AEM Forms: Rest API Integration as a Datasource

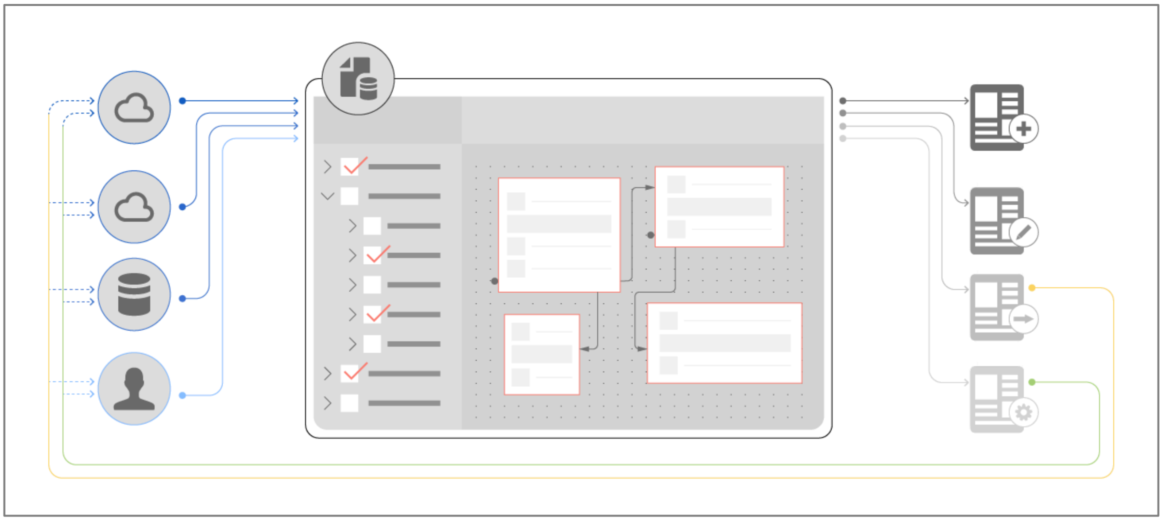
Samit Narula, Technical Architect, Adobe Partner Experience

Overview

The Swagger (OpenAPI) specification is quite detailed and defines various directives, constraints, and configurations to be used by an API Gateway that serves the API's consumers. However, for creating a datasource within AEM, only a subset of these Swagger directives and constraints need to be interpreted. AEM developers should be aware of which these are.

This whitepaper also discusses scenarios where APIs are missing documentation, or their documentation is based on other specifications, like RAML, and so on. Here, tools and techniques are available to create a Swagger file corresponding to such a REST API.

Data Integration in AEM Forms



Form Model Integration with Datasources

- Configure data sources
- Create form data model
- Test data model objects and services in form data model
- Use form data model in adaptive form workflows

Swagger requirements

The first step for configuring the datasource that corresponds to the REST API is to source a Swagger file. The developer may encounter the following cases for Swagger availability.

Already available as a published resource

The Swagger file is usually hosted on a public server that runs an API Management solution for a SaaS product. In this case, the API documentation is occasionally updated. However, the developer needs to keep track of the API documentation to note any deprecation or retirement of APIs. Though less frequent, these changes might result in the AEM datasource becoming out of sync with the hosted API due to mismatched lifecycles.

Sometimes the API is owned by the same organization and hosted on an accessible server. The lifecycles of the API documentation and the AEM Forms data model can be synchronized, using continuous integration and integration testing that is set up to account for the changes, and the data model and associated forms updated according to the AEM maintenance cycle.

Also, hosted APIs are detailed and their Swagger complex, in which case a subsection of the API can be extracted into a Swagger file on the filesystem and used.

Note: The AEM Forms data model may have issues with deeply nested complex objects used as parameters and defined inline within the Swagger file. It is recommended to use a single level of attributes within the parameters section and any complex objects extracted and referenced from the definitions section of the Swagger.

Authoring a handcrafted Swagger file

For REST APIs without any API documentation available, or for REST APIs documented as per RAML specification or some other non-Swagger specification. Authoring your Swagger file without using any tools is feasible for small and simple REST APIs. It is the developer's responsibility to keep the Swagger file updated with the changes in the hosted API's operations and data model, with the AEM Form Data Model being updated when the API and its Swagger changes.

When handcrafting the Swagger file, some external tools can simplify the process. The tools mentioned here are illustrative, and you can use others. A rough process can be:

1. Generate schema from JSON request or response for an API operation.
    - https://jsonschema.net/#/editor
2. Create Swagger using the above schema.
    - http://editor.swagger.io/#/

3. Test the API using the created Swagger.
    - https://swagger.io/swagger-ui/

The discussion on using these tools is outside the purview of this whitepaper.

Automated generation from code

Given a scenario where the AEM's developer organization has appropriate change request agreements with the product management of the SaaS solution, it is possible to generate a Swagger from the API code itself, if the design allows for it.

A Swagger resource for the API can be created during build time, using the following Maven plugin. This requires Swagger annotations to be inserted into the code.

```
<groupId>com.github.kongchen</groupId>
<artifactId>swagger-maven-plugin</artifactId>
```

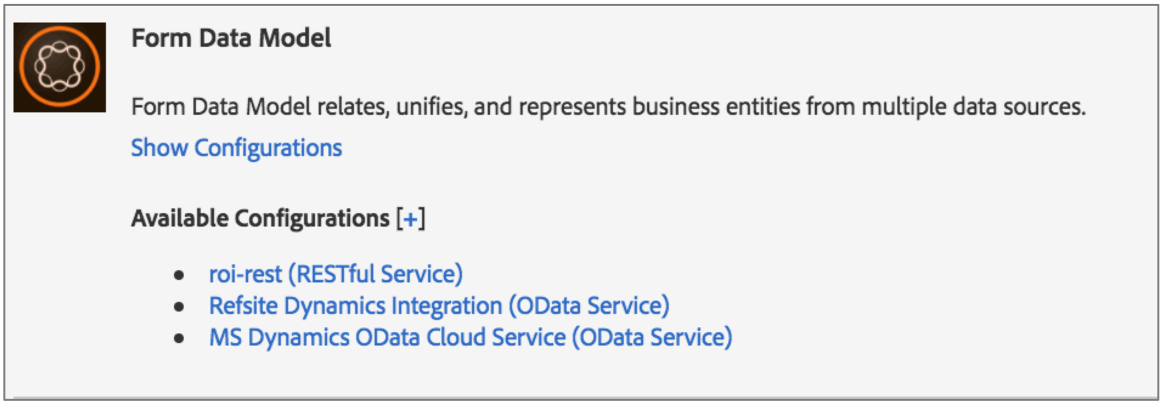
To generate Swagger resource at runtime where JAX-RS has been used to create the REST API, it is recommended to use the following Maven plugin, which again requires Swagger annotations in the code.

```
<groupId>io.swagger</groupId>
<artifactId>swagger-jaxrs</artifactId>
```

AEM Form Data Model configuration from Swagger

RESTful, SOAP-based, and OData services are configured in AEM Cloud Services under Form Data Model.

1. Log into AEM author instance using your admin credentials and navigate to: http://localhost:4502/etc/cloudservices.html



Form Data Model

Form Data Model relates, unifies, and represents business entities from multiple data sources.

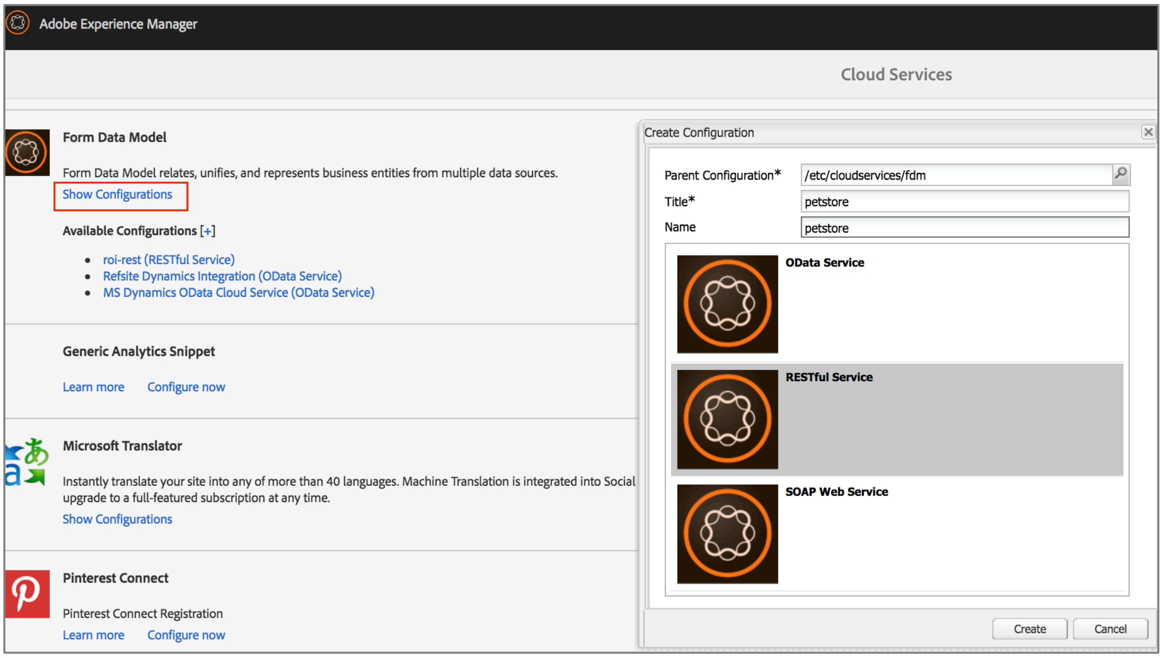
Show Configurations

Available Configurations [+]

- roi-rest (RESTful Service)
- Refsite Dynamics Integration (OData Service)
- MS Dynamics OData Cloud Service (OData Service)

2. Find ***Form Data Model*** under ***Third Party Services.***

3. Click **Configure Now** or **Show Configurations** based on if you have created a Form Data Model already. Configure the following screen:



Adobe Experience Manager

Cloud Services

Form Data Model

Form Data Model relates, unifies, and represents business entities from multiple data sources.

Show Configurations

Available Configurations [+]

- roi-rest (RESTful Service)
- Refsite Dynamics Integration (OData Service)
- MS Dynamics OData Cloud Service (OData Service)

Generic Analytics Snippet

Learn more   Configure now

Microsoft Translator

Instantly translate your site into any of more than 40 languages. Machine Translation is integrated into Social upgrade to a full-featured subscription at any time.

Show Configurations

Pinterest Connect

Pinterest Connect Registration

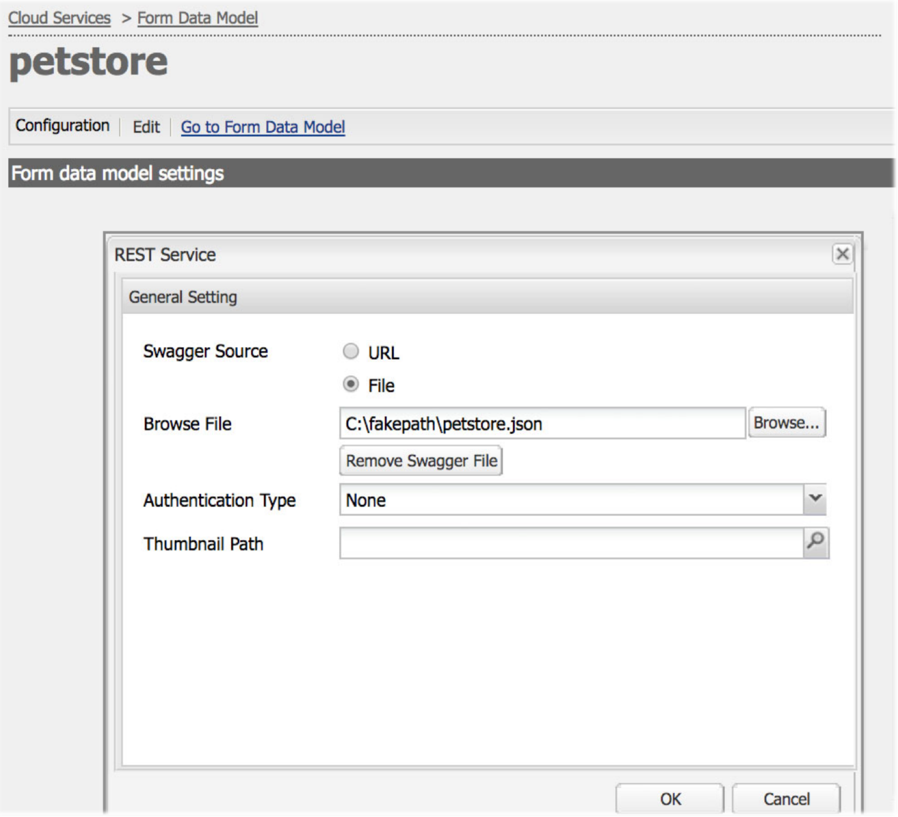
Learn more   Configure now

Create Configuration

Parent Configuration*: /etc/cloudservices/fdm

Title*: petstore

Name: petstore



Cloud Services > Form Data Model

petstore

Configuration | Edit | Go to Form Data Model

Form data model settings

REST Service

General Setting

Swagger Source: ○ URL ● File

Browse File: C:\fakepath\petstore.json

Authentication Type: None

Thumbnail Path:

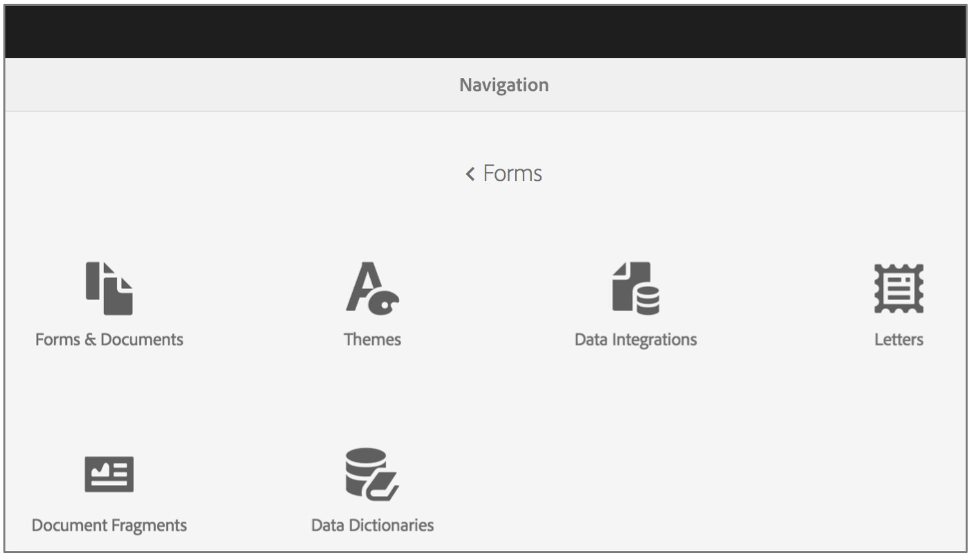
Here, the Swagger JSON file corresponding to a REST API can either be uploaded from the filesystem or a URL can be specified, when the file is hosted on a server that is accessible to the machine on which AEM instance is running.

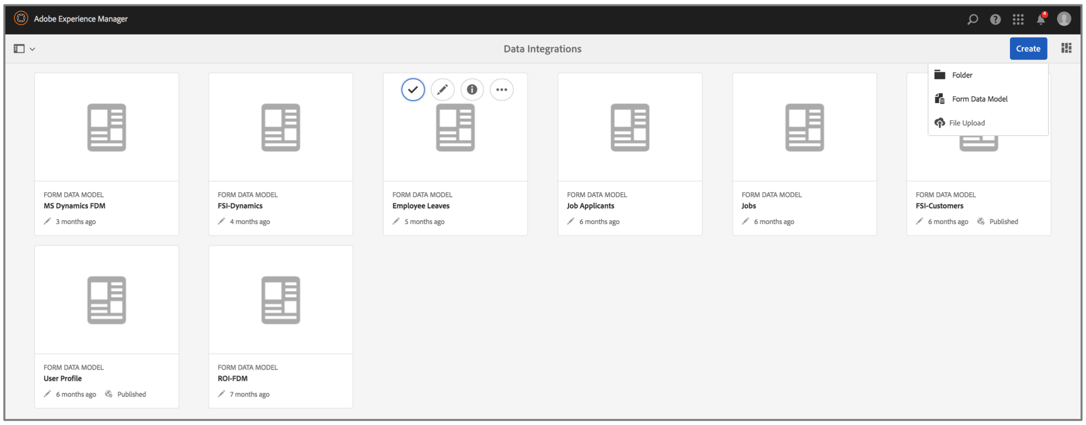
Select the Authentication Type from the drop-down: *None, OAuth2.0, Basic Authentication, API Key, or Custom Authentication*.

Note: AEM does not choose the security definitions and security constraints defined within the Swagger file. Configure the authentication requirement separately while configuring the REST service.

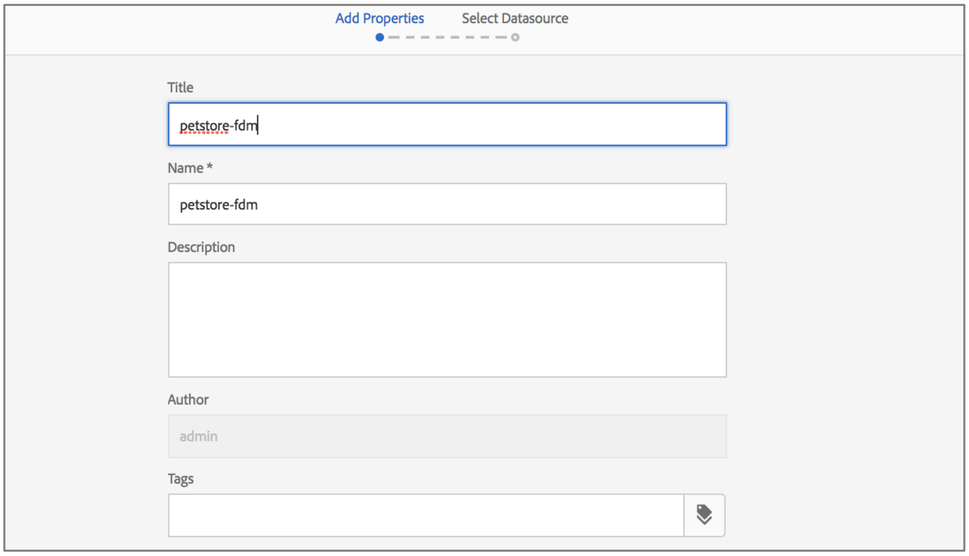
Once the REST API datasource has been configured, the schema from the Swagger becomes available to be used when creating a Form Data Model.
Navigate to:

> http://localhost:4502/aem/createfdm.html/content/dam/formsanddocuments-fdm





Create and enter basic information about the Form Data Model, and then click **Next**.



Add Properties Select Datasource

Title

petstore-fdm

Name *

petstore-fdm

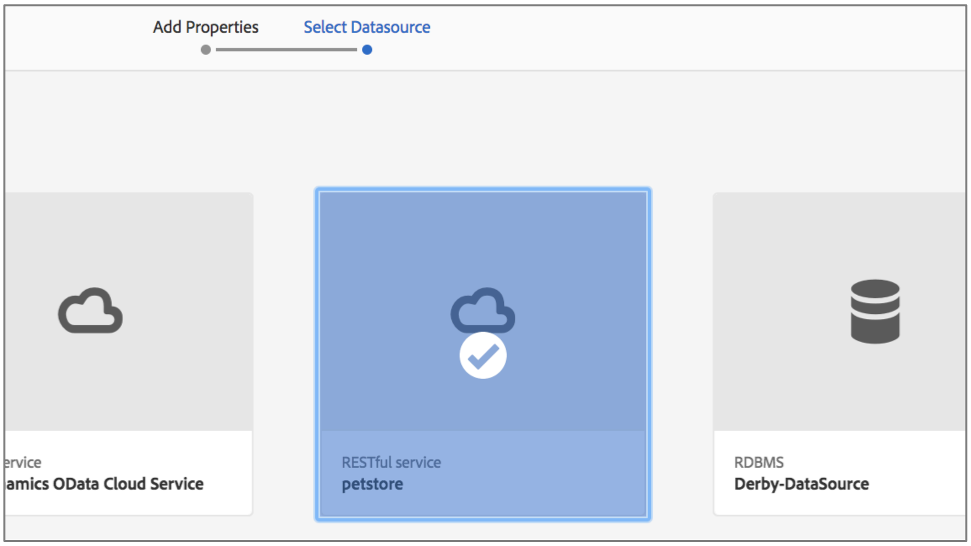
Description

Author

admin

Tags

On the next screen, choose the REST API datasource you created.



Add Properties Select Datasource

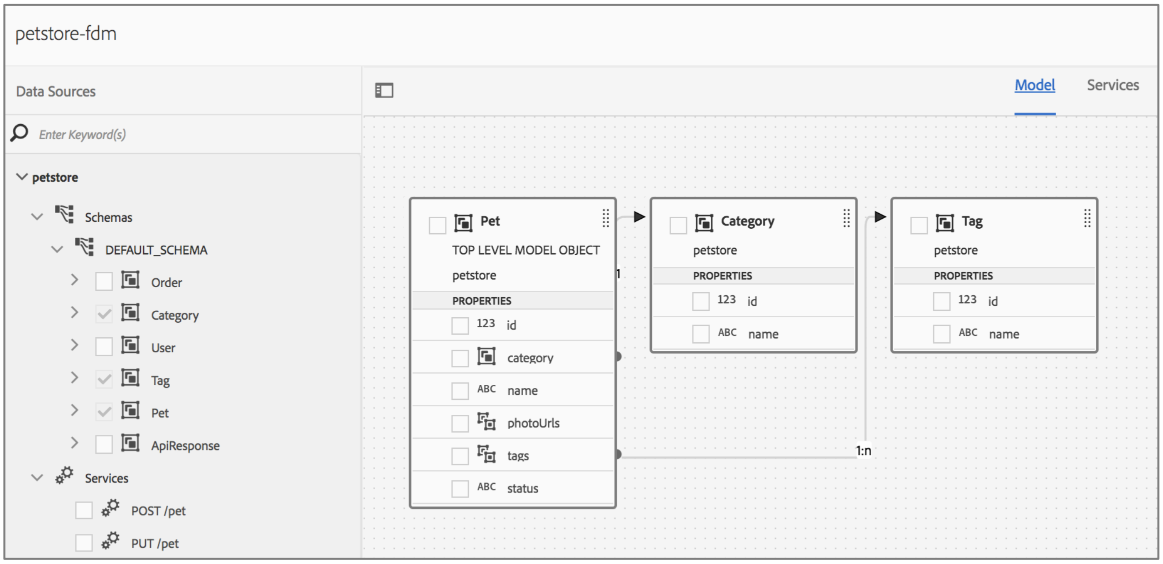
ervice
amics OData Cloud Service

RESTful service
petstore

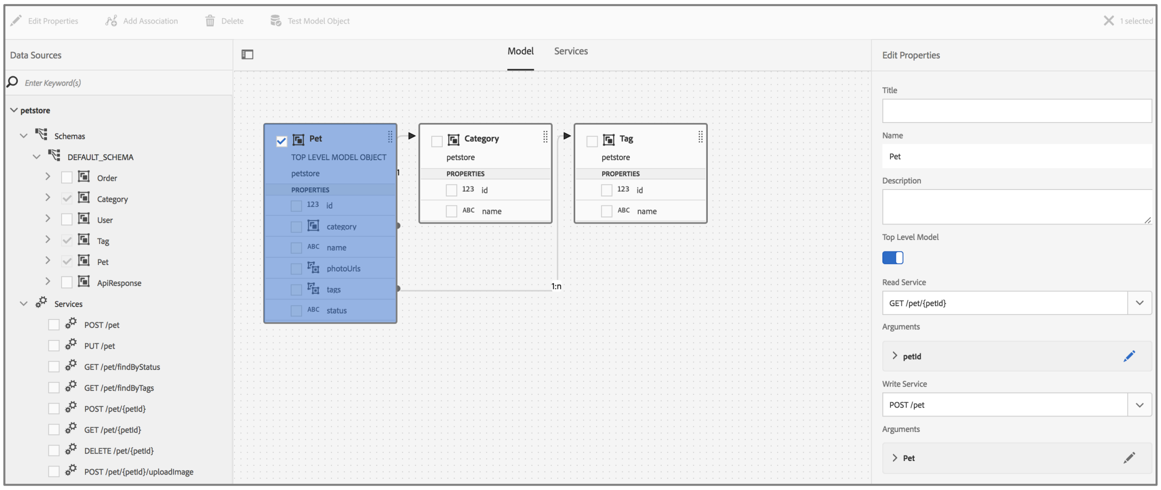
RDBMS
Derby-DataSource

Once the Form Data Model is created and you open it, you will see that the schema from the Swagger datasource is now available and you can configure the associated services from the Form Data Model Editor.

Select the Model objects that you want to use in your form.



Configure read and write service for the Model objects.



Title:

Name: Pet

Description:

Top Level Model: ●

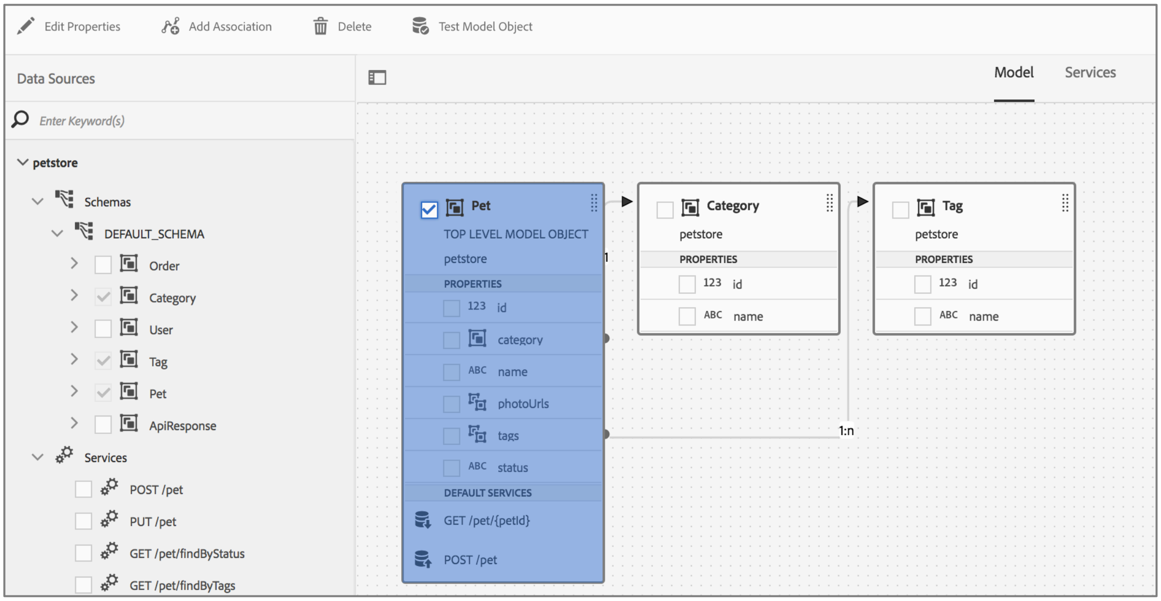
Read Service: GET /pet/{petId}

Arguments: petId

Write Service: POST /pet

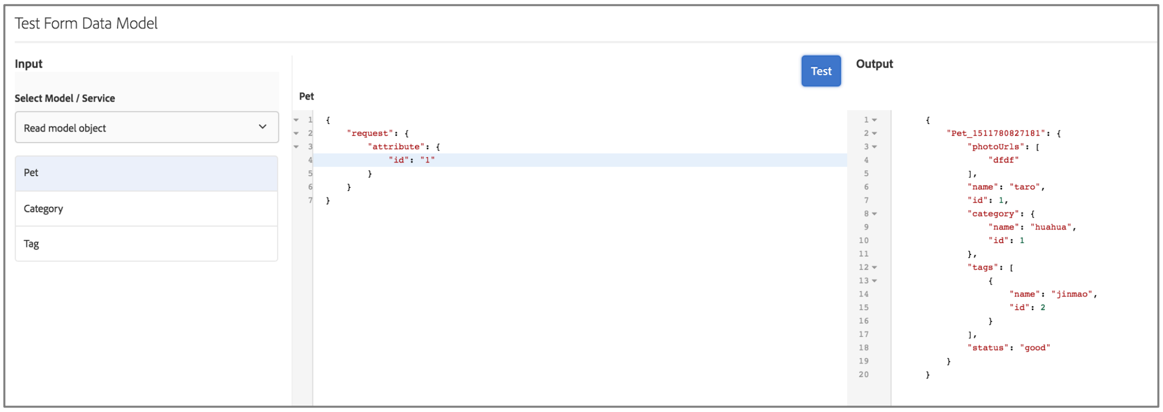
Arguments: Pet

Example 1



Example 2

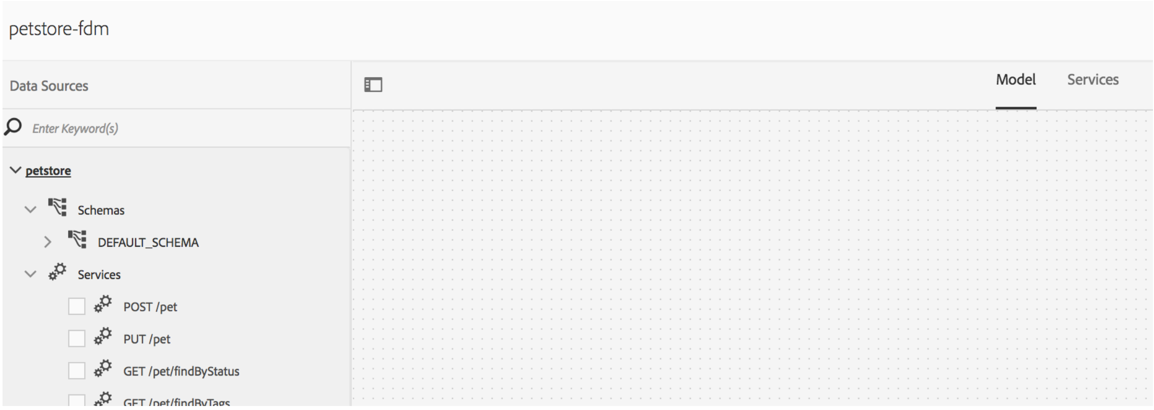
Test the Form Data Model.



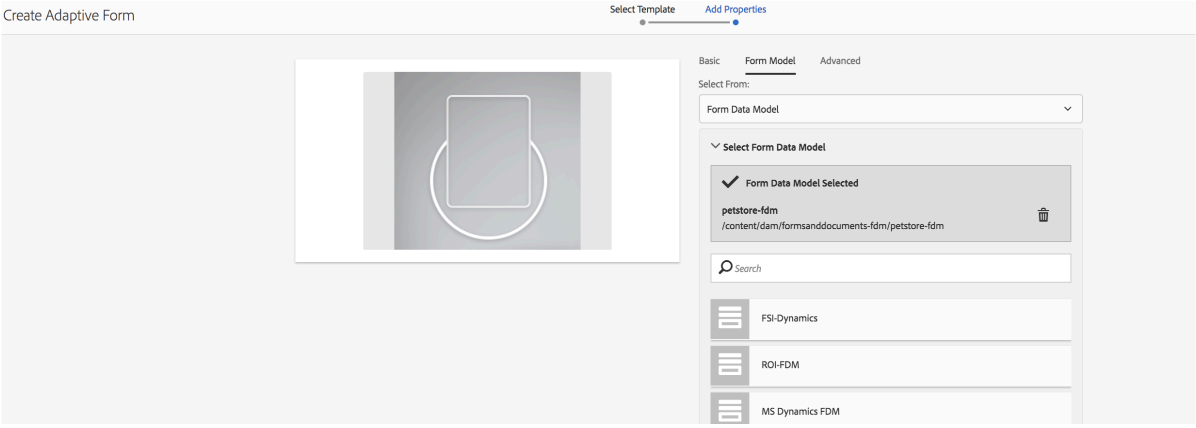
Pet

```
{
    "request": {
        "attribute": {
            "id": "1"
        }
    }
}
```

```
{
    "Pet_1511780827181": {
        "photoUrls": [
            "dfdf"
        ],
        "name": "taro",
        "id": 1,
        "category": {
            "name": "huahua",
            "id": 1
        },
        "tags": [
            {
                "name": "jinmao",
                "id": 2
            }
        ],
        "status": "good"
    }
}
```



And when authoring Adaptive Forms:



Select From: Form Data Model

Select Form Data Model

Form Data Model Selected

petstore-fdm
/content/dam/formsanddocuments-fdm/petstore-fdm