Add a Capability – Address Validator Tutorial

This document is a step-by-step tutorial about configuring an address validator capability.

Here are the differences between Mozu Extensions and Capabilities:

* Extensions are used to create new functionality not available in Mozu, capabilities replace an existing Mozu functionality.
* Extensions are called asynchronously (through events) whereas capabilities are invoked synchronously by the Mozu platform.
* A configured endpoint for a capability takes over the flow of execution in operations such as address validation, tax calculation, or order validation (i.e. fraud protection).

In this example, we will write a simple address validator capability which will successfully validate addresses as long as the address is in a set of zip codes.

Creating the Capability Definition

Unlike event endpoints on an extension where the definition needs to be a POST which accepts the MozuEvent in the request body, a capability definition will change (i.e. different Verbs and Request body types) depending on the capability type being supported. Since we will be implementing an address validator capability, navigate to the REST API documentation, http://developer.mozu.com/resources/, and click on the **Address Validation** link at the bottom of the page on the left, underneath **Capability Contracts**.

This documentation will describe the endpoint signature. In the case of the address validator, it’s a POST to the URL: https://<yourpassagewayurl>/MozuIntegrationTemplate/api/addressvalidationcapability/ with the Address type accepted for the body and a list of Addresses being returned as a result.

To see an example of the address validation capability, see the java code at: C:\Users\bob\_hewett\git\mozu-address-validation\java\src\main\java\com\mozu\addressvalidation\controllers\AddressValidationController.java

Creating the Capability Configuration Endpoint

Because most capabilities require information from the merchant for connecting to a third-party service, we’ll need to also host an endpoint for collecting that information in a web form. The Java example is a fully functioning servlet called **AddressValidator.** If you download the git repository, the java directory contains a fully functional web servlet. To run the servlet as a jetty server do the following.

1. On a windows machine you can run the jettyDebug.bat script.
2. On other machines there is a gradlew script available. Run the following to run the service:  
   gradlew clean build jettyRun

Hooking into the Address Validator Service

Now we’ll create a new application registration for this capability that in the Mozu Dev Center.

1. Navigate to Dev Center and click **Console > Develop > Applications**. Create a new application with the following data:

a) **Name**: Address Validator App – Your Name

b) **Application ID**: addrvalidator<yourname>

2. Right-click on the newly created application and select **Edit** from the context menu.

3. In the **Behaviors** section, add the following behaviors.

* SettingsPlatformAppDev
  + Application Status Read
  + Application Status Update

4. Click the **Save** button.

5. In the **Capabilities** tab, click **Add Capability**, choose **Address Validator** from the drop-down and click **OK**.

6. In the **ValidateAddress endpoint URL** field, place **your (passageway) URL** appended with **MozuIntegrationTemplate/api2/addressvalidationcapability.**

7. Select **United States** and **Canada** from the list of countries.

8. Click **Save**.

9. Now click **Install** and install it to your tenant.

10. Edit the **mozu\_config.properties** file in the directory java/src/main/resource

11. Update our **Application Key** and **SharedSecret** values with the new application’s values (from the **Application Core** tab in Dev Center).

12. Restart the application jetty server (and setup debugging for the project if you wish).

Configuring the Capability

1. Go to your tenant Sitebuilder sandbox and navigate to the **Applications** area (**Settings > Application**). The **Address Validator** should now be visible in the applications list.

2. Double-click the Address Validator App and select the two shipping countries.

3. Click **Enable**.

4. In order for the capability to be called at runtime, we also need to configure the site to validate addresses before they can be submitted. Navigate to the General Settings page in sitebuilder under **Settings > General Settings**.

5. Create a Channel by navigating to the **Channels** view in the top toolbar.

6. Click **Create New Channels**, enter the following data, and click **Save**.

a. **Code**: 123

b. **Name**: My Channel

c. **Country**: United States

7. Return to the **General Settings** view.

8. Choose the newly created channel from the **Channel** dropdown.

9. Scroll down to the **Address Validation** section and select the **Address Validation Enabled** checkbox.

10. Click **Save**.

Using the Capability in Mozu

1. Now that we are all set up, navigate to the **Catalog** view in Mozu Admin.

2. Right-click a product and select **View Live > <your site>** from the context menu.

3. Mozu opens the storefront with the selected product in view.

4. Click **Add to Cart** and then click **Proceed to Checkout** in the next window.

5. Type in **Shipping Information** and be sure to use a zip code not hardcoded in your solution.

6. Click the **Next** button and you will be presented with the three addresses your address validator recommends.