# Goal

To share System (SAPI) and Process (PAPI) API’s and Data Models across Medtronic teams. The system and process API’s need to be flexible to allow for reuse across Medtronic teams.

Contents

[Goal 1](#_Toc44429855)

[Versions 1](#_Toc44429856)

[Getting Started with an API 2](#_Toc44429857)

[Explore MuleSoft Exchange for an existing API. 2](#_Toc44429858)

[Explore MuleSoft Exchange for an existing Data Model. 2](#_Toc44429859)

[Navigate to Design Center and create your API and/or data model. 2](#_Toc44429860)

[Publish API to Exchange 6](#_Toc44429861)

[Begin the project in the Anypoint Studio client application 8](#_Toc44429862)

[Run the project 18](#_Toc44429863)

# Versions

|  |  |  |
| --- | --- | --- |
| Version 1.0 | Chad Jenness | 6/30/2020 |

# Getting Started with an API

## Explore MuleSoft Exchange for an existing API.

* If one exists, then contact the owner of the API and determine how it can be used by your project.
* If one does not exist, then continue to the next section

## Explore MuleSoft Exchange for an existing Data Model.

* If one exists, then contact the owner of the data model and determine how it can be used by your project.
* If one does not exist, then continue to the next section

## Navigate to Design Center and create your API and/or data model.

**Note: This is a subset of standards and patterns. For the full list, please see the Medtronic\_MuleSoft\_Development\_Best\_Practices\_v1.0 documentation.**

Standards:

* API Name = mdt-commercial-inventory-sapi
* Always lower case.
* 1st Segment – Always **mdt**.
* 2nd Segment – The business group or application that this API is creating the API.
* 3rd Segment – The business domain of the API.
* 4th Segment – Always one of the following to specify the API level:

**sapi** – System level API

**papi** – Process level API

**xapi** – Experience level API

* API Details:

**Note:** It is easier to use an existing working API that has been code reviewed as a model and copy the details from it over to your API. Make sure the Segment #4 for the API level is the same for when performing this.

**Note:** Please match the case usage as what is in the model API.

* Exchange Dependencies:

1. Latest version of the mdt-common-libraries
2. Latest version of any common data model

* Folders:

1. datatypes (All Lower Case)

Include all datatypes that are defined locally and not in the common data fragments.

1. examples (All Lower Case)

Include all JSON examples that are defined locally and not in the common data

* RAML:

1. title: Same name that was used for the API Name above.
2. version: Always start with v1.
3. description: Can you the same value as the title. Eventually we will want to give a more accurate description of the API.
4. baseUri: <https://localhost:8092/api/v1>
5. protocols: HTTPS
6. types:

* PingResponse: References the healthCheckResponse.raml from the imported mdt-common-libraries libraries.
* Any other local datatypes for your APIs

1. uses:

* CommonTraits: References the mdt-common-libraries from the imported mdt-common-libraries libraries.

1. Api Definitions:

* ping: See example below.

**Note:** The ‘is’ must include the following:

      # \*\*\*\* Common API Headers \*\*\*\*

      - CommonTraits.x-caller-name-required

      # \*\*\*\* Common API Error Codes/Responses \*\*\*\*

      - CommonTraits.standard-error-response

      # \*\*\*\* Client Credentials \*\*\*\*

      - CommonTraits.client-credentials-required

* API’s that you need to define: See example below.

**Note:** The ‘is’ must include the following for SAPI and PAPI level APIs:

      # \*\*\*\* Common API Headers \*\*\*\*

      - CommonTraits.x-caller-name-required

      - CommonTraits.x-correlation-id-required

      # \*\*\*\* Common API Error Codes/Responses \*\*\*\*

      - CommonTraits.standard-error-response

      # \*\*\*\* Client Credentials \*\*\*\*

      - CommonTraits.client-credentials-required

      # \*\*\*\* User Name \*\*\*\*

      - CommonTraits.x-user-name-required

**Note:** The ‘is’ must include the following for XAPI level APIs

        # \*\*\*\* Common API Headers \*\*\*\*

- commonTraits.x-correlation-id-required

        - CommonTraits.x-caller-name-required

        # \*\*\*\* Common API Error Codes/Responses \*\*\*\*

        - CommonTraits.standard-error-response

        # \*\*\*\* JWT Security Headers \*\*\*\*

        - CommonTraits.jwt-required

1. Example:

**Note:** This is an example of an API named as mdt-commercial-fieldinventory-sapi.

#%RAML 1.0

title: mdt-commercial-fieldinventory-sapi

version: v1

description: mdt-commercial-fieldinventory-sapi

baseUri: https://localhost:8082/api/v1

protocols:

  - HTTPS

types:

  PingResponse: !include exchange\_modules/0b475f2b-743d-4ce6-9af7-48997ef903f4/mdt-common-libraries/1.0.4/datatypes/healthCheckResponse.raml

  FieldInventoryResponse: !include exchange\_modules/0b475f2b-743d-4ce6-9af7-48997ef903f4/mdt-commercial-fieldinventory-datatypes/1.0.2/fieldInventory.raml

  FieldInventoryRequest: !include datatypes/fieldInventoryRequest.raml

uses:

  CommonTraits: exchange\_modules/0b475f2b-743d-4ce6-9af7-48997ef903f4/mdt-common-libraries/1.0.4/mdt-common-libraries.raml

/ping:

  get:

    is:

      # \*\*\*\* Common API Headers \*\*\*\*

      - CommonTraits.x-caller-name-required

      # \*\*\*\* Common API Error Codes/Responses \*\*\*\*

      - CommonTraits.standard-error-response

      # \*\*\*\* Client Credentials \*\*\*\*

      - CommonTraits.client-credentials-required

    responses:

      200:

        body:

          application/json:

            type: PingResponse

/fieldinventory:

  post:

    is:

      # \*\*\*\* Common API Headers \*\*\*\*

      - CommonTraits.x-caller-name-required

      - CommonTraits.x-correlation-id-required

      # \*\*\*\* Common API Error Codes/Responses \*\*\*\*

      - CommonTraits.standard-error-response

      # \*\*\*\* Client Credentials \*\*\*\*

      - CommonTraits.client-credentials-required

      # \*\*\*\* User Name \*\*\*\*

      - CommonTraits.x-user-name-required

    body:

      application/json:

        type: FieldInventoryRequest

        minProperties: 1

    responses:

      200:

        body:

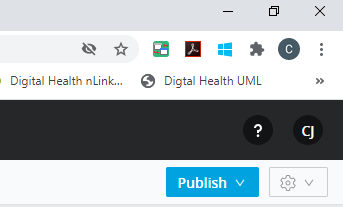
          application/json:

            type: array

            items: FieldInventoryResponse

## Publish API to Exchange

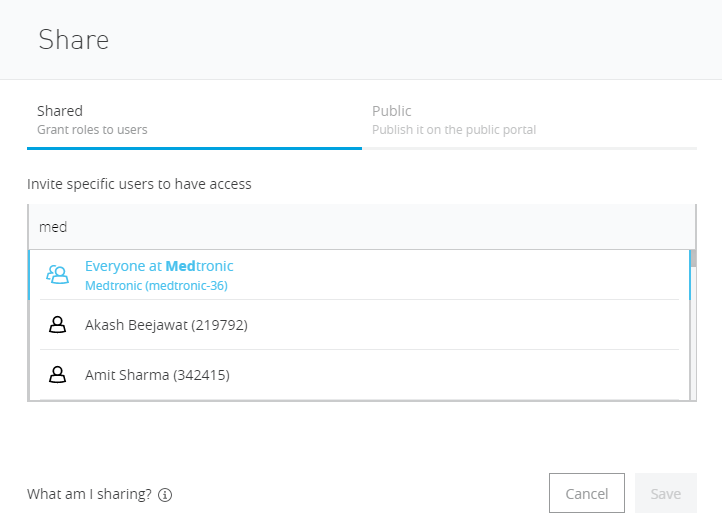
* Press the Publish button in the upper right hand corner.



* Select “Publish to Exchange”.
* When prompted for a version, type “1.0.0” for a version and press “Publish to Exchange”.

**Note:** Only continue with the next steps if this is not an application specific API.

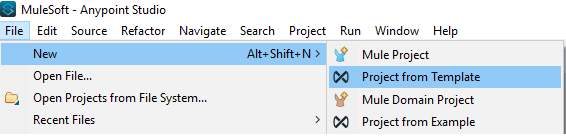
* Navigate to Exchange and select your API.
* In the upper right hand corner, press the “Share” button.
* Type “med” and select “Everyone at Medtronic”.



* Click the “Save” button.

# Begin the project in the Anypoint Studio client application

1. From the menu, select File -> New -> Project from Template



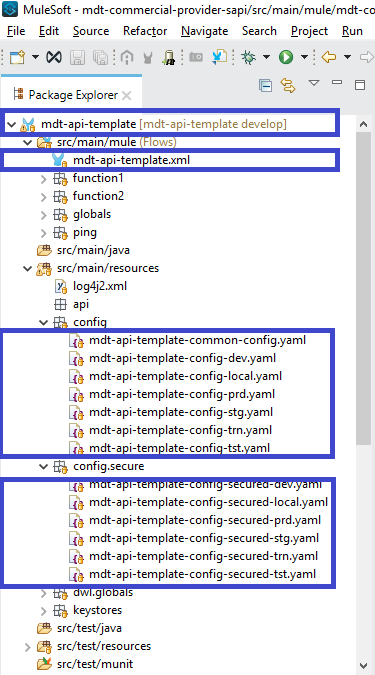
1. This will launch Exchange. If you are not logged into Exchange, please log in using your Azure Active Directory SSO credentials.
2. In Exchange, search for “mdt-api-template”.
3. Select “mdt-api-template”.
4. Press the “Open” button in the upper right hand corner.
5. This will download the “mdt-api-template” to your current local workspace.
6. Rename folders and files.

**Note:** For this step you will need to change “mdt-api-template” with the name of your API defined above.

**Note:** Picture below.

Folders/Files List:

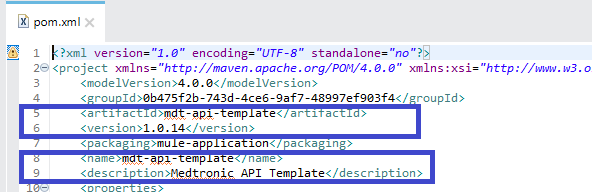
* Root level mdt-api-template
* src/main/mule/mdt-api-template.xml
* src/main/resources/config:
* mdt-api-template-common-config.yaml
* mdt-api-template-config-dev.yaml
* mdt-api-template-config-local.yaml
* mdt-api-template-config-prd.yaml
* mdt-api-template-config-stg.yaml
* mdt-api-template-config-trn.yaml
* mdt-api-template-config-tst.yaml
* src/main/resources/config.secure:
* mdt-api-template-config-secured-dev.yaml
* mdt-api-template-config-secured -local.yaml
* mdt-api-template-config-secured -prd.yaml
* mdt-api-template-config-secured -stg.yaml
* mdt-api-template-config-secured -trn.yaml
* mdt-api-template-config-secured -tst.yaml



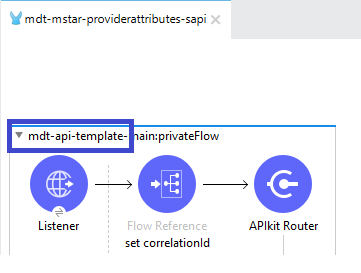
1. Rename file contents.

**Note:** For this step you will need to change “mdt-api-template” with the name of your API defined above.

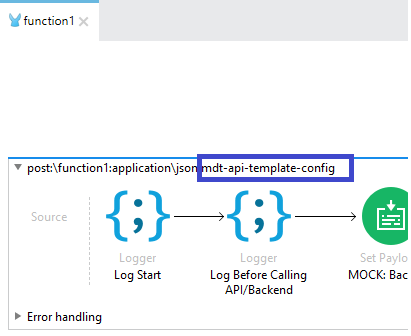
* Root/pom.xml
* artifactId
* version = Change this to 1.0.0
* name
* description = Change this to an accurate description of your API.



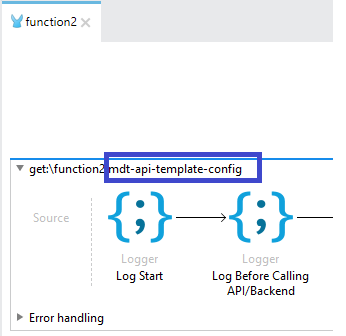
* src/main/mule/API\_NAME\_YOU\_DEFINED.xml. In this example the API\_NAME\_YOU\_DEFINED is “mdt-mstar-providerattributes-sapi”.



* src/main/mule/function1/function1.xml



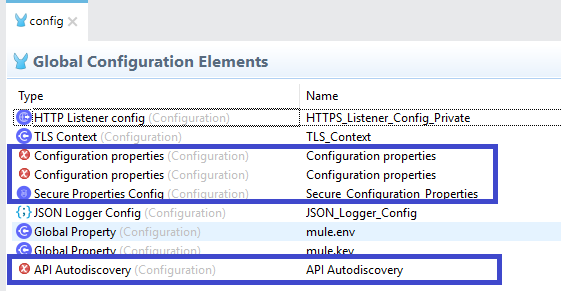
* src/main/mule/function2/function2.xml



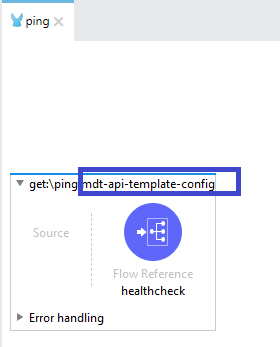
* src/main/mule/globals:

**Note:** For this step you will need to open each item and change “mdt-api-template” with the name of your API defined above.

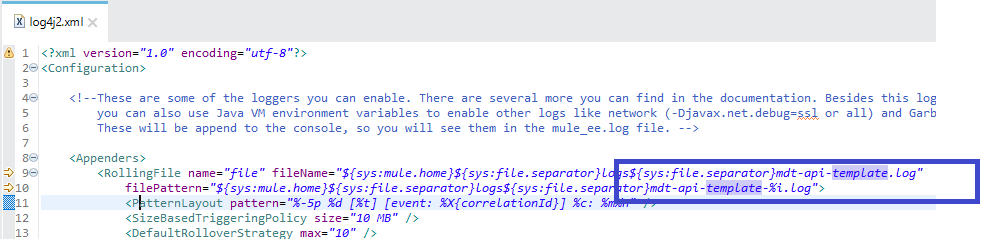
* Configuration properties
* Configuration properties
* Secure Properties Config
* API Autodiscovery



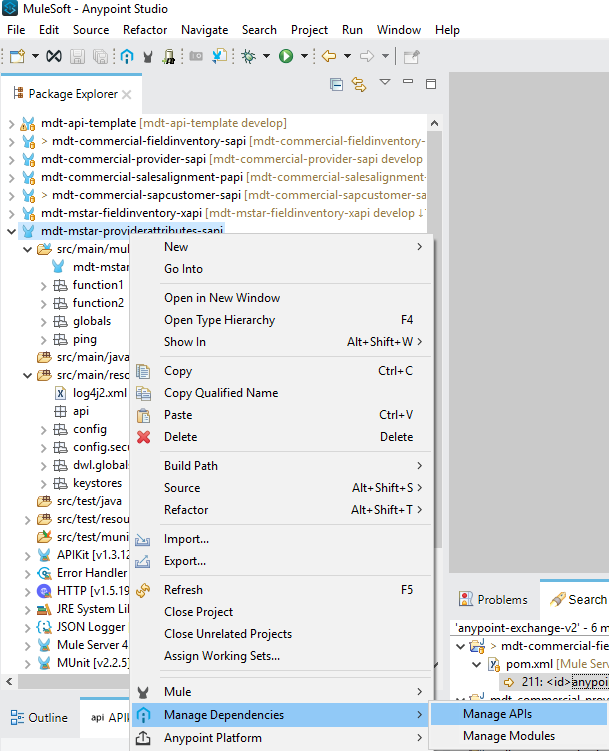
* src/main/mule/ping/ping.xml



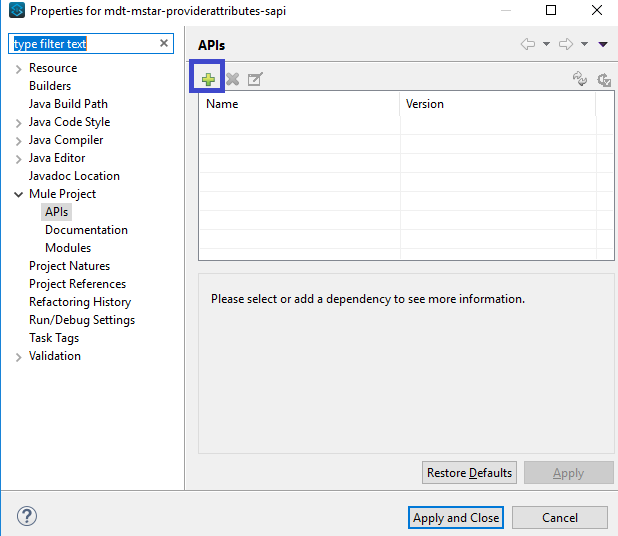
* src/main/resources/log4j2.xml



1. Select the root API workspace name and do a right mouse click.
2. Select Manage Dependencies -> Manage APIs

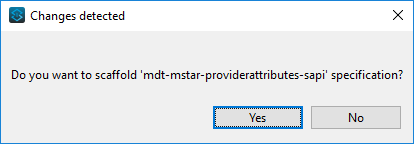


1. This will bring up a window to add APIs. In this window press the green plus sign and select “From Exchange”.



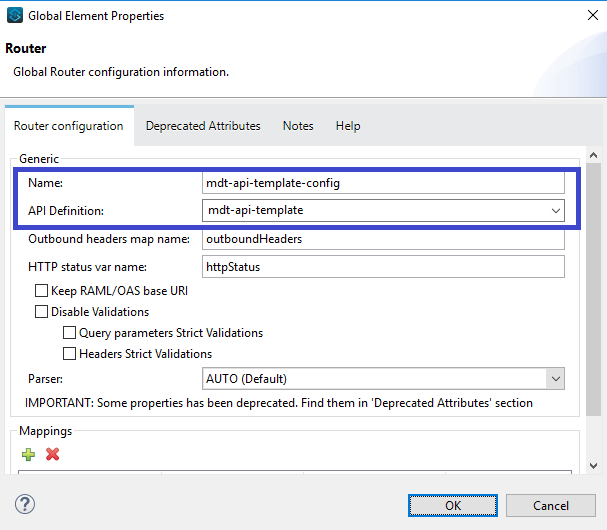
1. Search for the name of your API and select it in the Available Modules list.
2. Click the Add button.
3. Press the Finish button.
4. Press the Apply and Close button.
5. This will bring up the window to option to scaffold below.

**Note: It is important to always select No.**



1. Open src/main/mule/API\_NAME\_YOU\_DEFINED.xml. In this example the API\_NAME\_YOU\_DEFINED is “mdt-mstar-providerattributes-sapi” and select “Global Elements”.

* Select Router (Configuration) and press the Edit button.
* Name: Change “mdt-api-template” to the name of your API.
* API Definition: In the drop drown, select the name of your API.



* Select “Configuration XML” and change the below doc name:



# Run the project

1. In the canvas perform a right mouse click and select “Run project …”.
2. Verify that the application starts and deploys successfully.
3. Run a ping test.

* GET - <https://localhost:8092/api/v1/ping>
* Headers:

client\_secret:1234567890

client\_id:chad

x-caller-name:TestCaller