

続いてフロー定義の作成を行います。

6. 「サイトマップ」 → 「LogicDesigner」 → 「フロー定義」を開きます。
7. 「新規作成」リンクをクリックし、「ロジックフロー定義編集」画面を表示します。
8. ロジックフロー定義編集画面のデザイナーに、先ほど作成した「ODataV2Delete」タスクを追加し、「開始」「終了」と接続します。

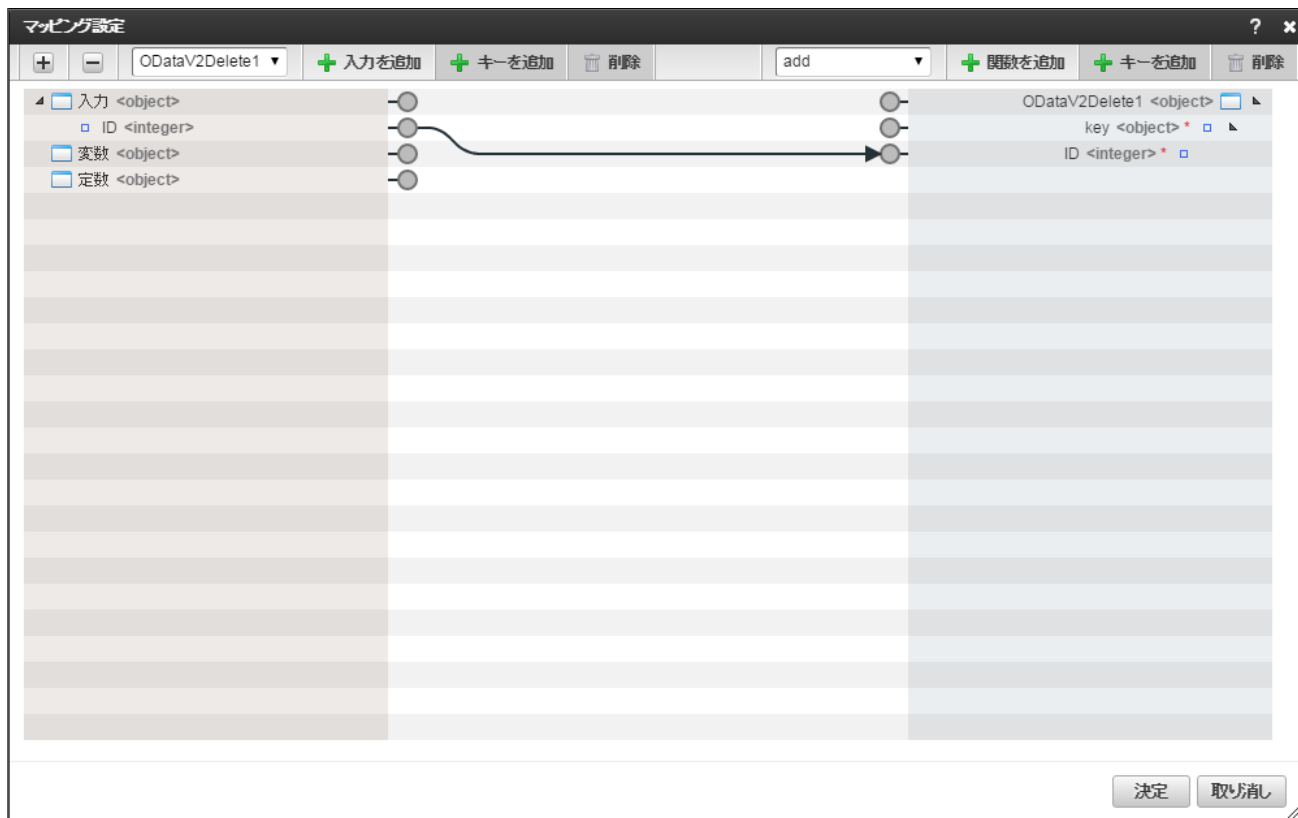


9. 「入出力設定」リンクをクリックし、「入力」に次のような項目を追加し、「決定」ボタンをクリックします。

キー名	型
-----	---

ID	integer
----	---------

10. 「ODataV2Delete」タスクをクリックし、「マッピング設定」ボタンをクリックし、次のように各項目を紐づけ、「決定」ボタンをクリックします。



11. 「新規保存」リンクをクリックし、新規保存ダイアログを表示します。ダイアログに次のような内容を入力し「決定」ボタンをクリックします。

フロー定義ID	ODataV2Delete
フロー定義名 標準	ODataV2Delete
フローカテゴリID	SampleCategory

12. 「確認」ダイアログで「決定」ボタンをクリックします。

これでフロー定義の作成が終了です。

動作確認

最後に動作確認を行います。IM-LogicDesigner のデバッグ機能を使ってフローを実行し、OData サービスのデータが削除されることを確認します。

13. 「ODataV2Delete」フローの「編集」アイコンをクリックします。
14. 「ロジックフロー定義編集」画面の「デバッグ」リンクをクリックします。
15. 「ロジックフロー定義のデバッグ」画面の「実行」リンクをクリックします。
16. 「デバッグの実行」ダイアログが表示されるので、次のように入力し「実行」ボタンをクリックし、「確認」ダイアログで「決定」ボタンをクリックします。

ID 8

17. ブラウザから「[https://services.odata.org/V2/\(S\(im_odata\)\)/OData/OData.svc/Products\(8\)](https://services.odata.org/V2/(S(im_odata))/OData/OData.svc/Products(8))」にアクセスします。

以下のように表示され、ID=8 のエンティティが削除され存在しないことが確認できます。

```
<error xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <code/>
  <message xml:lang="en-US">Resource not found for the segment 'Products'.</message>
</error>
```

まとめ

以上で [OData\(V2\) サービスのエンティティ操作](#) のチュートリアルはすべて完了です。

これまでのチュートリアルを通して、以下のことを学びました。

- OData(V2) サービスのエンティティを操作する。
 - 登録
 - 参照
 - 更新
 - 削除

次章 [OData\(V4\) サービスのエンティティ操作](#) では、OData(V4) サービスのエンティティを操作するチュートリアルを用意しています。

- チュートリアル
 - ゴールの説明
 - 準備
 - エンティティの登録
 - エンティティの取得
 - シングルエンティティの取得
 - エンティティの更新
 - エンティティの削除
 - まとめ

チュートリアル

このチュートリアルが終了すると、次のような事ができます。

- OData(V4) サービスのエンティティを操作する。
 - 登録
 - 参照
 - 更新
 - 削除

ゴールの説明

このチュートリアルでは、以下で公開されている OData(V4) リファレンスサービスを利用し、製品エンティティの操作を行います。



コラム

[https://services.odata.org/Experimental/OData/\(S\(im_odata\)\)/OData.svc](https://services.odata.org/Experimental/OData/(S(im_odata))/OData.svc)

1. エンティティの登録
 - エンティティを登録するユーザ定義タスクを作成します。
 - 動作確認のためのフロー定義を作成します。
 - 動作確認を行います。
2. エンティティの取得
 - エンティティを取得するユーザ定義タスクを作成します。
 - 動作確認のためのフロー定義を作成します。
 - 動作確認を行います。
3. シングルエンティティの取得
 - シングルエンティティを取得するユーザ定義タスクを作成します。
 - 動作確認のためのフロー定義を作成します。
 - 動作確認を行います。
4. エンティティの更新
 - エンティティを更新するユーザ定義タスクを作成します。
 - 動作確認のためのフロー定義を作成します。
 - 動作確認を行います。
5. エンティティの削除

- エンティティを削除するユーザ定義タスクを作成します。
- 動作確認のためのフロー定義を作成します。
- 動作確認を行います。

このような順番で OData(V4) サービスのエンティティの操作方法や作業の流れを説明します。

準備

チュートリアルの準備に書かれている準備が行われていることを確認してください。

エンティティの登録

IM-LogicDesignerのユーザ定義タスクを作成します。作成するのは「OData(V4)定義」です。

後述の「メソッド」を「POST」として作成することで OData サービスのエンティティの登録を行うようにします。

OData(V4)定義の作成

1. intra-mart Accel Platform にテナント管理者またはLogicDesigner管理者でログインします。
2. 「サイトマップ」→「LogicDesigner」→「OData(V4)定義新規作成」を開きます。
3. 「サービスURL」に「[https://services.odata.org/Experimental/OData/\(S\(im_odata\)\)/OData.svc](https://services.odata.org/Experimental/OData/(S(im_odata))/OData.svc)」を設定します。

OData(V4)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザーカテゴリ *

検索 新規作成

ユーザーカテゴリID *

ユーザーカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

返却値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

OData(V4)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="削除"/>

メソッド * GET シングルエンティティ

クエリオプション

クエリオプション名 *	クエリオプション値 *	削除
-------------	-------------	----

エンティティセット名 *

入力返却値取得 *

i コラム

本チュートリアルでは必要ありませんが、利用する OData サービス に認証がかけられている場合、必要に応じて「認証種別」を「BASIC認証」または「OAuth認証」に設定し、認証情報を設定してください。

i コラム

OData サービスが独自のリクエストヘッダで認証を行う場合、以下のようにリクエストヘッダを追加して認証情報を設定してください。

リクエストヘッダ		+ 追加	
ヘッダ名 *	ヘッダ値 *		削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0		✕
Authorization	my custom authorization		✕

- リクエストヘッダ名に独自のリクエストヘッダ名を設定（上記画像では「Authorization」を設定）
- リクエストヘッダ値に認証情報を設定（上記画像では「my custom authorization」を設定）

i コラム

OData サービスが独自のリクエストヘッダで認証を行い、かつフロー実行時にリクエストヘッダを設定する必要がある場合、次のようにして認証情報を設定してください。

入力値

+ string + integer + date + object

boolean + 追加 配下に配置する

JSON入力

配列型にする キー名を変更 型を変更

削除 ✕ 全削除

- ▾ body <object> *
- ID <integer>
- Name <string>
- Description <string>
- ReleaseDate <sqltimestamp> *
- DiscontinuedDate <sqltimestamp>
- Rating <short> *
- Price <double> *
- authorization <string>

返却値

+ string + integer + date + object

boolean + 追加 配下に配置する

JSON入力

配列型にする キー名を変更 型を変更

削除 ✕ 全削除

- ▾ data <object>
- ID <integer>
- Name <string>
- Description <string>
- ReleaseDate <sqltimestamp>
- DiscontinuedDate <sqltimestamp>
- Rating <short>
- Price <double>

OData(V4)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ		+ 追加	
ヘッダ名 *	ヘッダ値 *		削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0		✕
Authorization	\${authorization}		✕

- 入力値に独自のプロパティを追加（上記の画像例では「authorization」）
- リクエストヘッダ値に「\${authorization}」を設定
 - 「\${authorization}」を設定することで、実行時に入力値「authorization」プロパティに渡された値を設定できます

4. 「エンティティセット名」の横にあるリロードアイコンをクリックします。

OData(V4)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザーカテゴリ *

検索 新規作成

ユーザーカテゴリID *

ユーザーカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

返却値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

OData(V4)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="削除"/>

メソッド * GET シングルエンティティ

クエリオプション

クエリオプション名 *	クエリオプション値 *	削除
-------------	-------------	----

エンティティセット名 *

入力返却値取得 *

5. 「エンティティセット名」に「Products」を選択します。

OData(V4)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザーカテゴリ *

検索 新規作成

ユーザーカテゴリID *

ユーザーカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

返却値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

OData(V4)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="削除"/>

メソッド * シングルエンティティ

クエリオプション

クエリオプション名 *	クエリオプション値 *	削除
-------------	-------------	----

エンティティセット名 *

入力返却値取得 *

6. 「メソッド」に「POST」を選択します。

OData(V4)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザーカテゴリ *

🔍 検索 📄 新規作成

ユーザーカテゴリID *

ユーザーカテゴリ名

ソート番号 *

アイコン

🔍 選択 🗑️ クリア

入力値

+ string + integer + date + object

boolean ▼ + 追加 配下に配置する

JSON入力

配列型にする 📌 キー名を変更 🔄 型を変更

🗑️ 削除 ✖️ 全削除

返却値

+ string + integer + date + object

boolean ▼ + 追加 配下に配置する

JSON入力

配列型にする 📌 キー名を変更 🔄 型を変更

🗑️ 削除 ✖️ 全削除

OData(V4)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	✖️

メソッド * **POST** ▼

エンティティセット名 * 🔄

入力返却値取得 *

7. 「入力/返却値取得」の横にある「取得」ボタンをクリックします。

OData(V4)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザーカテゴリ *

検索 新規作成

ユーザーカテゴリID *

ユーザーカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

- ▾ body <object> *
 - ID <integer>
 - Name <string>
 - Description <string>
 - ReleaseDate <sqltimestamp> *
 - DiscontinuedDate <sqltimestamp>
 - Rating <short> *
 - Price <double> *

返却値

+ string + integer + date + object

boolean 配下に配置する

配列型にする

- ▾ data <object>
 - ID <integer>
 - Name <string>
 - Description <string>
 - ReleaseDate <sqltimestamp>
 - DiscontinuedDate <sqltimestamp>
 - Rating <short>
 - Price <double>

OData(V4)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="X"/>

メソッド *

エンティティセット名 *

入力返却値取得 *

i コラム

「入力値」ツリーの「body」配下にあるプロパティが登録するデータ、「返却値」ツリーの「data」配下にあるプロパティが、登録されたデータを表します。

8. その他の項目は以下のように入力してください。

ユーザー定義ID ODataV4Post

ユーザ定義名	ODataV4Post
ユーザカテゴリID	SampleCategory

- 「登録」ボタンをクリックします。
 - 確認ダイアログで「OK」ボタンをクリックします。
- これで「OData(V4)定義」の作成は終了です。

フロー定義の作成

続いてフロー定義の作成を行います。

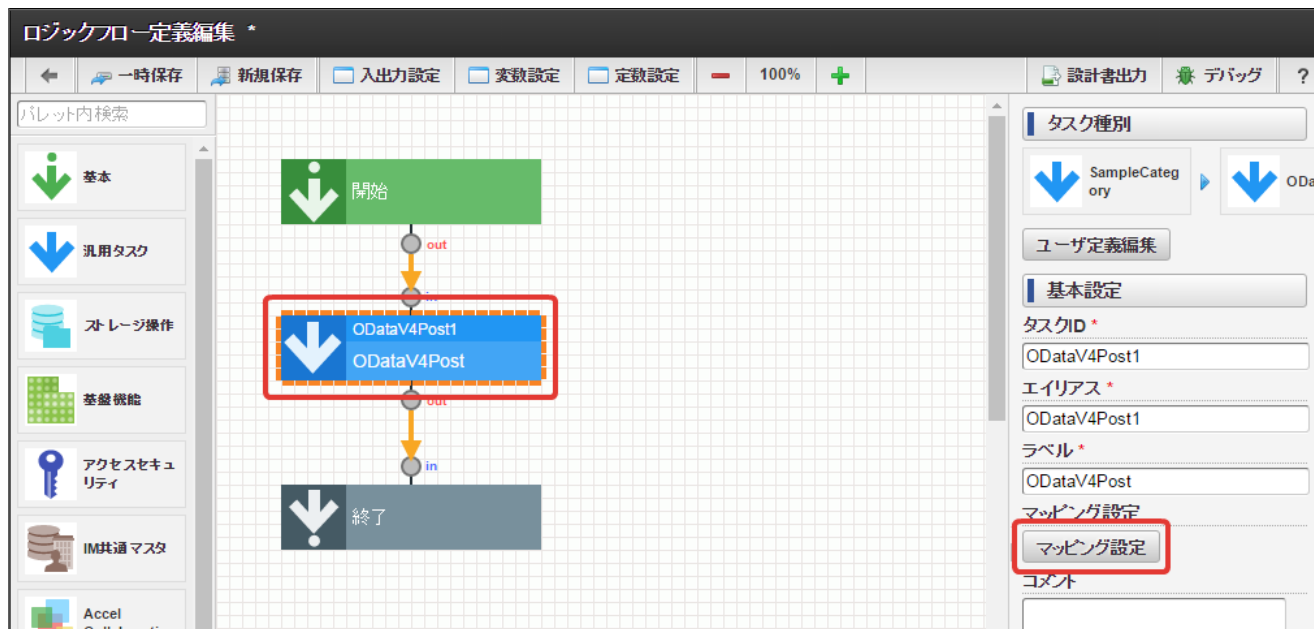
- 「サイトマップ」→「LogicDesigner」→「フロー定義」を開きます。
- 「新規作成」リンクをクリックし、「ロジックフロー定義編集」画面を表示します。
- 「入出力設定」リンクをクリックし、「入力」に次のような項目を追加し、「決定」ボタンをクリックします。

キー名	型
ID	integer
Name	string
Description	string
ReleaseDate	sqltimestamp
DiscontinuedDate	sqltimestamp
Rating	short
Price	double

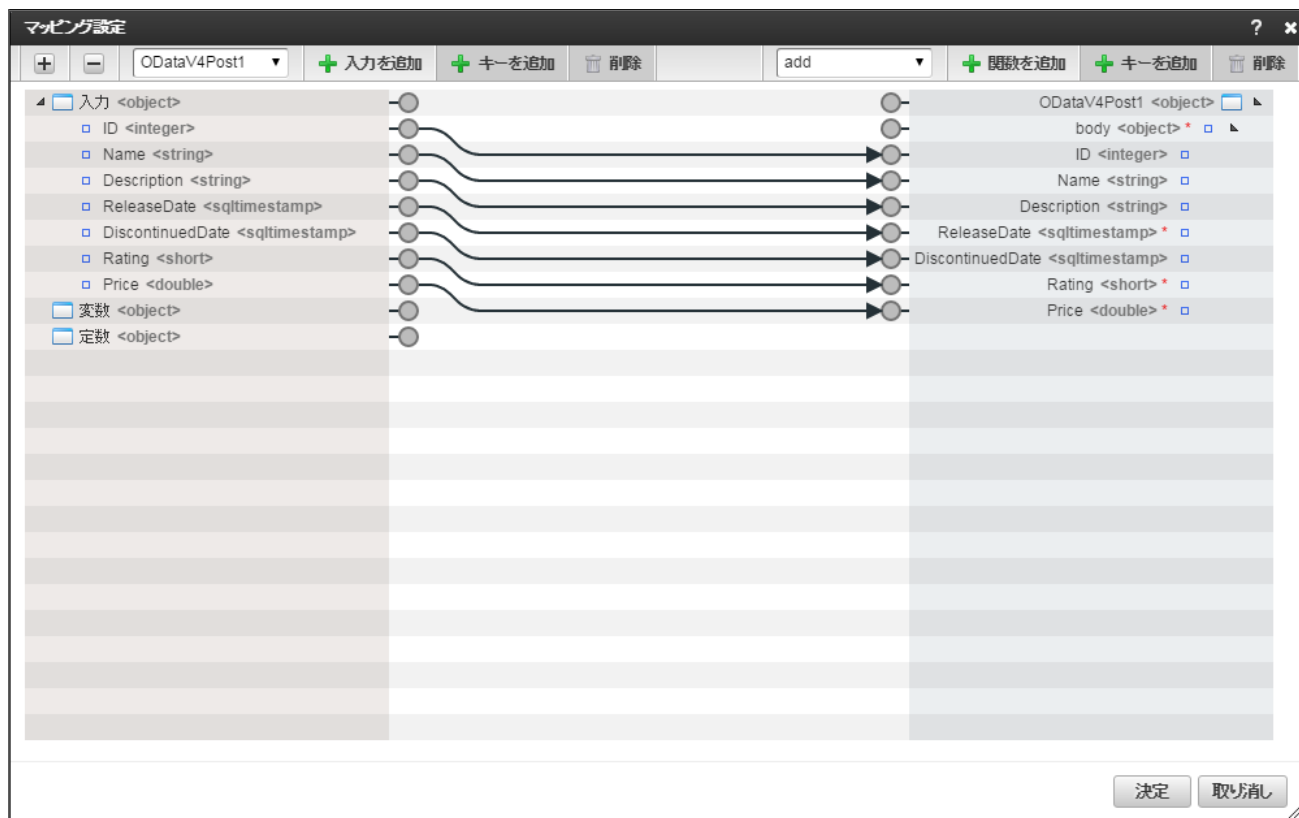
- ロジックフロー定義編集画面のデザイナーに、先ほど作成した「ODataV4Post」タスクを追加し、「開始」「終了」と接続します。



- 「ODataV4Post」タスクをクリックし、「マッピング設定」ボタンをクリックします。



16. 「入力」の各項目と、「ODataV4Post」の各項目を紐づけ、「決定」ボタンをクリックします。



17. 「新規保存」リンクをクリックし、新規保存ダイアログを表示します。ダイアログに次のような内容を入力し「決定」ボタンをクリックします。

フロー定義ID ODataV4Post
 フロー定義名 標準 ODataV4Post
 フローカテゴリID SampleCategory

18. 「確認」ダイアログで「決定」ボタンをクリックします。

動作確認

最後に動作確認を行います。IM-LogicDesigner のデバッグ機能を使ってフローを実行し、OData サービスにエンティティが作成されることを確認します。

19. 「ODataV4Post」フローの「編集」アイコンをクリックします。
20. 「ロジックフロー定義編集」画面の「デバッグ」リンクをクリックします。
21. 「ロジックフロー定義のデバッグ」画面の「実行」リンクをクリックします。
22. 「デバッグの実行」ダイアログが表示されるので、次のように入力し「実行」ボタンをクリックし、「確認」ダイアログで「決定」ボタンをクリックします。

ID	11
Name	テスト(Name)
Description	テスト(Description)
ReleaseDate	2016-12-01T10:00:00Z
DiscontinuedDate	2017-12-01T10:00:00Z
Rating	100
Price	1000

23. これで、OData サービスにエンティティが作成されました。

本当に作成されたかどうかを確認します。

24. ブラウザから「[https://services.odata.org/Experimental/OData/\(S\(im_odata\)\)/OData.svc/Products](https://services.odata.org/Experimental/OData/(S(im_odata))/OData.svc/Products)」にアクセスします。
25. フローのデバッグ画面で入力した値でエンティティが作成されていることを確認できます。

```

    Price: 1.01
  },
- {
  @odata.type: "#ODataDemo.FeaturedProduct",
  ID: 10,
  Name: "Coffee",
  Description: "Bulk size can of instant coffee",
  ReleaseDate: "1982-12-31T00:00:00Z",
  DiscontinuedDate: null,
  Rating: 1,
  Price: 6.99
},
- {
  ID: 11,
  Name: "テスト(Name)",
  Description: "テスト(Description)",
  ReleaseDate: "2016-12-01T10:00:00Z",
  DiscontinuedDate: "2017-12-01T10:00:00Z",
  Rating: 100,
  Price: 1000
}

```

エンティティの取得

次に、エンティティを参照する IM-LogicDesigner のユーザ定義タスクを作成します。作成するのは「OData(V4)定義」です。

「メソッド」を「GET」として作成することで OData サービスのエンティティの取得を行うようにします。

OData(V4)定義の作成

エンティティの登録の 1 から 5 までの手順を繰り返してください。

1. 「メソッド」に「GET」を選択します。
2. 「クエリオプション」の横にある「追加」アイコンをクリックし、以下のクエリオプションを追加します。
 - 「クエリオプション名」を「\$orderby」
 - 「クエリオプション値」を「ID」



コラム

「\$orderby」に「ID」を指定することで、IDでソートされた結果を取得する事が出来ます。
クエリオプションの詳細については以下のドキュメントを参照してください。

- [System Query Options](#)

3. 「入力/返却値取得」の横にある「取得」ボタンをクリックします。

OData(V4)定義編集 *

ユーザー定義一覧

ユーザー定義共通設定

ユーザー定義ID *

バージョン * 1

ユーザー定義名 *
 標準 *
 日本語
 英語
 中国語 (中華人民共和国)

ユーザーカテゴリ *
 検索
 ユーザーカテゴリID *
 ユーザーカテゴリ名

ソート番号 *

アイコン 選択

入力値
 boolean 配下に配置する
 配列型にする

返却値
 boolean 配下に配置する
 配列型にする


```

data <object[]>
  ID <integer>
  Name <string>
  Description <string>
  ReleaseDate <sqltimestamp>
  DiscontinuedDate <sqltimestamp>
  Rating <short>
  Price <double>
    
```

OData(V4)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="X"/>

メソッド * シングルエンティティ

クエリオプション

クエリオプション名 *	クエリオプション値 *	削除
\$orderby	ID	<input type="button" value="X"/>

エンティティセット名 *

入力返却値取得 *



コラム

「返却値」ツリーの「data」配下にあるプロパティが取得するデータを表します。



コラム

「返却値」ツリーの「data」配下にあるプロパティは、選択後「削除」をクリックすることで削除できます。不要なプロパティを削除することで、余計なプロパティの取得を防ぎ、実行時の負荷を軽減できます。

4. その他の項目は以下のように入力してください。

ユーザ定義ID	ODataV4Get
ユーザ定義名	ODataV4Get
ユーザカテゴリID	SampleCategory

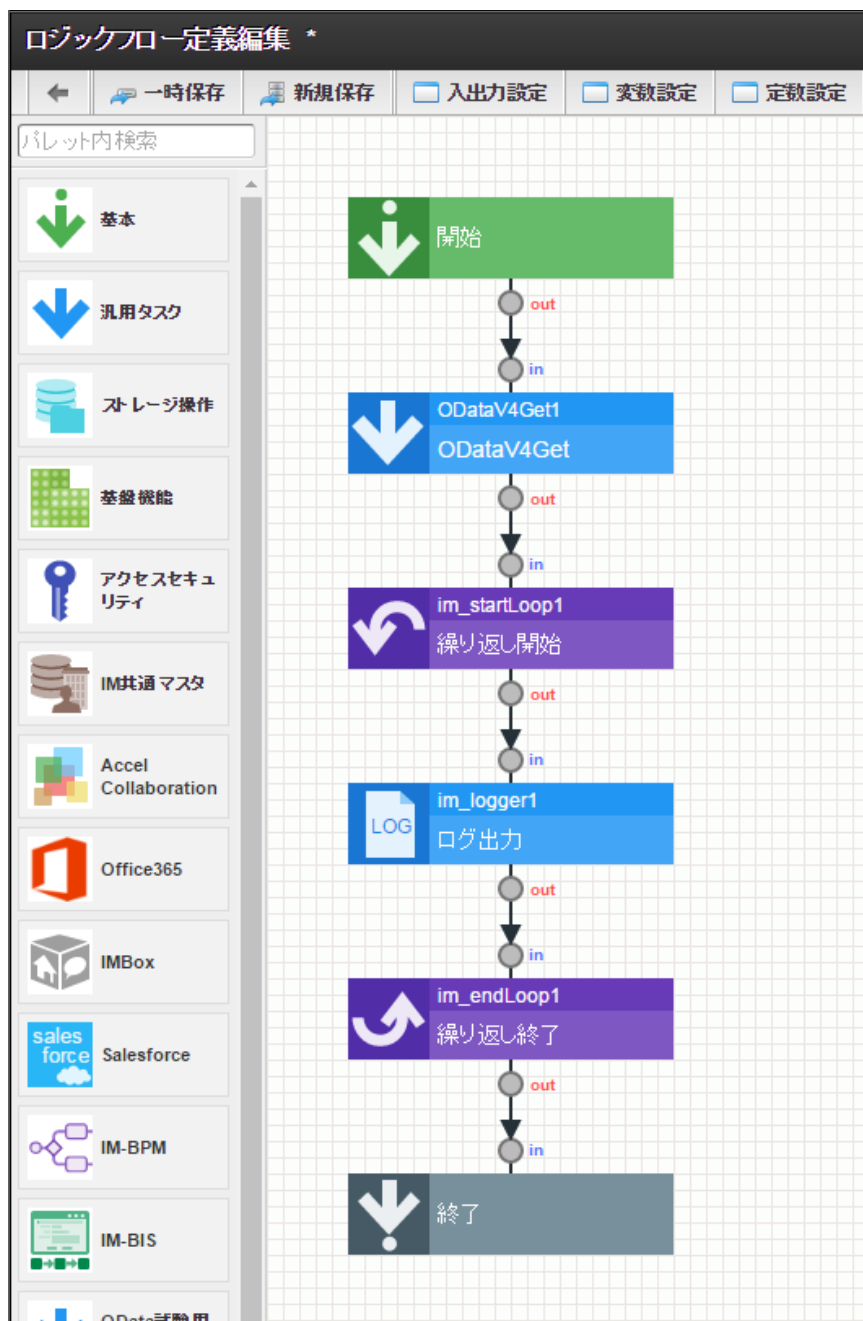
5. 「登録」ボタンをクリックします。
6. 確認ダイアログで「OK」ボタンをクリックします。

これで「OData(V4)定義」の作成は終了です。

フロー定義の作成

続いてフロー定義の作成を行います。

7. 「サイトマップ」→「LogicDesigner」→「フロー定義」を開きます。
8. 「新規作成」リンクをクリックし、「ロジックフロー定義編集」画面を表示します。
9. ロジックフロー定義編集画面のデザイナーに、先ほど作成した「ODataV4Get」タスクと、「繰り返し開始」「繰り返し終了」「ログ出力」タスクを追加し、「開始」「終了」と接続します。



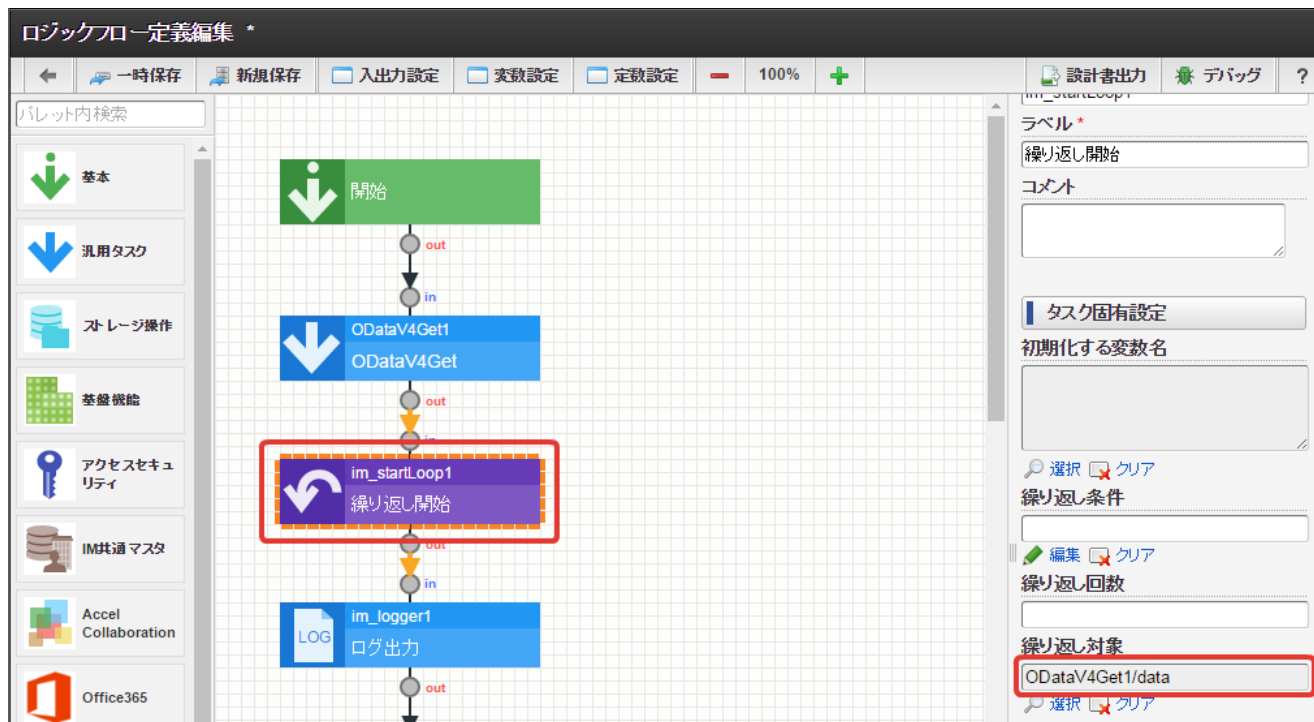
10. 「定数設定」リンクをクリックし、次のような項目を追加し、「決定」ボタンをクリックします。

- 「定数ID」を「e1」
- 「定数値」を以下

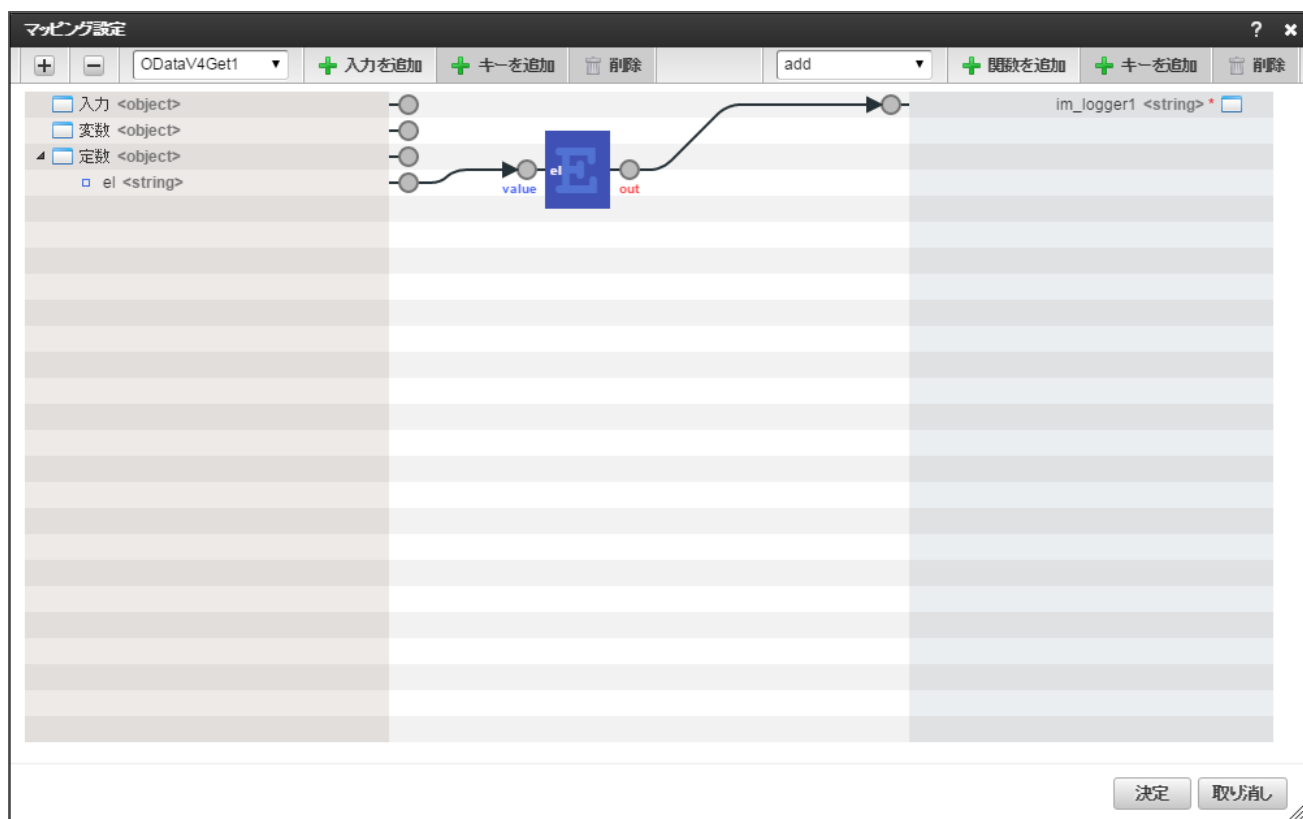
```

ID:${im_startLoop1.item.ID}
Name:${im_startLoop1.item.Name}
Description:${im_startLoop1.item.Description}
ReleaseDate:${im_startLoop1.item.ReleaseDate}
DiscontinuedDate:${im_startLoop1.item.DiscontinuedDate}
Rating:${im_startLoop1.item.Rating}
Price:${im_startLoop1.item.Price}
    
```

11. 「繰り返し開始」タスクをクリックし、「繰り返し対象」に「ODataV4Get1/data」を設定します。



12. 「ログ出力」タスクをクリックし、「マッピング設定」ボタンをクリックし、次のように各項目を紐づけ、「決定」ボタンをクリックします。



13. 「新規保存」リンクをクリックし、新規保存ダイアログを表示します。ダイアログに次のような内容を入力し「決定」ボタンをクリックします。

フロー定義ID ODataV4Get
 フロー定義名 標準 ODataV4Get
 フローカテゴリID SampleCategory

14. 「確認」ダイアログで「決定」ボタンをクリックします。

これでフロー定義の作成が終了です。

動作確認

最後に動作確認を行います。IM-LogicDesigner のデバッグ機能を使ってフローを実行し、OData サービスのデータが取得されることを確認します。

15. 「ODataV4Get」フローの「編集」アイコンをクリックします。
16. 「ロジックフロー定義編集」画面の「デバッグ」リンクをクリックします。
17. 「ロジックフロー定義のデバッグ」画面の「実行」リンクをクリックします。
18. 「デバッグの実行」ダイアログが表示されるので、「実行」ボタンをクリックし、「確認」ダイアログで「決定」ボタンをクリックします。
19. 以下のログが出力されます。

```
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:0
Name:Bread
Description:Whole grain bread
ReleaseDate:1992-01-01 09:00:00.0
DiscontinuedDate:
Rating:4
Price:2.5
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:1
Name:Milk
Description:Low fat milk
ReleaseDate:1995-10-01 09:00:00.0
DiscontinuedDate:
Rating:3
Price:3.5
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:2
Name:Vint soda
Description:Americana Variety - Mix of 6 flavors
ReleaseDate:2000-10-01 09:00:00.0
DiscontinuedDate:
Rating:3
Price:20.9
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:3
Name:Havina Cola
Description:The Original Key Lime Cola
ReleaseDate:2005-10-01 09:00:00.0
DiscontinuedDate:2006-10-01 09:00:00.0
Rating:3
Price:19.9
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:4
Name:Fruit Punch
Description:Mango flavor, 8.3 Ounce Cans (Pack of 24)
ReleaseDate:2003-01-05 09:00:00.0
DiscontinuedDate:
Rating:3
Price:22.99
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:5
Name:Cranberry Juice
Description:16-Ounce Plastic Bottles (Pack of 12)
ReleaseDate:2006-08-04 09:00:00.0
DiscontinuedDate:
Rating:3
Price:22.8
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:6
Name:Pink Lemonade
Description:36 Ounce Cans (Pack of 3)
ReleaseDate:2006-11-05 09:00:00.0
DiscontinuedDate:
Rating:3
Price:18.8
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:7
Name:DVD Player
Description:1080P Upconversion DVD Player
```

```

Description:100W Upconversion DVD Player
ReleaseDate:2006-11-15 09:00:00.0
DiscontinuedDate:
Rating:5
Price:35.88
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:8
Name:LCD HDTV
Description:42 inch 1080p LCD with Built-in Blu-ray Disc Player
ReleaseDate:2008-05-08 09:00:00.0
DiscontinuedDate:
Rating:3
Price:1088.8
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:9
Name:Lemonade
Description:Classic, refreshing lemonade (Single bottle)
ReleaseDate:1970-01-01 09:00:00.0
DiscontinuedDate:
Rating:7
Price:1.01
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:10
Name:Coffee
Description:Bulk size can of instant coffee
ReleaseDate:1982-12-31 09:00:00.0
DiscontinuedDate:
Rating:1
Price:6.99
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:11
Name:テスト(Name)
Description:テスト(Description)
ReleaseDate:2016-12-01 19:00:00.0
DiscontinuedDate:2017-12-01 19:00:00.0
Rating:100
Price:1000.0

```

20. ブラウザから「[https://services.odata.org/Experimental/OData/\(S\(im_odata\)\)/OData.svc/Products](https://services.odata.org/Experimental/OData/(S(im_odata))/OData.svc/Products)」にアクセスします。

ブラウザに表示されるデータと、出力されたログの内容が一致することが確認できます。

シングルエンティティの取得

次に、シングルエンティティを参照する IM-LogicDesigner のユーザ定義タスクを作成します。作成するのは「OData(V4)定義」です。

「メソッド」を「GET」、「シングルエンティティ」を選択することで OData サービスのシングルエンティティの取得を行うようにします。

OData(V4)定義の作成

[エンティティの登録](#)の 1 から 5 までの手順を繰り返してください。

1. 「メソッド」に「GET」を選択します。
2. 「メソッド」の横にある「シングルエンティティ」を選択します。
3. 「入力/返却値取得」の横にある「取得」ボタンをクリックします。

The screenshot shows the 'OData(V4)定義編集' interface. The 'OData(V4)情報' section is expanded, showing the following configuration:

- サービスURL: `http://services.odata.org/Experimental/OData/(S(im_notice))/OData.svc`
- 認証種別: 認証なし
- メソッド: GET (with 'シングルエンティティ' checked)
- 入力返却値取得: 取得

The '入力値' (Input Values) tree shows a 'key' property with an 'ID' sub-property. The '返却値' (Return Values) tree shows a 'data' object with properties: ID, Name, Description, ReleaseDate, DiscontinuedDate, Rating, and Price.

i コラム

「入力値」ツリーの「key」配下にあるプロパティが取得するシングルエンティティを表すキープロパティとして扱われます。

i コラム

「返却値」ツリーの「data」配下にあるプロパティが取得するデータを表します。

i コラム

「返却値」ツリーの「data」配下にあるプロパティは、選択後「削除」をクリックすることで削除できます。不要なプロパティを削除することで、余計なプロパティの取得を防ぎ、実行時の負荷を軽減できます。

4. その他の項目は以下のように入力してください。

ユーザ定義ID	ODataV4GetSingle
ユーザ定義名	ODataV4GetSingle

ユーザカテゴリID SampleCategory

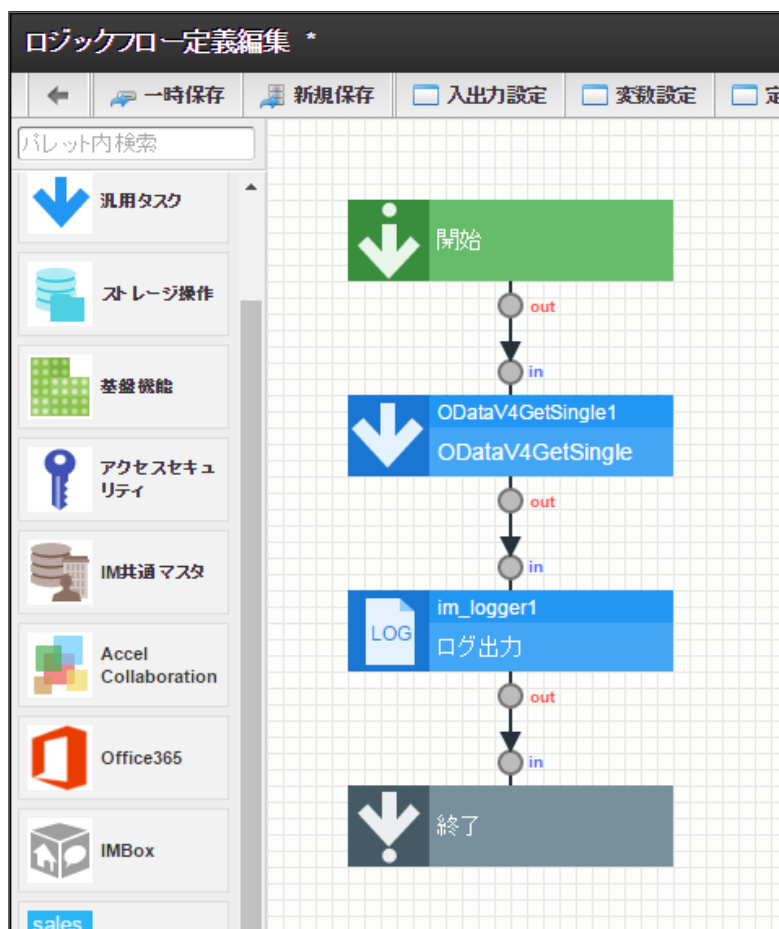
5. 「登録」 ボタンをクリックします。
6. 確認ダイアログで「OK」 ボタンをクリックします。

これで「OData(V4)定義」の作成は終了です。

フロー定義の作成

続いてフロー定義の作成を行います。

7. 「サイトマップ」→「LogicDesigner」→「フロー定義」を開きます。
8. 「新規作成」リンクをクリックし、「ロジックフロー定義編集」画面を表示します。
9. ロジックフロー定義編集画面のデザイナーに、先ほど作成した「ODataV4GetSingle」タスクと「ログ出力」タスクを追加し、「開始」「終了」と接続します。



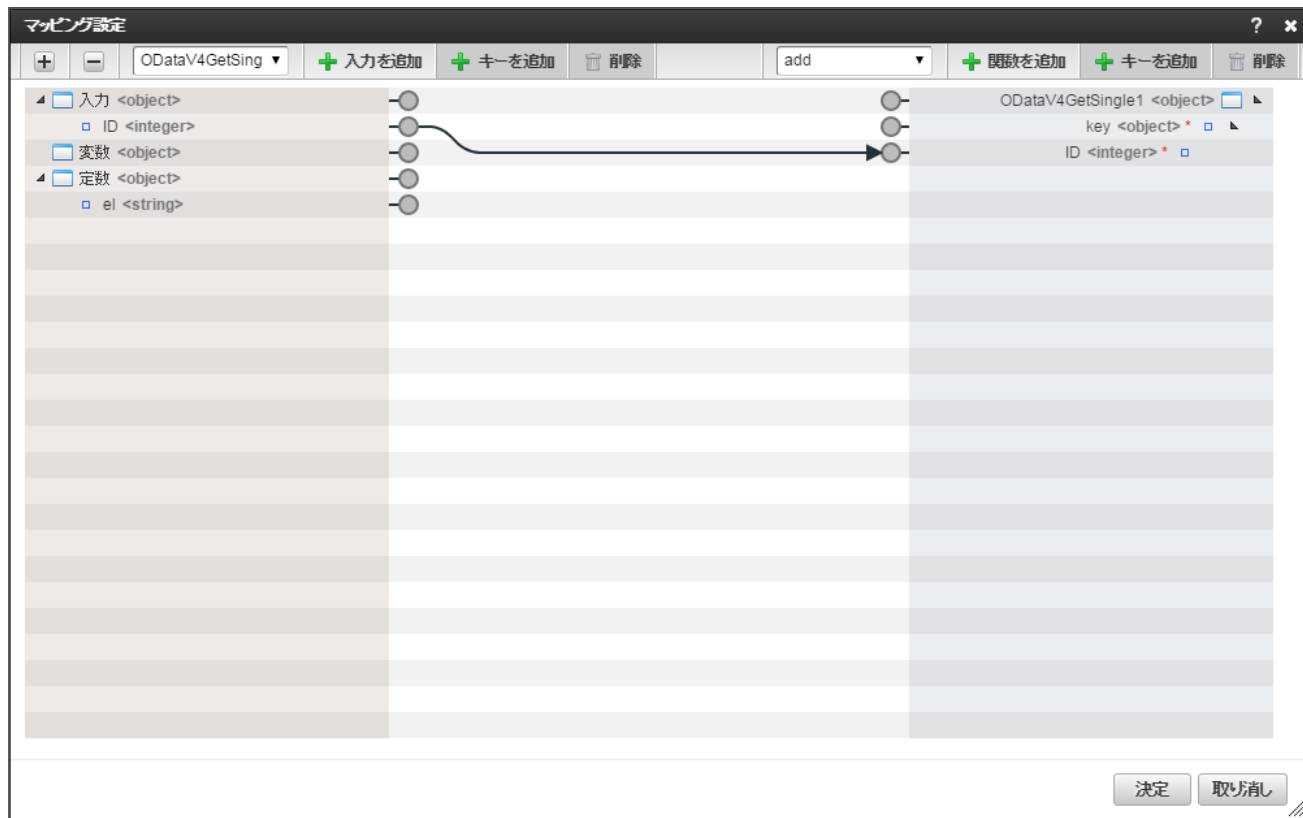
10. 「入出力設定」リンクをクリックし、「入力」に次のような項目を追加し、「決定」ボタンをクリックします。

キー名	型
ID	integer

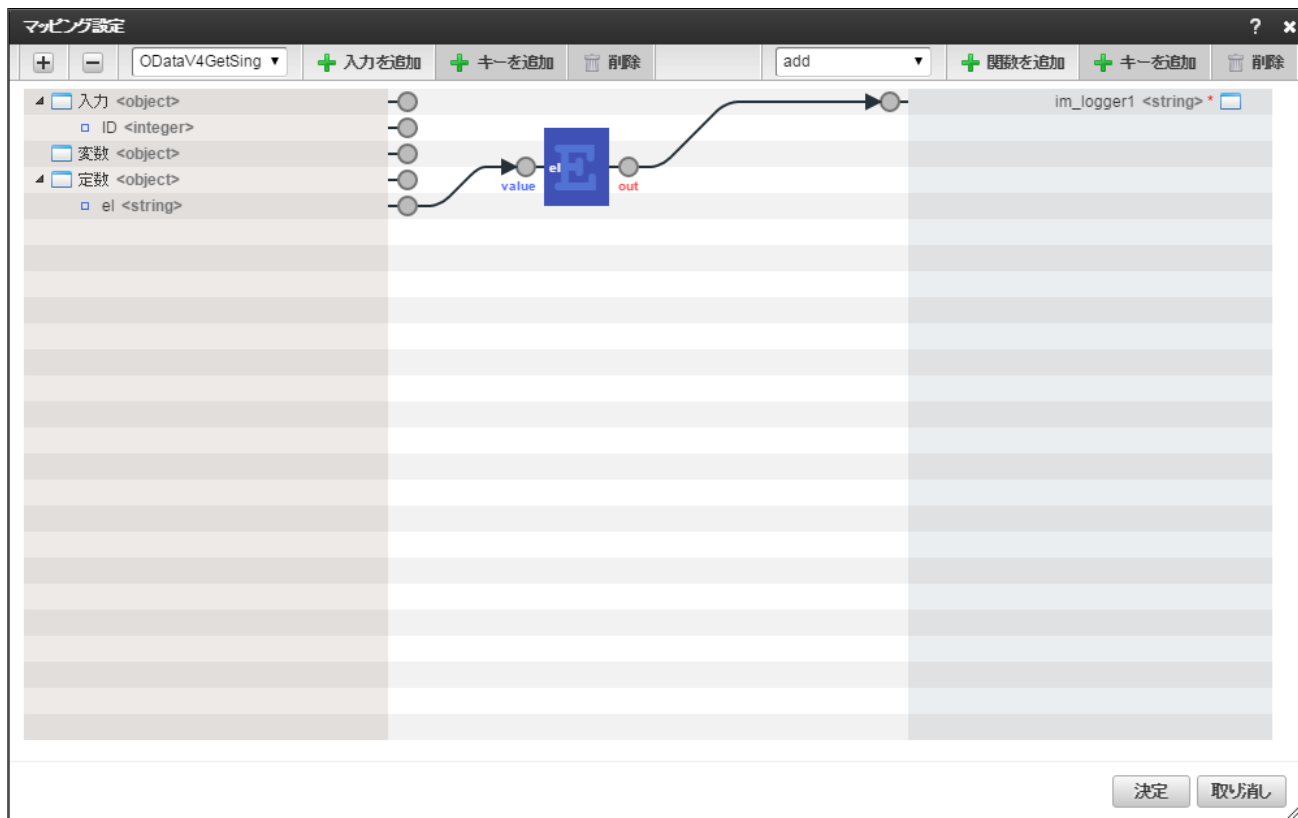
11. 「定数設定」リンクをクリックし、次のような項目を追加し、「決定」ボタンをクリックします。
 - 「定数ID」を「e1」
 - 「定数値」を以下

```
ID:${ODataV4GetSingle1.data.ID}
Name:${ODataV4GetSingle1.data.Name}
Description:${ODataV4GetSingle1.data.Description}
ReleaseDate:${ODataV4GetSingle1.data.ReleaseDate}
DiscontinuedDate:${ODataV4GetSingle1.data.DiscontinuedDate}
Rating:${ODataV4GetSingle1.data.Rating}
Price:${ODataV4GetSingle1.data.Price}
```

12. 「ODataV4GetSingle」タスクをクリックし、「マッピング設定」ボタンをクリックし、次のように各項目を紐づけ、「決定」ボタンをクリックします。



13. 「ログ出力」タスクをクリックし、「マッピング設定」ボタンをクリックし、次のように各項目を紐づけ、「決定」ボタンをクリックします。



14. 「新規保存」リンクをクリックし、新規保存ダイアログを表示します。ダイアログに次のような内容を入力し「決定」ボタンをクリックします。

フロー定義ID	ODataV4GetSingle
フロー定義名 標準	ODataV4GetSingle
フローカテゴリID	SampleCategory

15. 「確認」ダイアログで「決定」ボタンをクリックします。

これでフロー定義の作成が終了です。

動作確認

最後に動作確認を行います。IM-LogicDesigner のデバッグ機能を使ってフローを実行し、OData サービスのデータが取得されることを確認します。

16. 「ODataV4GetSingle」フローの「編集」アイコンをクリックします。
17. 「ロジックフロー定義編集」画面の「デバッグ」リンクをクリックします。
18. 「ロジックフロー定義のデバッグ」画面の「実行」リンクをクリックします。
19. 「デバッグの実行」ダイアログが表示されるので、次のように入力し「実行」ボタンをクリックし、「確認」ダイアログで「決定」ボタンをクリックします。

ID 7

20. 以下のログが出力されます。

```
[INFO] j.c.i.f.l.e.g.OutputLogTask - [] ID:7  
Name: DVD Player  
Description: 1080P Upconversion DVD Player  
ReleaseDate: 2006-11-15 09:00:00.0  
DiscontinuedDate:  
Rating: 5  
Price: 35.88
```

21. ブラウザから「[https://services.odata.org/Experimental/OData/\(S\(im_odata\)\)/OData.svc/Products\(7\)](https://services.odata.org/Experimental/OData/(S(im_odata))/OData.svc/Products(7))」にアクセスします。

ブラウザに表示されるデータと、出力されたログの内容が一致することが確認できます。

エンティティの更新

次に、エンティティを更新する IM-LogicDesigner のユーザ定義タスクを作成します。作成するのは「OData(V4)定義」です。

「メソッド」を「PUT」または「PATCH」として作成することで OData サービスのエンティティの更新を行うようにします。

コラム

「PUT」と「PATCH」はどちらもエンティティの更新を行いますが、「PUT」は入力プロパティの更新を行い、入力プロパティに与えられなかったプロパティは初期値に更新されます。

これとは逆に「PATCH」では入力プロパティに与えられたプロパティの更新のみを行い、入力プロパティに与えられなかったプロパティは更新されません。

本チュートリアルでは「メソッド」に「PATCH」を指定してエンティティの更新を行います。

OData(V4)定義の作成

[エンティティの登録](#)の 1 から 5 までの手順を繰り返してください。

1. 「メソッド」に「PATCH」を選択します。
2. 「入力/返却値取得」の横にある「取得」ボタンをクリックします。

OData(V4)定義編集 *

ユーザ定義一覧

ユーザ定義共通設定

ユーザ定義ID *

バージョン * 1

ユーザ定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザカテゴリ *

検索 新規作成

ユーザカテゴリID *

ユーザカテゴリ名

ソート番号 *

アイコン

選択 クリア

入力値

+ string + integer + date + object

boolean + 追加 配下に配置する

配列型にする キー名を変更 型を変更

- key <object> *
 - ID <integer> *
- body <object> *
 - ID <integer>
 - Name <string>
 - Description <string>
 - ReleaseDate <sqltimestamp>
 - DiscontinuedDate <sqltimestamp>
 - Rating <short>
 - Price <double>

返却値

+ string + integer + date + object

boolean + 追加 配下に配置する

配列型にする キー名を変更 型を変更

OData(V4)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="削除"/>

メソッド *

エンティティセット名 *

入力返却値取得 *



コラム

「入力値」ツリーの「key」配下にあるプロパティが更新するエンティティを表すキープロパティとして扱われます。



コラム

「入力値」ツリーの「body」配下にあるプロパティが更新するデータを表します。

3. その他の項目は以下のように入力してください。

ユーザ定義ID	ODataV4Patch
ユーザ定義名	ODataV4Patch
ユーザカテゴリID	SampleCategory

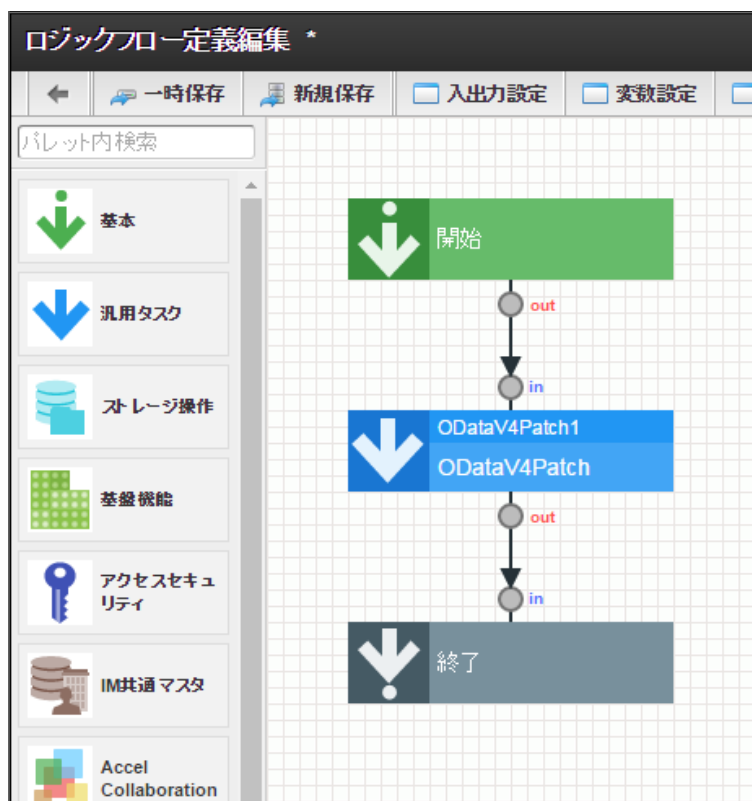
- 「登録」ボタンをクリックします。
- 確認ダイアログで「OK」ボタンをクリックします。

これで「OData(V4)定義」の作成は終了です。

フロー定義の作成

続いてフロー定義の作成を行います。

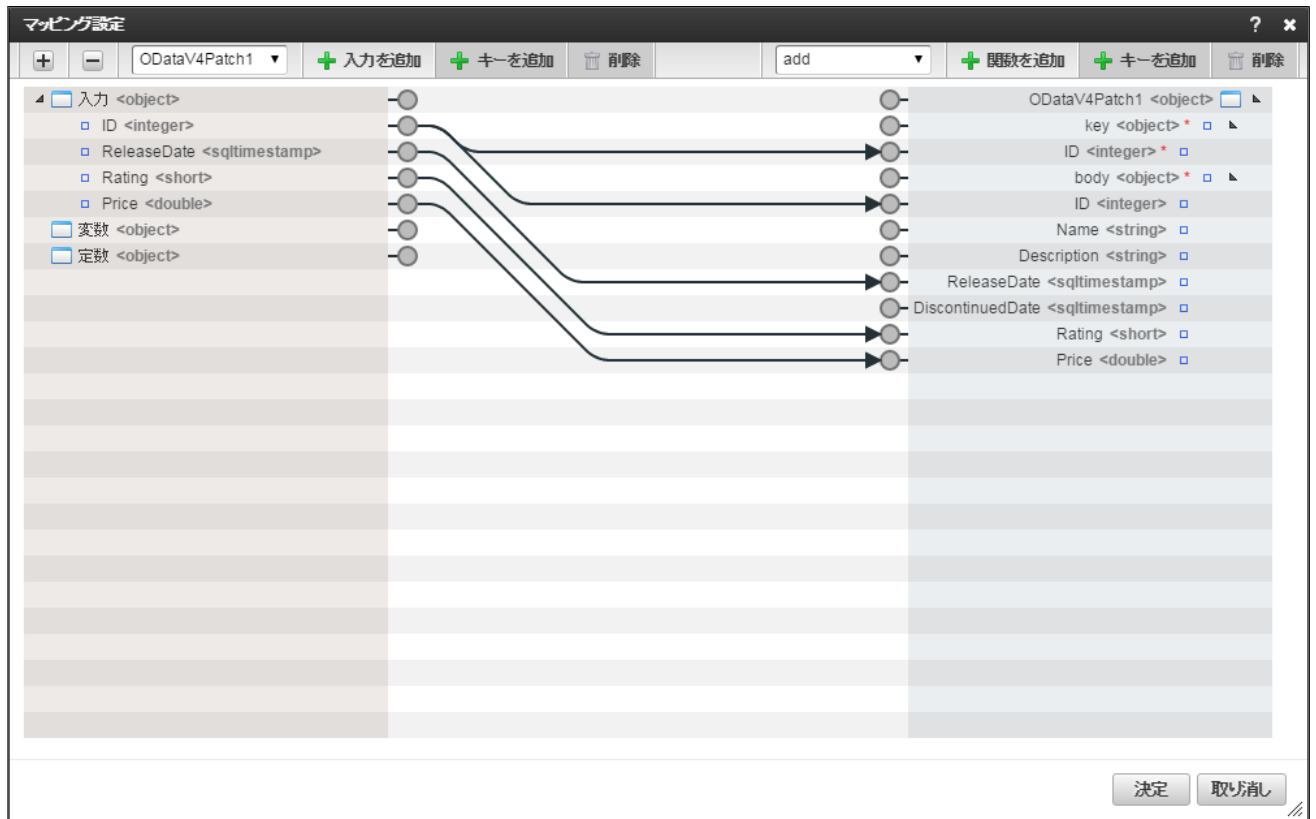
- 「サイトマップ」→「LogicDesigner」→「フロー定義」を開きます。
- 「新規作成」リンクをクリックし、「ロジックフロー定義編集」画面を表示します。
- ロジックフロー定義編集画面のデザイナーに、先ほど作成した「ODataV4Patch」タスクを追加し、「開始」「終了」と接続します。



- 「入出力設定」リンクをクリックし、「入力」に次のような項目を追加し、「決定」ボタンをクリックします。

キー名	型
ID	integer
ReleaseDate	sqltimestamp
Rating	short
Price	double

- 「ODataV4Patch」タスクをクリックし、「マッピング設定」ボタンをクリックし、次のように各項目を紐づけ、「決定」ボタンをクリックします。



11. 「新規保存」リンクをクリックし、新規保存ダイアログを表示します。ダイアログに次のような内容を入力し「決定」ボタンをクリックします。

フロー定義ID	ODataV4Patch
フロー定義名 標準	ODataV4Patch
フローカテゴリID	SampleCategory

12. 「確認」ダイアログで「決定」ボタンをクリックします。

これでフロー定義の作成が終了です。

動作確認

最後に動作確認を行います。IM-LogicDesigner のデバッグ機能を使ってフローを実行し、OData サービスのデータが更新されることを確認します。

13. 「ODataV4Patch」フローの「編集」アイコンをクリックします。
14. 「ロジックフロー定義編集」画面の「デバッグ」リンクをクリックします。
15. 「ロジックフロー定義のデバッグ」画面の「実行」リンクをクリックします。
16. 「デバッグの実行」ダイアログが表示されるので、次のように入力し「実行」ボタンをクリックし、「確認」ダイアログで「決定」ボタンをクリックします。

ID	3
ReleaseDate	2016-12-01T10:00:00Z
Rating	100
Price	1000

17. これで、OData サービスのエントリティが更新されました。

本当に作成されたかどうかを確認します。

18. ブラウザから「[https://services.odata.org/Experimental/OData/\(S\(im_odata\)\)/OData.svc/Products\(3\)](https://services.odata.org/Experimental/OData/(S(im_odata))/OData.svc/Products(3))」にアクセスし

ます。

19. フローのデバッグ画面で入力した値でエンティティが更新されていることを確認できます。

```
{
  @odata.context: "http://services.odata.org/Experimental/OData/(S(im_notice))/OData.svc/$metadata#Products/$entity",
  ID: 3,
  Name: "Havina Cola",
  Description: "The Original Key Lime Cola",
  ReleaseDate: "2016-12-01T10:00:00Z",
  DiscontinuedDate: "2016-10-01T00:00:00Z",
  Rating: 100,
  Price: 1000
}
```



コラム

更新処理では、キープロパティを変更することはできません。



コラム

入力データとして与えられた「ReleaseDate」「Rating」「Price」プロパティが更新され、入力データに与えられなかった「Name」、「Description」、「DiscontinuedDate」プロパティは更新されていないことが確認できます。

メソッドに「PUT」を指定した場合、入力データとして与えられた「ReleaseDate」「Rating」「Price」プロパティは同様に更新されますが、入力データに与えられなかった「Name」、「Description」、「DiscontinuedDate」プロパティは初期値に更新されます。

エンティティの削除

最後に、エンティティを削除する IM-LogicDesignerのユーザ定義タスクを作成します。作成するのは「OData(V4)定義」です。

「メソッド」を「DELETE」として作成することで OData サービスのエンティティの削除を行うようにします。

OData(V4)定義の作成

[エンティティの登録](#)の 1 から 5 までの手順を繰り返してください。

1. 「メソッド」に「DELETE」を選択します。
2. 「入力/返却値取得」の横にある「取得」ボタンをクリックします。

OData(V4)定義編集 *

ユーザ定義一覧

ユーザ定義共通設定

ユーザ定義ID *

バージョン * 1

ユーザ定義名 *

標準 *	<input type="text"/>
日本語	<input type="text"/>
英語	<input type="text"/>
中国語 (中華人民共和国)	<input type="text"/>

ユーザカテゴリ *

検索 新規作成

ユーザカテゴリID *

ユーザカテゴリ名

ソート番号 *

アイコン

入力値

+ string + integer + date + object

boolean + 追加 配下に配置する

配列型にする キー名を変更 型を変更

- key <object> *
 - ID <integer> *

返却値

+ string + integer + date + object

boolean + 追加 配下に配置する

配列型にする キー名を変更 型を変更

OData(V4)情報

サービスURL *

認証種別 * 認証なし BASIC認証 OAuth認証

リクエストヘッダ

+ 追加

ヘッダ名 *	ヘッダ値 *	削除
User-Agent	LOGIC-DESIGNER INTRAMART/8.0 Version/8.0.0	<input type="button" value="削除"/>

メソッド *

エンティティセット名 *

入力返却値取得 *



コラム

「入力値」ツリーの「key」配下にあるプロパティが削除するエンティティを表すキープロパティとして扱われます。

3. その他の項目は以下のように入力してください。

ユーザ定義ID ODataV4Delete

ユーザ定義名 ODataV4Delete

ユーザカテゴリID SampleCategory

4. 「登録」 ボタンをクリックします。
5. 確認ダイアログで「OK」 ボタンをクリックします。

これで「OData(V4)定義」の作成は終了です。

フロー定義の作成

続いてフロー定義の作成を行います。

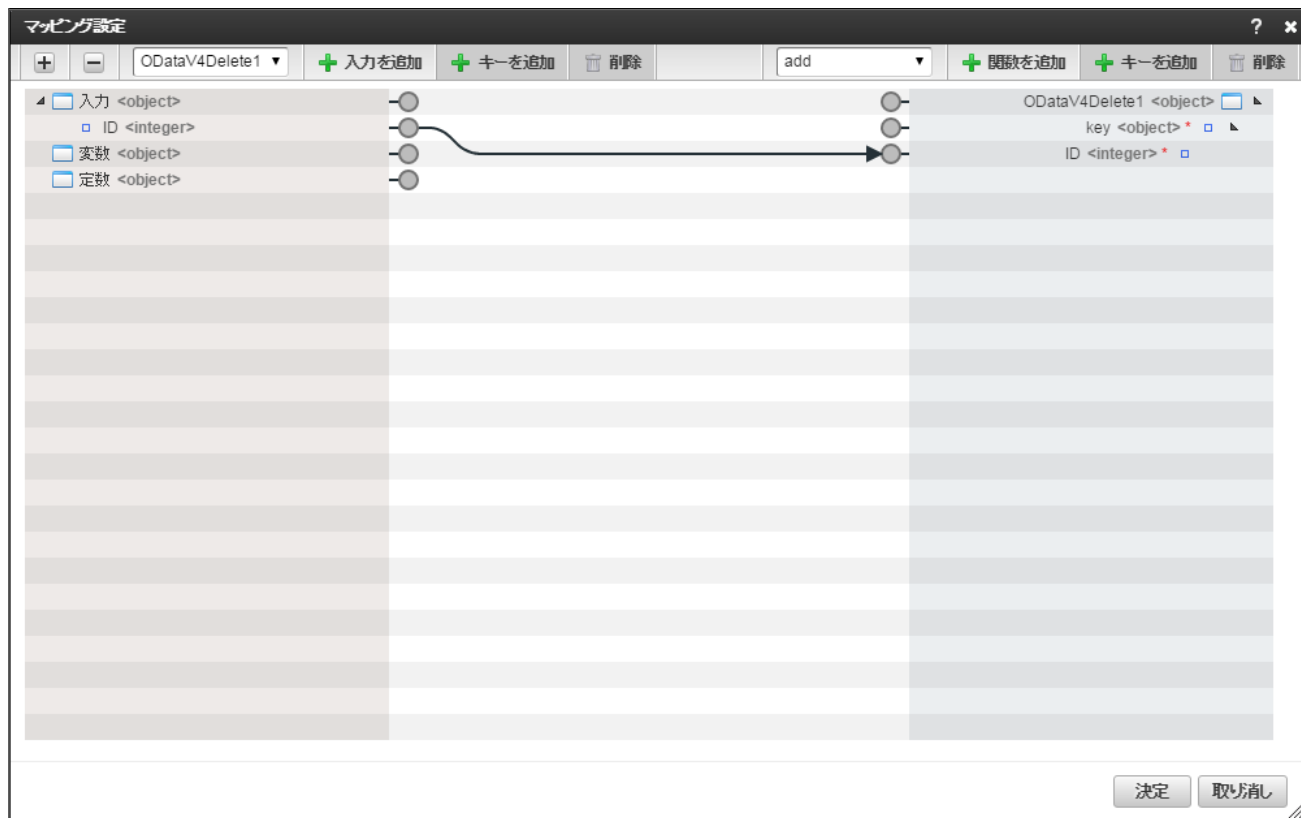
6. 「サイトマップ」→「LogicDesigner」→「フロー定義」を開きます。
7. 「新規作成」リンクをクリックし、「ロジックフロー定義編集」画面を表示します。
8. ロジックフロー定義編集画面のデザイナーに、先ほど作成した「ODataV4Delete」タスクを追加し、「開始」「終了」と接続します。



9. 「入出力設定」リンクをクリックし、「入力」に次のような項目を追加し、「決定」ボタンをクリックします。

キー名	型
ID	integer

10. 「ODataV4Delete」タスクをクリックし、「マッピング設定」ボタンをクリックし、次のように各項目を紐づけ、「決定」ボタンをクリックします。



11. 「新規保存」リンクをクリックし、新規保存ダイアログを表示します。ダイアログに次のような内容を入力し「決定」ボタンをクリックします。

フロー定義ID	ODataV4Delete
フロー定義名 標準	ODataV4Delete
フローカテゴリID	SampleCategory

12. 「確認」ダイアログで「決定」ボタンをクリックします。

これでフロー定義の作成が終了です。

動作確認

最後に動作確認を行います。IM-LogicDesigner のデバッグ機能を使ってフローを実行し、OData サービスのデータが削除されることを確認します。

13. 「ODataV4Delete」フローの「編集」アイコンをクリックします。
14. 「ロジックフロー定義編集」画面の「デバッグ」リンクをクリックします。
15. 「ロジックフロー定義のデバッグ」画面の「実行」リンクをクリックします。
16. 「デバッグの実行」ダイアログが表示されるので、次のように入力し「実行」ボタンをクリックし、「確認」ダイアログで「決定」ボタンをクリックします。

ID 10

17. ブラウザから「[https://services.odata.org/Experimental/OData/\(S\(im_odata\)\)/OData.svc/Products\(10\)](https://services.odata.org/Experimental/OData/(S(im_odata))/OData.svc/Products(10))」にアクセスします。

以下のように表示され、ID=10 のエンティティが削除され存在しないことが確認できます。

```
<m:error xmlns:m="http://docs.oasis-open.org/odata/ns/metadata">  
  <m:code/>  
  <m:message>Resource not found for the segment 'Products'.</m:message>  
</m:error>
```

まとめ

以上で *OData(V4) サービスのエンティティ操作* のチュートリアルはすべて完了です。

これまでのチュートリアルを通して、以下のことを学びました。

- OData(V4) サービスのエンティティを操作する。
 - 登録
 - 参照
 - 更新
 - 削除