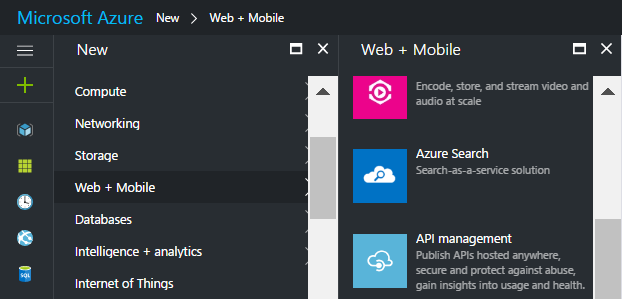
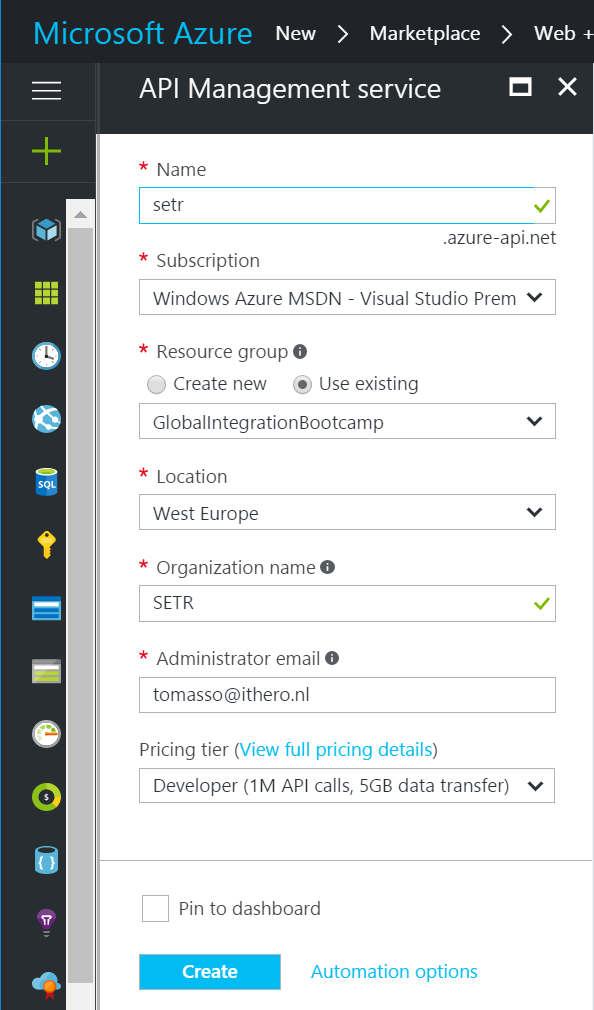
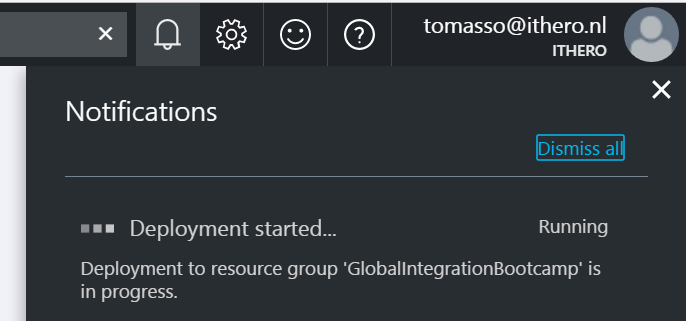
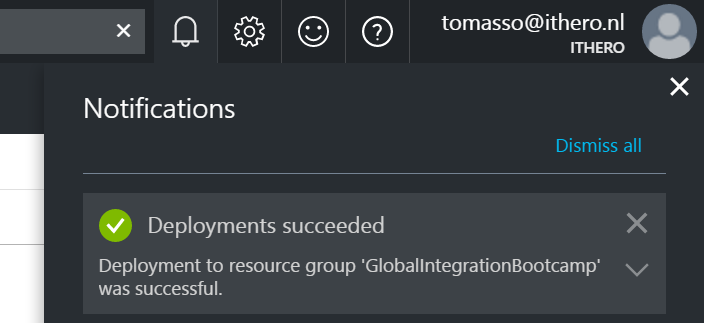
# Create an API Management instance

The first step in working with API Management is to create an API Management instance.

***Note***

It takes about 10 minutes to create the API Management instance in Azure.

1. Sign in to the [Azure Portal](https://portal.azure.com/) and click **New**, **Web + Mobile**, **API Management**.  
   
2. For Name, specify a unique sub-domain name to use for the service URL.  
   Choose the desired Subscription, Resource group and Location for your service instance.  
   Enter SETR for the Organization Name, and enter your email address in the Administrator E-Mail field.  
     
   ***Note***  
   The email address is used for notifications from the API Management system.  
     
     
   ***Note***  
   API Management service instances are available in three tiers: Developer, Standard, and Premium. You can complete this lab by using the Developer tier.
3. Click **Create** to start provisioning your service instance.
4. The Deployment takes several minutes. Click in the Header toolbar on Notifications to follow the Deployment  
   
5. Once the service instance is created, click on the notification to open API Management.  
   

# How protect a Web API in API Management with Azure Active Directory

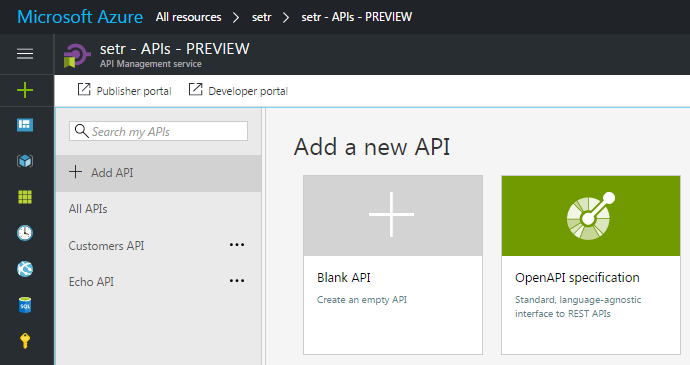
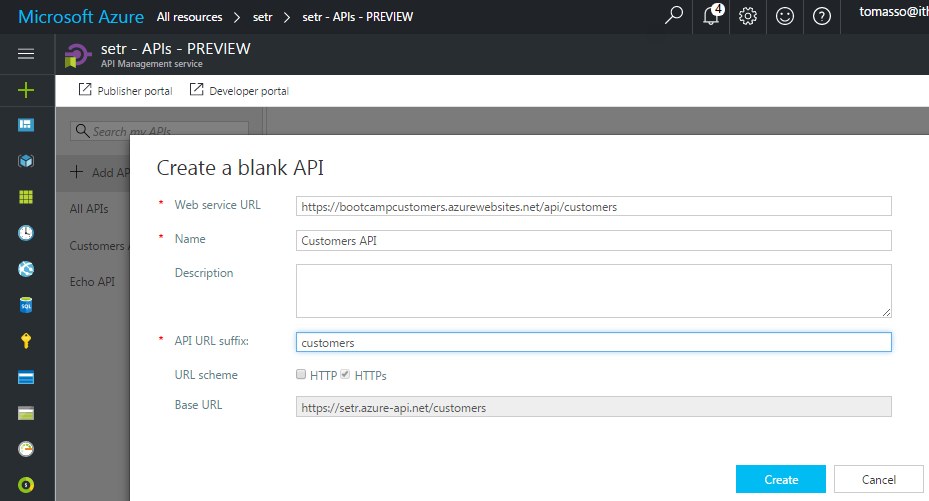
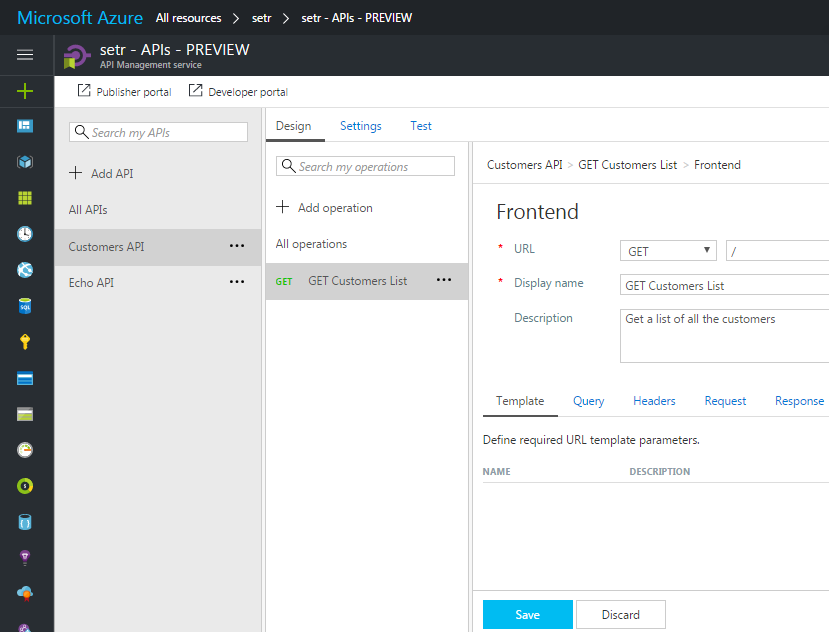
In this lab you will configure OAuth 2.0 user authorization for the Web API in API Management . At the end of this lab you will a Web App in API Management that is authenticated with Azure Active Directory.

The following activities are going to be done:

* Import the Web API into API Management
* Add the API to a Product
* Call the API unsuccessfully from the developer portal
* Register the developer portal as an AAD application
* Configure an API Management OAuth 2.0 authorization server
* Enable OAuth 2.0 user authorization for the Web API
* Successfully call the Web API from the Developer Portal

## Import the Web API into API Management

APIs are configured from the API publisher portal, which is accessed through the Azure Portal. To reach it, click Publisher portal from the toolbar of your API Management service.

1. Operations can be added to APIs manually, or they can be imported. To add a Web API manually, click APIs-PREVIEW from the API Management menu on the left, and then click Blank API.  
   
2. Add the Web service URL, the Name, the API URL suffix and click on Create.  
   
3. Click on the Add operation button add the GET operation.
4. Add a GET operation to get a list of all the customers and click on Save.  
   

## Add the API to a Product

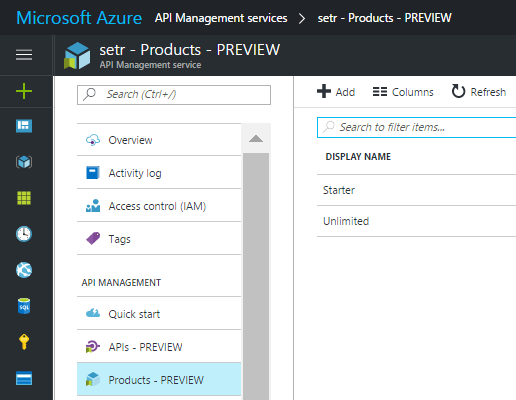
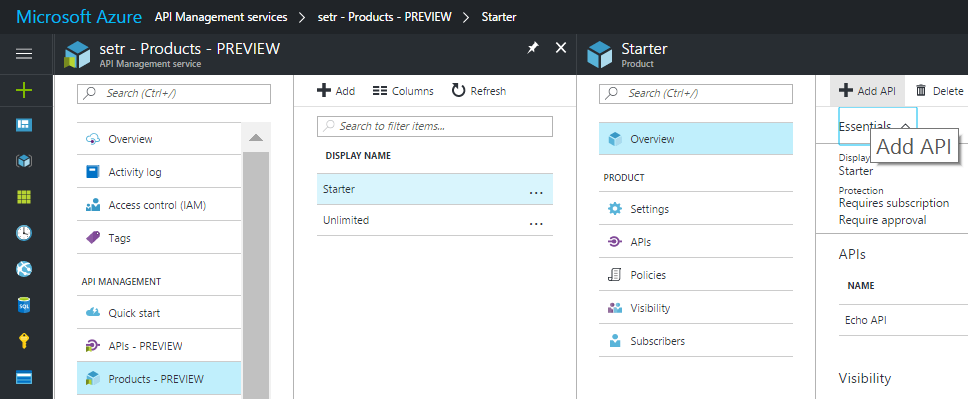
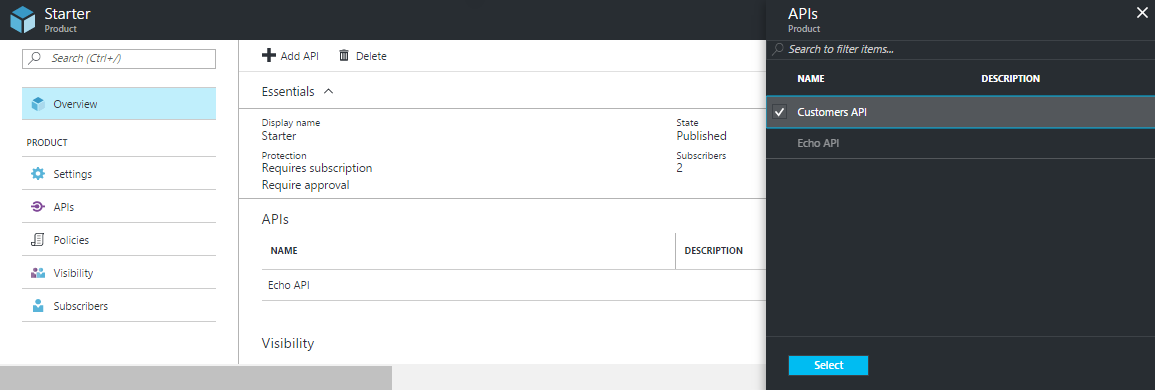
By default, each API Management instance comes with two sample products:

• Starter

• Unlimited

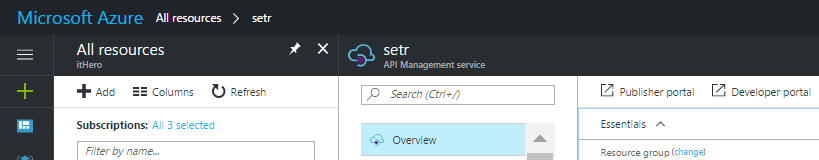
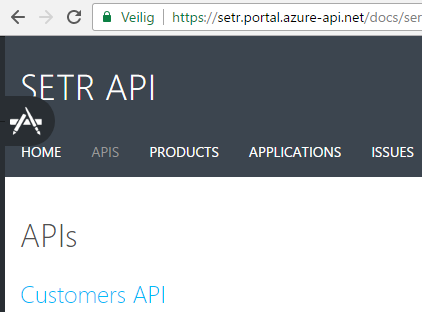
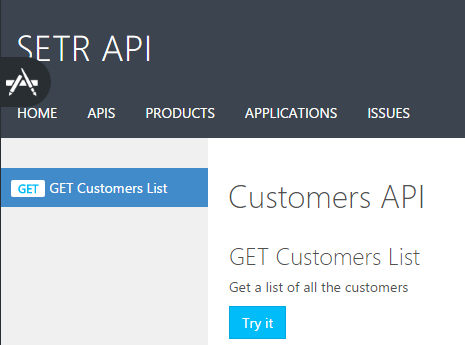
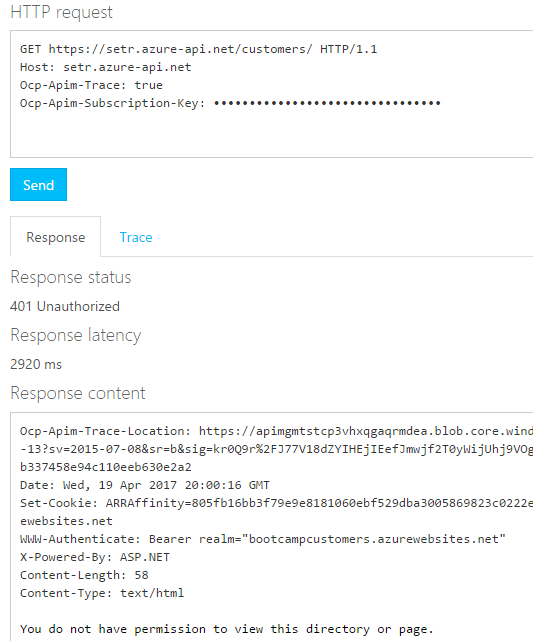
In order to make calls to an API, developers must first subscribe to a product that gives them access to it. Developers can subscribe to products in the developer portal, or administrators can subscribe developers to products in the publisher portal. You are an administrator since you created the API Management instance in the previous steps in the tutorial, so you are already subscribed to every product by default.

In this step, we are going to add the created Web API to the Starter product.

1. Click in the menu on Products-PREVIEW and then on the Starter product name.
2. Click on the ADD API button.
3. Select the created Web API and click Select.

## Call the API unsuccessfully from the developer portal

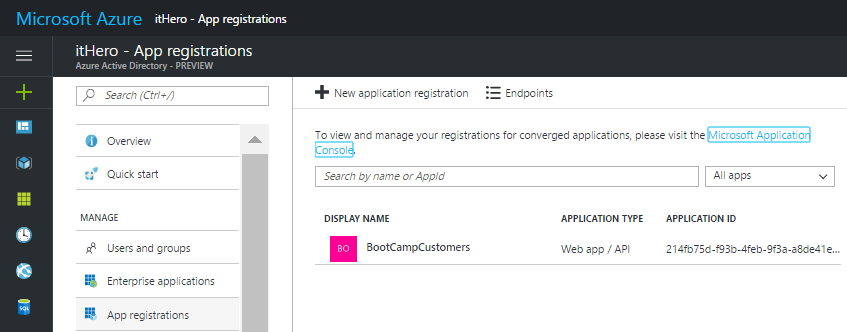
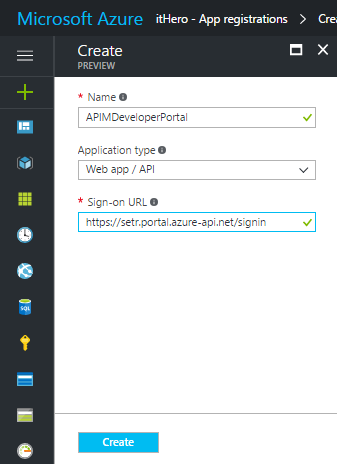
At this point, the API has been imported into API Management, but cannot yet be called successfully from the developer portal because the backend Web API is protected with Azure AD authentication.

1. Click on the Developer portal link from the Azure portal or from the top-right side of the Publisher portal.  
   
2. Click APIs and click the created Web API.  
   
3. Click Try it.  
   
4. Click Send and note the response status of 401 Unauthorized.  
   

The request is unauthorized because the backend API is protected by Azure Active Directory. Before successfully calling the API the Developer portal must be configured to authorize developers using OAuth 2.0 and pass on the AAD token to the Web API. This process is described in the following sections.

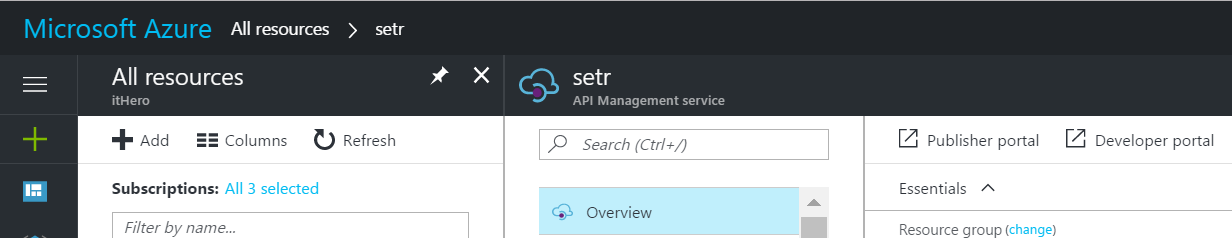
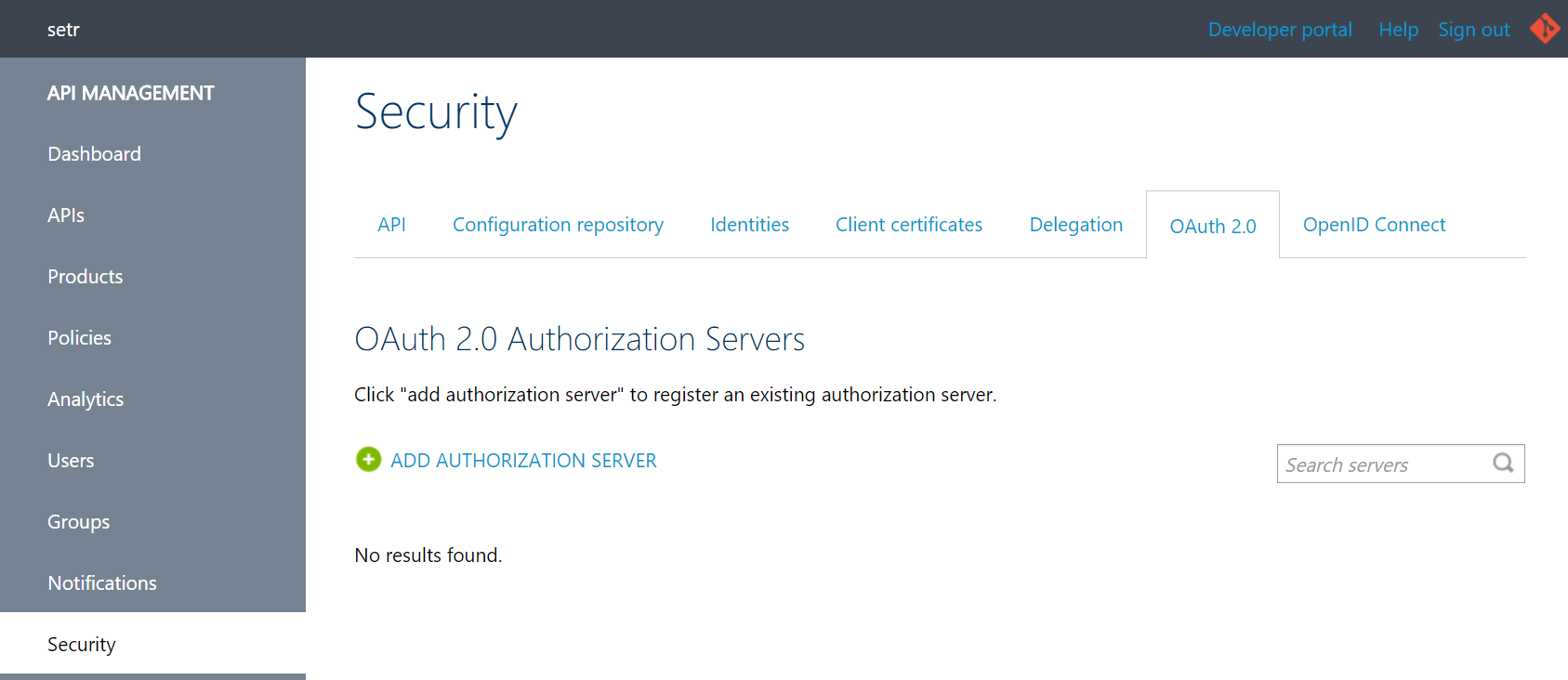
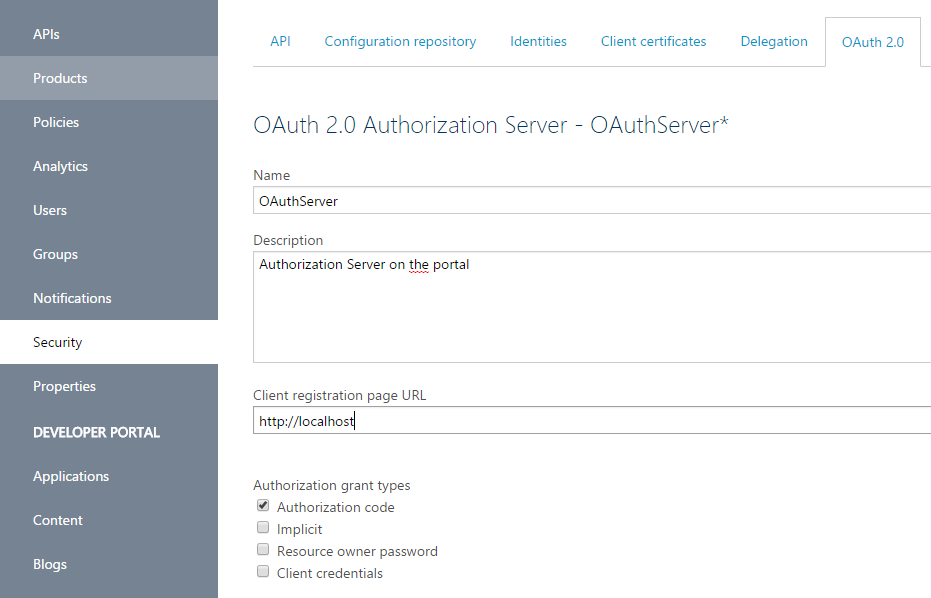
## Register the Developer Portal as an AAD application

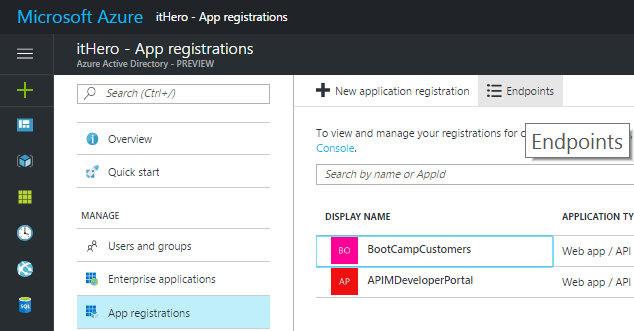
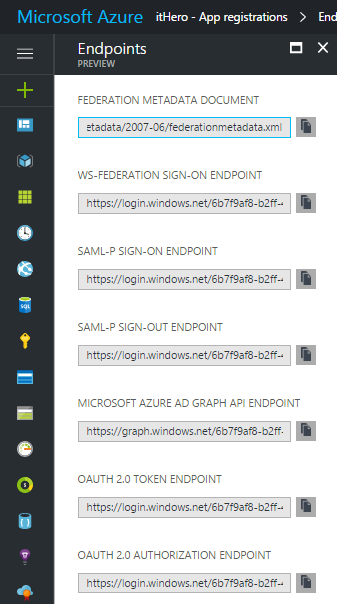
The first step in configuring the Developer portal to authorize developers using OAuth 2.0 is to register the Developer portal as an AAD application.

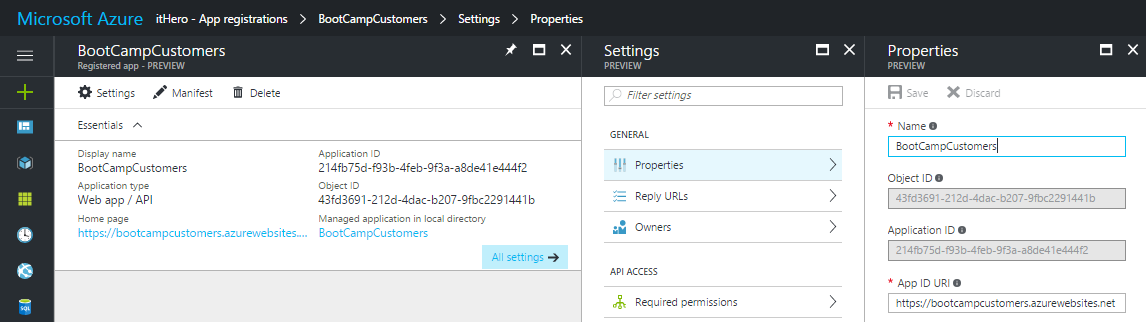
1. Navigate to the created Azure AD tenant from the AAD labs, in this example **itHero** and select App registrations in the menu.  
   
2. Click the New application registration button to create a new Azure Active Directory application.
3. enter a name. In this example APIMDeveloperPortal is used. Choose Web app /API and for Sign-on URL enter the URL of your API Management service and append /signin. In this example https://setr.portal.azure-api.net/signin is used.  
   
4. When the desired App properties are configured, click the Create button to create the application.

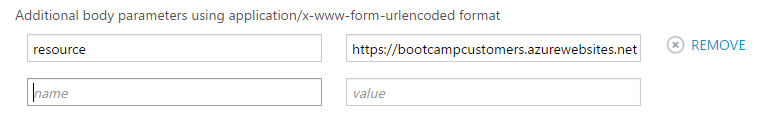
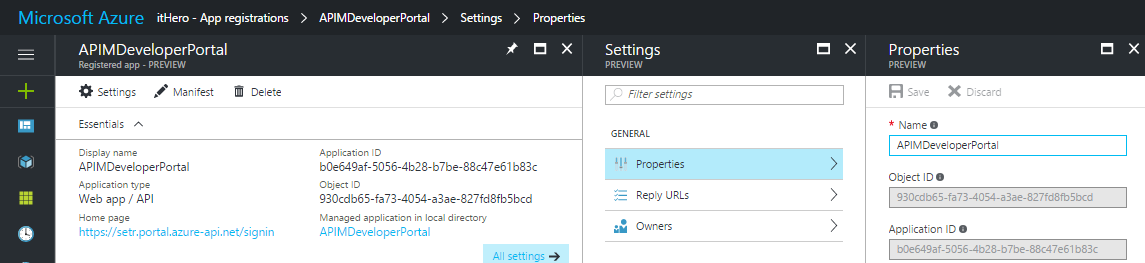
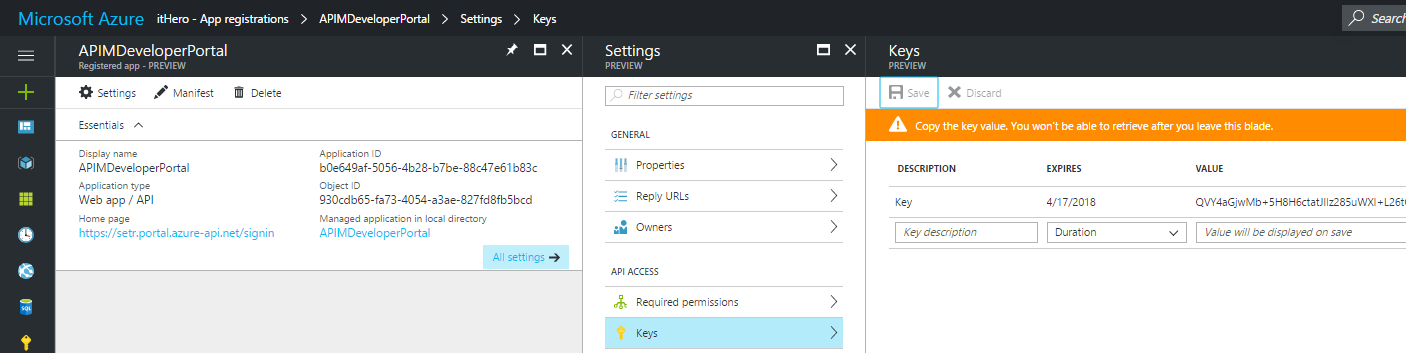
## Configure an API Management OAuth 2.0 authorization server

The next step is to configure an OAuth 2.0 authorization server in API Management. This functionality is not yet available in the Azure Portal and is configured from the API Publisher Portal.

1. To reach the Publisher portal, click on the Publisher portal link from the toolbar of your API Management service.  
   
2. Click Security from the API Management menu on the left, click OAuth 2.0, and then click Add Authorization Server.  
   
3. Enter a name and an optional description in the Name and Description fields. These fields are used to identify the OAuth 2.0 authorization server within the API Management service instance. In this example OAuthServer is used. Later when you specify an OAuth 2.0 server to be used for authentication for an API, you will select this name.
4. For the Client registration page URL enter a placeholder value such as http://localhost. The Client registration page URL points to the page that users can use to create and configure their own accounts for OAuth 2.0 providers that support user management of accounts. In this example users do not create and configure their own accounts so a placeholder is used.  
   
5. To specify the Authorization endpoint URL and Token endpoint URL you need the Authorization endpoint and the Token endpoint from the Azure Active Directory you created in the AAD Lab.

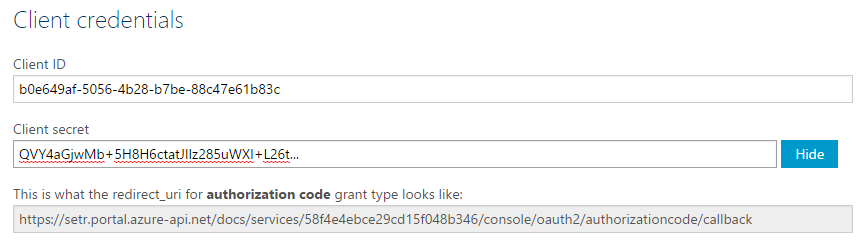
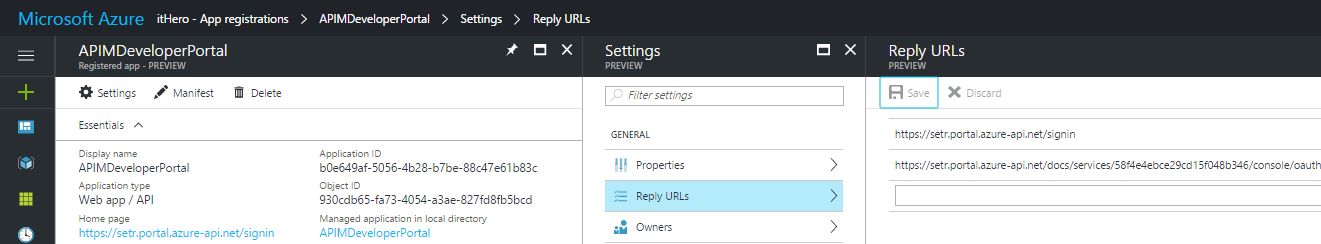
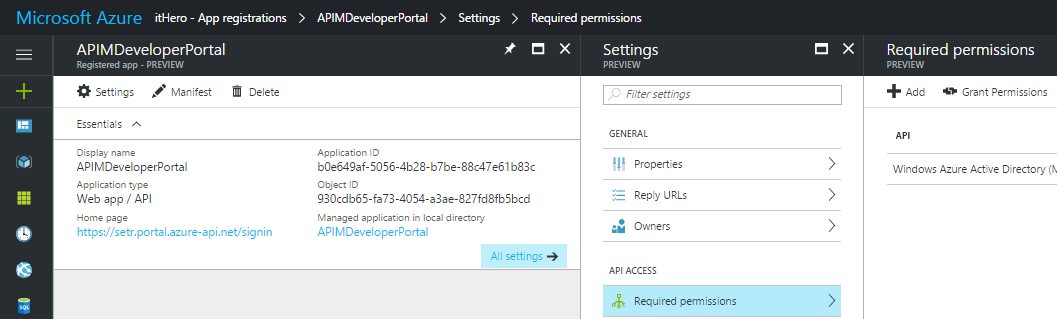
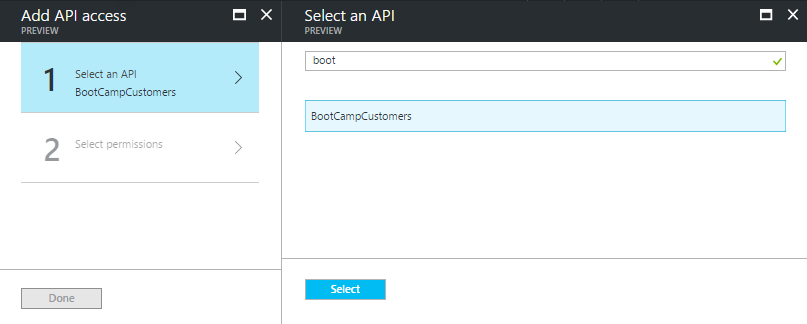
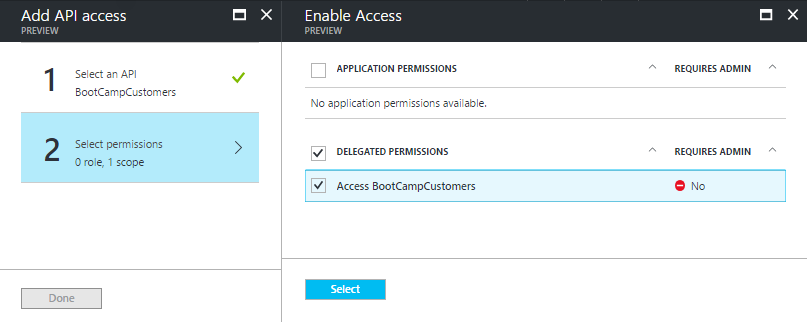
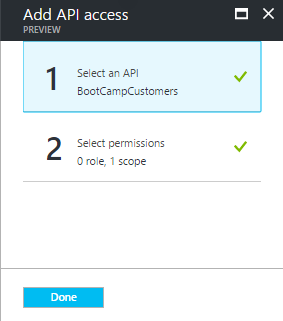
To access the endpoints navigate to App registrations and click on Endpoints.  
  


1. Copy the OAuth 2.0 authorization endpoint and paste it into the Authorization endpoint URL textbox.
2. Copy the OAuth 2.0 token endpoint and paste it into the Token endpoint URL textbox.  
   
3. Copy the App ID URI from the AAD application for the Web API that was created in the previous lab and you used in the previous exercise (exposing a web api).  
   

1. Add an additional body parameter named resource and for the value paste the App ID URI from the AAD application for the backend service.  
   
2. Next, specify the client credentials. These are the credentials for the resource you want to access, in this case the Developer portal.  
   To get the Application ID, navigate to the Configure tab of the AAD application for the Developer portal and copy it. (In the Old Azure portal the Client Id)  
   
3. To get the Client Secret click the Select duration drop-down in the Keys section and specify an interval. In this example 1 year is used.
4. Click Save to save the configuration and display the key.  
   

**Important**

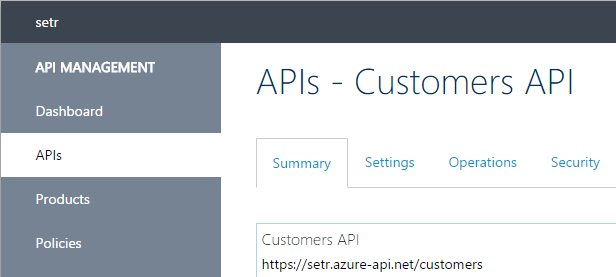
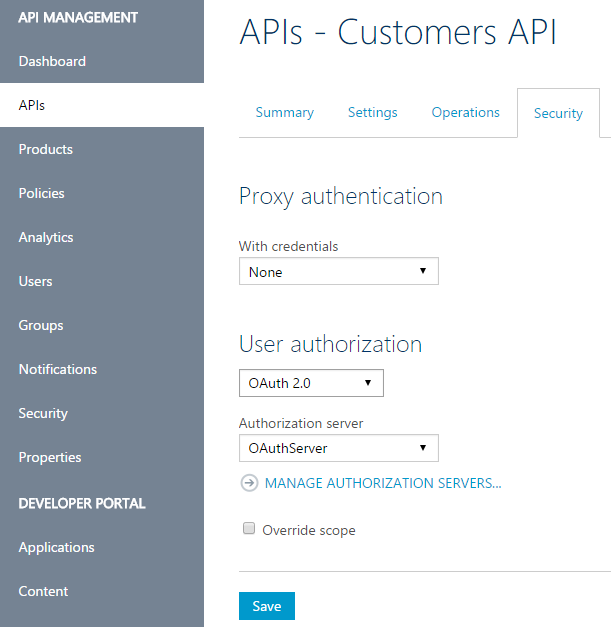
Make a note of this key. Once you close the Azure Active Directory configuration window, the key cannot be displayed again.

1. Copy the key to the clipboard, switch back to the Publisher portal, paste the key into the Client Secret textbox, and click Save.  
   
2. Immediately following the client credentials is an authorization code grant. Copy this authorization code and switch back to your Azure AD Developer portal application configure page, and paste the authorization grant into the Reply URL field, and click Save again.  
   
3. The next step is to configure the permissions for the Developer portal AAD application. Click Required Permissions and then click Add application.  
   
4. Click the search icon, type the first part of the Web API application into the Starting with box, select the Application, and click Select.  
   
5. Click Delegated Permissions and check the box for Access [Application], and click Select. This allows the Developer portal application to access the backend service.  
   
6. Click on Done to finish.  
   

## Enable OAuth 2.0 user authorization for the Web API

Now that the OAuth 2.0 server is configured, you can specify it in the security settings for your Web API.

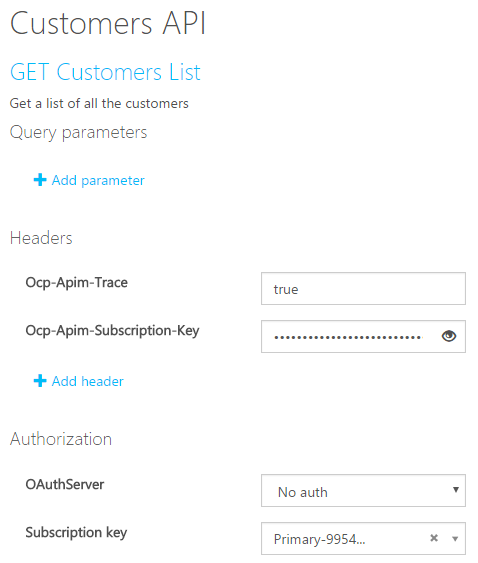
***Note***   
Security settings are net yet available in the new Azure Portal and have to be set in the Publisher Portal.

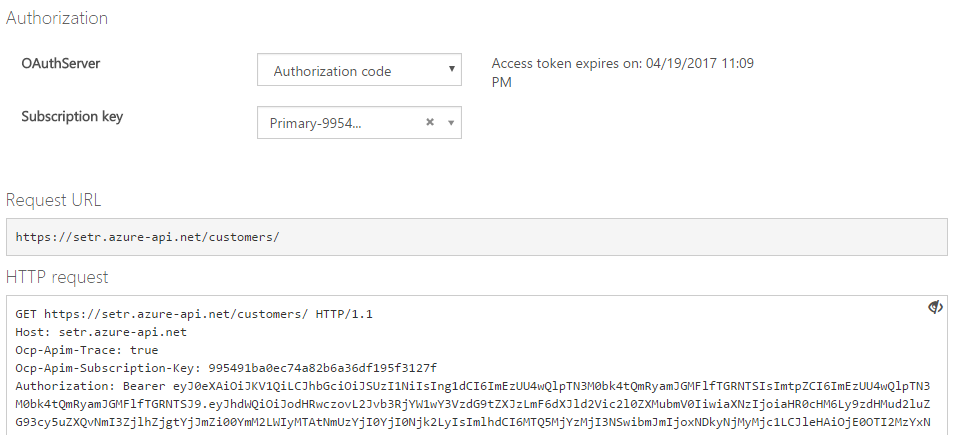
1. Go to the Publisher Portal, click APIs in the left menu, and click the created API to view and configure its settings.  
   
2. Navigate to the Security tab, check the OAuth 2.0 checkbox, select the desired authorization server from the Authorization server drop-down, and click Save.  
   

## Successfully call the Web API from the Developer portal

Now that the OAuth 2.0 authorization is configured on the API, its operations can be successfully called from the Developer portal.

1. Navigate back to the Add two integers operation of the calculator service in the developer portal and click Try it.

**Note** the new item in the Authorization section corresponding to the authorization server you just added.  


1. Select Authorization code from the authorization drop-down list and enter the credentials of the account to use. If you are already signed in with the account you may not be prompted.  
   
2. Click Send and note the Response status of 200 OK and the results of the operation in the response content.  
   