

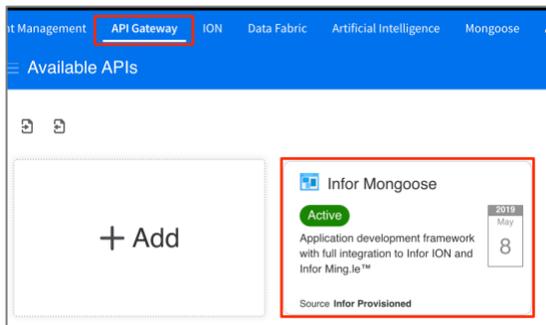


# Infor OS App Designer Guide to Mongoose Data Services

This guide will walk you through the process of creating a data service in Infor OS App Designer that connects to the Mongoose APIs.

## Prerequisites

- Verify that the Infor Mongoose suite has been provisioned in the tenant. Under API Gateway, verify that the Infor Mongoose suite is active:



- A strongly typed API has already been generated and is available to access within the API Gateway. For instructions on how to use the REST API Wizard within Mongoose, reference: [KB-2294576](#)
- Verify that the following two endpoints exists:

Available APIs / Infor Mongoose

Application development framework with full integration to Infor ION and Infor Ming.le™

Endpoints

Endpoint	Description	Indexing Status	Documentation
MONGOOSE/IDORestService/MGRestService.svc	Mongoose REST Service	📄	📄
MONGOOSE/IDORestService/ido	Mongoose REST Service V2	📄	📄
MONGOOSE/IDORestService/ido/dynamic	Mongoose REST V2 Swagger...	📄	📄

Page 1 of 1 5.36

- The name of the Mongoose configuration you want to access.
- Ensure that the user has the following security roles added in OS Portal before proceeding:
  - IAD-DataMaintainer
  - IAD-Designer
  - IAD-Publisher

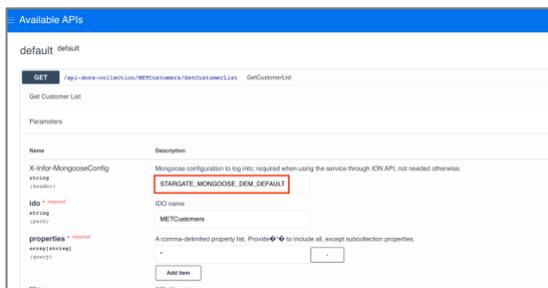
# Step 1: Identify and Test the API in the Infor API Gateway

Testing the data in the API Gateway is an optional step, however it is good to perform this step just to verify that the API is working as expected before creating a data service.

- From the OS Portal landing page, select **API Gateway**.
- Select the **Infor Mongoose** suite.
- Generally speaking, there are two types of APIs that can be used with OAD:
  - **Strongly Typed API (recommended)**: This is an API endpoint that has been dynamically created using the REST API Wizard within Mongoose. This API can have the collection, properties, and parameters all predefined eliminating the need for additional configuration when creating a data service in OAD.
  - **Standard Load Collection API**: This is a standard API that can be used to call a specific collection, properties, and filtering parameters. Using this method would require a more in-depth knowledge of the data itself and requires more configuration when configuring a data service in OAD.

## Testing a Strongly Typed API

- Click the documentation icon for the **MONGOOSE/IDORRequestService/ido/dynamic** endpoint.
- Click the documentation icon for the strongly typed API that you created using the REST API Wizard as stated in the prerequisites. In this example, we are going to reference the METCustomers business class that was dynamically created in my environment.
- Navigate to the **GET /api-docs-collection/METCustomers/GetCustomerList** API. In the X-Infor-MongooseConfig parameter, specify the configuration name. Leave the default values for ido and properties.







## Step 2: Create the Data Service in Infor OS App Designer

Now that we have successfully retrieved the data we need in the API Gateway, we can now create a data service that will call this same API.

### Creating a Strongly Typed Data Service

- Click on App Designer from the application ribbon within OS Portal.
- Click Data Services from the left pane and create a new data service.
- Select **ION API** for the type.
- Search for **GetCustomerList** and select **Get** for the method type. Select the API that we tested earlier:

Create Data Service - Find API

Search:  Suite:  Methods:

Name	Rest Path	Proxy Path		
GetCustomerList	/api-docs-collection/METC...	STARGATE_DEM/MONGOO...	<a href="#">Details</a>	<a href="#">Select</a>

- Give the data service a name and click **Next**.
- Next to the **Request Header** field, click the pencil icon to add in a request header parameter. Add in the **Key** of **X-Infor-MongooseConfig** and the **Value** should be the Mongoose configuration name you want to access:

Request Header Editor

+

Key	Value	Actions
accept	application/json	<input type="button" value=""/>
X-Infor-MongooseConfig	STARGATE_MONGOOSE_DEM_DEFAULT	<input type="button" value=""/>



Create Data Service

Name & Description    Input    Output    Summary

Basic Request Information

JSON    VIEW JSON

Root Element:     Object Id:

Output Parameters    Generate parameters    JSON Help    Add Parameter

Name	Alias
.CustNum	CustNum
.Name	Name

Test Data Service

Previous    Next    Finish

- Click the **Test Data Service** button, and then click the **Run Data Service** button at the bottom to verify the output parameters are correct:

Test Data Service

loadType:

bookmark:

pgc:

readOnly:

Run Data Service

Response Body

Name	Description
.CustNum	1065
.Name	Aura Cafe & Restaurant

OK

- Click **Ok** then click **Next** and **Finish**. Your data service is now ready to be used when building widgets within Infor OS App Designer.

## Creating a Standard Load Collection Data Service

- Click on **App Designer** from the application ribbon within OS Portal.
- Click on **Data Services** from the left pane and then click the Create a new data service button.
- Select **ION API** for the type.
- Search for **LoadCollectionAdvanced**, select **Infor Mongoose** as the suite, and **Get** for the method. Select the following API:

Create Data Service - Find API

Search:  Suite:  Methods:

Name	Rest Path	Proxy Path		
LoadCollectionAdvanced	/xml/{ido}/{props}/adv	STARGATE_DEM/MONGOO...	<a href="#">Details</a>	<a href="#">Select</a>
LoadCollectionAdvanced	/json/{ido}/{props}/adv	STARGATE_DEM/MONGOO...	<a href="#">Details</a>	<a href="#">Select</a>
LoadCollectionAdvanced	/xml/{ido}/adv	STARGATE_DEM/MONGOO...	<a href="#">Details</a>	<a href="#">Select</a>
LoadCollectionAdvanced	/json/{ido}/adv	STARGATE_DEM/MONGOO...	<a href="#">Details</a>	<a href="#">Select</a>

- Give the data service a name and click **Next**.
- Next to the **Request Header** field, click the pencil icon to add in a request header parameter. Add in the **Key** of **X-Infor-MongooseConfig** and the **Value** should be the Mongoose configuration name you want to access.

Request Header Editor

+

Key	Value	Actions
accept	application/json	
X-Infor-MongooseConfig	STARGATE_MONGOOSE_DEM_DEFAULT	

- In the **Input Parameters** section, add in the default IDO collection name, as well as any additional parameters you want to include such as a filter, rowcap, orderby, etc.

Create Data Service

GET `json/{ido}/{props}adv?filter={filter}&orderby={orderby}&rowcap={rowcap}`

Request Header

accept:application/json  
X-Infor-MongooseConfig:ADCQA\_Mongoose\_TST\_DEFAULT

VIEW JSON

Input Parameters JSON Help + Add Parameter

Required	Name	Type	Alias (Optional)	Default (Optional)	
<input checked="" type="checkbox"/>	ido	Path		METCustomers	
<input checked="" type="checkbox"/>	props	Path		CustNum_Name	
<input type="checkbox"/>	filter	Query		CustNum=1065	
<input type="checkbox"/>	orderby	Query		CustNum desc	
<input type="checkbox"/>	rowcap	Query		10	

Previous Next Finish

Make sure you remove any input parameters that you are not using by clicking the trash can icon on the right.

- Click the **Test Data Service** button, and then click the **Run Data Service** button at the bottom to verify a successful connection:

Test Data Service

CustNum desc

rowcap

10

**Run Data Service**

Response Body

```
{
  "Bookmark": "<B><P><P><CustNum</p></P><D><B>true</f></D><F><v>1065</v>
</F><L><v>1065</v></L></B>",
  "Items": [
    {
      "Name": "CustNum",
      "Value": "1065"
    },
    {
      "Name": "Name",
      "Value": "Aura Cafe & Restaurant"
    },
    {
      "Name": "_ItemId",
      "Value": "FBT:[METCustomers] CUST.DT=[2023-03-27 19:11:25.793] CUST.ID=
      ...
    }
  ]
}
```

OK

- Click **Next**.
- In this case, the **Root Element** will need to be changed to **\$.Items[\*]** because of this particular APIs JSON format. *It is important to note that this root element may change if you choose to use a different API.*
- Using the **Add Parameter** button, add in each output parameter you want in the format demonstrated below:

Create Data Service

Progress: Name & Description | **Input** | Output | Summary

⚠ Limited Documentation: Type is missing in schema, review your swagger file. X

**Basic Response Information**

JSON VIEW JSON

Root Element:  Object Id:

**Output Parameters** Generate parameters JSON Help **+ Add Parameter**

Name	Alias	
.0.Value	CustNum	<input type="checkbox"/>
.1.Value	Name	<input type="checkbox"/>

**Test Data Service**

Previous Next Finish

- Use the **Test Data Service** button at the bottom to ensure the output parameters piece is working as expected.

**Test Data Service**

props

CustNum\_Name  Required

filter

CustNum=1065

orderby

CustNum desc

rowcap

10

**Run Data Service**

Response Body

Name	Description
CustNum	1065
Name	Aura Cafe & Restaurant

OK

- Click **Next** and then click **Finish**. Your data service is now ready to be used when building widgets within Infor OS App Designer.