# Lab: Creating a multi-tier solution by using services in Azure

## Microsoft Azure user interface

Given the dynamic nature of Microsoft cloud tools, you might experience Azure user interface (UI) changes after the development of this training content. These changes might cause the lab instructions and steps to not match up.

The Microsoft Worldwide Learning team updates this training course as soon as the community brings needed changes to our attention. However, because cloud updates occur frequently, you might encounter UI changes before this training content is updated. **If this occurs, adapt to the changes and work through them in the labs as needed.**

## Instructions

### Before you start

#### Sign in to the lab virtual machine

* Sign in to your **Windows 10** virtual machine by using the following credentials:
  + **Username**: Admin
  + **Password**: Pa55w.rd

**Note**: Lab virtual machine sign-in instructions will be provided to you by your instructor.

#### Review installed applications

* Observe the taskbar located at the bottom of your **Windows 10** desktop. The taskbar contains the icons for the applications you will use in this lab:
  + Microsoft Edge
  + File Explorer
  + Microsoft Azure Storage Explorer

#### Download the lab files

1. On the taskbar, select the **Windows PowerShell** icon.
2. In the PowerShell command prompt, change the current working directory to the **Allfiles (F):\** path:

* cd F:

1. Within the command prompt, enter the following command and press Enter to Clone the **microsoftlearning/AZ-203-DevelopingSolutionsForAzure** project hosted on GitHub into the **Labfiles** directory:

* git clone --depth 1 --no-checkout https://github.com/microsoftlearning/AZ-203-DevelopingSolutionsForMicrosoftAzure .

1. Within the command prompt, enter the following command and press **Enter** to check out the lab files necessary to complete the **AZ-201.02** lab:

* git checkout master -- Allfiles/\*

1. Close the currently running **Windows PowerShell** command prompt application.

### Exercise 1: Creating an Azure Search service in the portal

#### Task 1: Open the Azure portal

1. On the taskbar, select the **Microsoft Edge** icon.
2. In the open browser window, navigate to the [**Azure portal**](https://portal.azure.com) (portal.azure.com).
3. At the sign-in page, enter the **email address** for your Microsoft account.
4. Select **Next**.
5. Enter the **password** for your Microsoft account.
6. Select **Sign in**.

* **Note**: If this is your first time signing in to the **Azure portal**, a dialog box will display offering a tour of the portal. Select **Get Started** to skip the tour and begin using the portal.

#### Task 2: Create an Azure Search account

1. In the left navigation pane of the portal, select **+ Create a resource**.

* **Note**: If you cannot find the link, the Create a resource icon is a plus-sign character located on the left side of the portal.

1. At the top of the **New** blade, locate the **Search the Marketplace** field.
2. In the search field, enter the text **Search** and press Enter.
3. In the **Everything** search results blade, select the **Azure Search** result.
4. In the **Azure Search** blade, select **Create**.
5. In the **New Search Service** blade, observe the tabs at the top of the blade, such as **Basics**.

* **Note**: Each tab represents a step in the workflow to create a new **search account**. At any time, you can select **Review + create** to skip the remaining tabs.

1. In the **Basics** tab, perform the following actions:
   1. Leave the **Subscription** field set to its default value.
   2. In the **Resource group** section, select **Create new**, in the pop-up field enter **MultiTierService**, and then select **OK**.
   3. In the **URL** field, enter the value **prodsearch[*your name in lowercase*]**.
   4. In the **Location** list, select **East US**.
   5. Select the **Pricing tier** link. In the **Pricing tier** blade, select **Basic** and then select **Select**.
   6. Select **Review + Create**.
2. In the **Review + Create** tab, review the options that you selected during the previous steps.
3. Select **Create** to create the search account by using your specified configuration.
4. Wait for the creation task to complete before you move forward with this lab.
5. In the left navigation pane of the portal, select **Resource groups**.
6. In the **Resource groups** blade, select the **MultiTierService** resource group that you created earlier in this lab.
7. In the **MultiTierService** blade, select the **prodsearch\*** Search service that you created earlier in this lab.
8. In the **Search Service** blade, in the **Settings** section, select the **Keys** link.
9. In the **Keys** section, select any one of the keys and record the value. You will use this value later in the lab.

* **Note**: It does not matter which connection string you choose to use. They are interchangeable.

#### Task 3: Create an index

1. In the left navigation pane of the portal, select **Resource groups**.
2. In the **Resource groups** blade, select the **MultiTierService** resource group that you created earlier in this lab.
3. In the **MultiTierService** blade, select the **prodsearch\*** Search service that you created earlier in this lab.
4. In the **Search Service** blade, select **Add index**.
5. In the **Add index** blade, perform the following actions:
   1. In the **Index name** field, enter **retail**.
   2. In the **Key** list, select **id**.
   3. Leave the **Suggester name** field blank.
   4. Leave the **Search mode** list blank.
6. Within the **Add index** blade, a list of fields is displayed. Perform the following actions to configure the **id** field:
   1. In the **Field Name** field, observe the hard-coded value of **id**.
   2. In the **Type** list, observe the hard-coded option of **Edm.String**.
   3. In the **Retrievable** option, observe that it is hard-coded to **true**.
   4. Leave **Filterable** unselected.
   5. Select **Sortable**.
   6. Leave **Facetable** unselected.
   7. Leave **Searchable** unselected.
   8. Select **+ Add field**.
7. Within the **Add index** blade, perform the following actions to configure the new **name** field:
   1. Select **+ Add field**.
   2. In the **Field Name** field, enter **name**.
   3. In the **Type** list, select **Edm.String**.
   4. Select **Retrievable**.
   5. Leave **Filterable** unselected.
   6. Select **Sortable**.
   7. Leave **Facetable** unselected.
   8. Select **Searchable**.
   9. In the **Analyzer** list, select **Standard - Lucene**.
   10. Select **+ Add field**.
8. Within the **Add index** blade, perform the following actions to configure the new **price** field:
   1. Select **+ Add field**.
   2. In the **Field Name** field, enter **price**.
   3. In the **Type** list, select **Edm.Double**.
   4. Select **Retrievable**.
   5. Select **Filterable**.
   6. Select **Sortable**.
   7. Select **Facetable**.
   8. Select **+ Add field**.
9. Within the **Add index** blade, perform the following actions to configure the new **quantity** field:
   1. Select **+ Add field**.
   2. In the **Field Name** field, enter **quantity**.
   3. In the **Type** list, select **Edm.Int32**.
   4. Select **Retrievable**.
   5. Select **Filterable**.
   6. Select **Sortable**.
   7. Select **Facetable**.
   8. Select **+ Add field**.
10. Within the **Add index** blade, perform the following actions to configure the new **manufacturer** field:
    1. Select **+ Add field**.
    2. In the **Field Name** field, enter **manufacturer**.
    3. In the **Type** list, select **Edm.String**.
    4. Select **Retrievable**.
    5. Select **Filterable**.
    6. Select **Sortable**.
    7. Select **Facetable**.
    8. Leave **Searchable** unselected.
11. Within the **Add index** blade, select **Create**.

#### Review

In this exercise, you created a new Azure Search account and built an index within the account.

### Exercise 2: Index an Azure Storage table in Azure Search

#### Task 1: Create an Azure Storage account

1. In the left navigation pane of the portal, select **+ Create a resource**.
2. At the top of the **New** blade, locate the **Search the Marketplace** field.
3. In the search field, enter **Storage** and press Enter.
4. In the **Everything** search results blade, select the **Storage Account** result.
5. In the **Storage Account** blade, select **Create**.
6. In the **Create Storage Account** blade, observe the tabs at the top of the blade.

* **Note**: Each tab represents a step in the workflow to create a new **storage account**. At any time, you can select **Review + create** to skip the remaining tabs.

1. In the **Basics** tab, perform the following actions:
   1. Leave the **Subscription** field set to its default value.
   2. In the **Resource group** list, select **MultiTierService**.
   3. In the **Storage Account Name** field, enter **prodstorage[*your name in lowercase*]**.
   4. In the **Location** list, select **(US) East US**.
   5. In the **Performance** section, select **Standard**.
   6. In the **Account kind** list, select **StorageV2 (general purpose v2)** .
   7. In the **Replication** list, select **Read-access geo-redundant storage (RA-GRS)**.
   8. In the **Access tier** section, ensure that **Hot** is selected.
   9. Select **Review + Create**.
2. In the **Review + Create** tab, review the options that you entered in the previous steps.
3. Select **Create** to create the Storage account by using your specified configuration.
4. Wait for the creation task to complete before you move forward with this lab.
5. In the left navigation pane of the portal, select **Resource groups**.
6. In the **Resource groups** blade, select the **MultiTierService** resource group that you created earlier in this lab.
7. In the **MultiTierService** blade, select the **prodstorage\*** storage account that you created earlier in this lab.
8. In the **Storage Account** blade, on the left side of the blade, in the **Settings** section, select the **Access keys** link.
9. In the **Access keys** blade, select any one of the keys and record the value of either of the **Connection string** fields. You will use this value later in this lab.

* **Note**: It does not matter which connection string you choose. They are interchangeable.

#### Task 2: Upload table entities to Azure Storage

1. In the left navigation pane of the portal, select **Resource groups**.
2. In the **Resource groups** blade, select the **MultiTierService** resource group that you created earlier in this lab.
3. In the **MultiTierService** blade, select the **prodstorage\*** storage account that you created earlier in this lab.
4. In the **Storage Account** blade, on the left side of the blade, in the **Table service** section, select the **Tables** link.
5. In the **Tables** section, select **+ Table**.
6. In the **Add table** window, perform the following actions:
   1. In the **Table** **Name** field, enter **products**.
   2. Select **OK**.
7. Back in the **Tables** section, on the left side of the blade, select the **Overview** link.

* **Note**: You might have to scroll up or down the menu on the left side of the blade.

1. Back in the **Overview** section, select **Open in Explorer**.
2. In the **Azure Storage Explorer** window, select the **Open Azure Storage Explorer** link.

* **Note**: If this is your first time opening the **Azure Storage Explorer** by using the portal, you might be prompted to allow the portal to open these types of links in the future. You should accept the prompt.

1. In the **Azure Storage Explorer** application that appears, locate and expand the **prodstorage\* S**torage account that you created earlier in this lab.
2. Within the **prodstorage\*** storage account, locate and expand the **Tables** node.
3. Within the **Tables** node, select the **products** table that you created earlier in this lab.
4. In the **Products Table** tab, select **Import**.
5. In the **File Explorer** dialog box that opens, perform the following actions:
   1. Go to **Allfiles (F):\Allfiles\Labs\06\Starter**.
   2. Select the **products.csv** file.
   3. Select **Open**.
6. In the **Import Entities** window that appears, select **Insert**.
7. Wait for the table entities to be uploaded before you continue with this lab.
8. Observe the five entities that were added to your **products** table.
9. Return to the browser window showing the **Azure portal**.

#### Task 3: Create an Azure Search indexer

1. In the left navigation pane of the portal, select **Resource groups**.
2. In the **Resource groups** blade, select the **MultiTierService** resource group that you created earlier in this lab.
3. In the **MultiTierService** blade, select the **prodsearch\*** Search service you created earlier in this lab.
4. In the **Search Service** blade, select **Import Data**.
5. In the **Import Data** blade, observe the tabs at the top of the blade.
6. In the **Connect to your data** tab, perform the following actions:
   1. In the **Data Source** list, select **Azure Table Storage**.
   2. In the **Name** field, enter **tabledatasource**.
   3. In the **Connection string** field, select **Choose an existing connection**. In the **Choose storage account** window, select the **prodstorage\*** Storage account that you created earlier in this lab.
   4. In the **Table name** field, enter **products**.
   5. Leave the **Query** field empty.
   6. Leave the **Description** field empty.
   7. Select **Next: Add cognitive search (Optional)**.
   * **Note**: Azure Search validates your settings at each step. It can take a few minutes to complete validation and move to the next tab in the list.
7. In the **Add cognitive search (optional)** tab, select **Skip to: Customize target index**.
8. In the **Customize target index** tab, perform the following actions:
   1. In the **Index name** field, enter **products**.
   2. In the **Key** list, select **RowKey**.
9. In the **Customize target index** tab, a list of fields is displayed. Leave the configuration of the **PartitionKey**, **RowKey**, **ETag,** and **Timestamp** fields set to their default values.
10. In the **Customize target index** tab, perform the following actions to configure the **Key** field:
    1. In the **Field Name** field, observe the hard-coded value of **Key**.
    2. In the **Type** list, observe the hard-coded option of **Edm.String**.
    3. Leave **Retrievable** selected.
    4. Leave **Filterable** unselected.
    5. Select **Sortable**.
    6. Leave **Facetable** unselected.
    7. Leave **Searchable** unselected.
11. In the **Customize target index** tab, perform the following actions to configure the **name** field:
    1. In the **Field Name** field, observe the hard-coded value of **name**.
    2. In the **Type** list, observe the hard-coded option of **Edm.String**.
    3. Select **Retrievable**.
    4. Leave **Filterable** unselected.
    5. Select **Sortable**.
    6. Leave **Facetable** unselected.
    7. Select **Searchable**.
    8. In the **Analyzer** list, select **Standard - Lucene**.
12. In the **Customize target index** tab, perform the following actions to configure the **price** field:
    1. In the **Field Name** field, observe the hard-coded value of **price**.
    2. In the **Type** list, observe the hard-coded option of **Edm.Double**.
    3. Select **Retrievable**.
    4. Select **Filterable**.
    5. Select **Sortable**.
    6. Select **Facetable**.
13. In the **Customize target index** tab, perform the following actions to configure the **quantity** field:
    1. In the **Field Name** field, observe the hard-coded value of **quantity**.
    2. In the **Type** list, observe the hard-coded option of **Edm.Int32**.
    3. Select **Retrievable**.
    4. Select **Filterable**.
    5. Select **Sortable**.
    6. Select **Facetable**.
14. In the **Customize target index** tab, select **Next: Create an indexer**.

* **Note**: Azure Search validates your settings at each step. It can take a few minutes to complete validation and move to the next tab in the list.

1. In the **Create an indexer** tab, perform the following actions:
   1. In the **Name** field, enter **tableindexer**.
   2. In the **Schedule** section, select **Custom**.
   3. In the **Interval** field, enter **5**.
   4. Set the **Start time** field to midnight UTC on today’s date.
   5. Leave the **Track deletions** field set to its default value.
   6. Leave the **Description** field blank.
   7. Select **Submit**.
2. Back in the **Search Service** blade, select the **Indexers** tab.

* **Note**: If you do not see your indexer yet, you may need to refresh the blade.

1. In the **Indexers** tab, select the **tableindexer** indexer that you created earlier in this lab.
2. In the **Indexer** blade, perform the following actions:
   1. Select **Run**.
   2. When prompted for confirmation, select **Yes**.
   3. Close the **Indexer** blade.
3. Wait for the indexer to finish running and then select **Refresh** at the top of the blade.

* **Note**: It can take from one to five minutes for the indexer to run. You will know that the indexer was successful if its status is listed as **Success** in the **Search Service** blade.

1. In the **Indexers** tab, observe the metadata of the **tableindexer** indexer, such as the document count and the status of the last indexing operation.

#### Task 4: Validate the indexed table data

1. In the **Search Service** blade, select **Search Explorer** at the top of the blade.
2. In the **Search Explorer** blade, select **Search**.
3. Observe the results of a search for all documents.
4. In the **Query string** field, enter the following query and then press **Search**:

* search=seat

1. Observe the results of the search query.
2. In the **Query string** field, enter the following query and then press **Search**:

* $filter=price lt 100

1. Observe the results of the search query.
2. In the **Query string** field, enter the following query and then press **Search**:

* facet=quantity,interval:25

1. Observe the results of the search query.
2. In the **Query string** field, enter the following query and then press **Search**:

* $filter=quantity gt 25&facet=price,values:100|1000|10000

1. Observe the results of the search query.
2. Close the **Search Explorer** blade.

#### Task 5: Retrieve your Azure Search base URL

1. In the left navigation pane of the portal, select **Resource groups**.
2. In the **Resource groups** blade, select the **MultiTierService** resource group that you created earlier in this lab.
3. In the **MultiTierService** blade, select the **prodsearch\*** Search service that you created earlier in this lab.
4. In the **Search Service** blade, copy the value of the **URL** field. You will use this value later in this lab.

#### Review

In this exercise, you created an Azure Storage account and indexed a Storage table within the account by using Azure Search. After the table was indexed, you were able to issue search queries against a copy of the entities in the Storage table.

### Exercise 3: Build an API proxy tier by using Azure API Management

#### Task 1: Create an API Management resource

1. In the left navigation pane of the portal, select **+ Create a resource**.
2. At the top of the **New** blade, locate the **Search the Marketplace** field.
3. In the search field, enter **API** and press Enter.
4. In the **Everything** search results blade, select the **API Management** result.
5. In the **API Management** blade, select **Create**.
6. In the **API Management Service** blade, perform the following actions:
   1. In the **Name** field, enter **prodapi[*your name in lowercase*]**.
   2. Leave the **Subscription** field set to its default value.
   3. In the **Resource group** list, select **MultiTierService**.
   4. In the **Location** list, select **East US**.
   5. In the **Organization name** field, enter **Contoso**.
   6. Leave the **Administrator email** field set to its default value.
   7. In the **Pricing tier** list, select **Developer (No SLA)**.
   8. Select **Create**.
7. Wait for the creation task to complete before you move forward with this lab.

* **Note**: It usually takes between 20 and 30 minutes to create an API Management service.

#### Task 2: Define a new API

1. In the left navigation pane of the portal, select **Resource groups**.
2. In the **Resource groups** blade, select the **MultiTierService** resource group that you created earlier in this lab.
3. In the **MultiTierService** blade, select the **prodapi\*** API Management account that you created earlier in this lab.
4. In the **API Management Service** blade, on the left side of the blade, in the **API Management** section, select **APIs** .
5. In the **Add a new API** section, select **Blank API**.
6. In the **Create a blank API** window, perform the following actions:
   1. In the **Display name** field, enter **Search API**.
   2. In the **Name** field, enter **search-api**.
   3. In the **Web service URL** field, enter the URL from the **Search Service URL** field that you copied earlier in this lab.
   4. Append the value of the **Web service URL** field with the following relative URL:

* /indexes/products/docs
* **Note**: For example, if your web service URL is https://prodsearchstudent.search.windows.net, then your new URL will be https://prodsearchstudent.search.windows.net/indexes/products/docs.

1. In the **API URL suffix** field, enter **search**.
2. In the **Products** field, select both **Starter** and **Unlimited**.
3. Select **Create.**
4. Wait for the new API to finish being created.
5. In the **Design** tab, select **+ Add operation**.
6. In the **Add operation** section, perform the following actions:
   1. In the **Display name** field, enter **List All Documents**.
   2. In the **Name** field, enter **list-all-documents**.
   3. In the **URL** list, select **GET**.
   4. In the **URL** field, enter **/**.
   5. Select **Save**.
7. Back in the **Design** tab, in the list of operations, select **All Operations**.
8. In the **Design** section for **All Operations**, locate the **Inbound processing** tile and select **+ Add policy**.
9. In the **Add inbound policy** section, select the **Set headers** tile.
10. In the **Inbound processing, Set Headers** section, perform the following actions:
    1. In the **Name** field, enter **api-key**.
    2. In the **Value** field, select the list, select **+ Add Value**, and then enter the value for the **Search Service Key** that you recorded earlier in this lab.
    3. In the **Action** list, select the **override** option.
    4. Select **Save**.
11. Back in the **Design** tab, in the list of operations, select **All Operations**.
12. In the **Design** section for **All Operations**, locate the **Inbound processing** tile and select **+ Add policy**.
13. In the **Add inbound policy** section, select the **Set query parameters** tile.
14. In the **Inbound processing, Set Query Parameters** section, perform the following actions:
    1. In the **Name** field, enter **api-version**.
    2. In the **Value** field, enter **2017-11-11**.
    3. In the **Action** list, select **override**.
    4. Select **Save**.
15. Back in the **Design** tab, in the list of operations, select **List All Documents**.
16. In the **Design** section for the **List All Documents** operation, locate the **Inbound processing** tile and select the **+ Add policy** button.
17. In the **Add inbound policy** section, select the **Set query parameters** tile.
18. In the **Inbound processing, Set Query Parameters** section, perform the following actions:
    1. In the **Name** field, enter **search**.
    2. In the **Value** field, enter **\***.
    3. In the **Action** list, select **override**.
    4. Select **Save**.
19. Back in the **Design** tab, in the list of operations, select **List All Documents**.
20. Select the **Test** tab.
21. Select the **List All Documents** operation.
22. In the **List All Documents** section, select **Send**.
23. Observe the results of the API request.

* **Note**: Observe how there is a large amount of Azure Search metadata in the response. You might not want API users to know the implementation details that occur behind the scenes. In the next task, you will obfuscate much of this data.

1. Select the **Design** tab to return to the list of operations.

#### Task 3: Manipulate an API response

1. Back in the **Design** tab, in the list of operations, select **List All Documents** .
2. In the **Design** section for the **List All Documents** operation, locate the **Outbound processing** tile and select **+ Add policy**.
3. In the **Add outbound policy** section, select the **Other policies** tile.
4. In the policy code editor, locate the following block of XML content:

* <outbound>  
   <base />  
  </outbound>

1. Replace that block of XML with the following XML:

* <outbound>  
   <base />  
   <set-body>  
   @{   
   var response = context.Response.Body.As<JObject>();  
   return response.Property("value").Value.ToString();  
   }  
   </set-body>  
  </outbound>

1. In the policy code editor, select **Save**.
2. Back in the **Design** tab, in the list of operations, select **List All Documents**.
3. In the **Design** section for the **List All Documents** operation, locate the **Outbound processing** tile and select **+ Add policy**.
4. In the **Add outbound policy** section, select the **Set headers** tile.
5. In the **Outbound processing, Set Headers** section, perform the following actions:
   1. In the **Name** field, enter **preference-applied**.
   2. In the **Action** list, select **delete**.
   3. Select **+** **Add header**.
   4. In the new **Name** field, enter **odata-version**.
   5. In the new **Action** list, select **delete**.
   6. Select **+ Add header**.
   7. In the new **Name** field, enter **powered-by**.
   8. In the new **Value** field, select the list, select the **+ Add Value** link, and then enter **Contoso**.
   9. In the new **Action** list, select **override**.
   10. Select **Save**.
6. Back in the **Design** tab, in the list of operations, select **List All Documents**.
7. Select the **Test** tab.
8. Select the **List All Documents** operation.
9. In the **List All Documents** section, select **Send**.
10. Observe the results of the API request.

* **Note**: You will observe that the **preference-applied** and **odata-version** headers that you specified have been deleted and replaced with a new **powered-by** header. You will also notice that the response doesn’t contain context data about the OData response, but instead contains a flattened JSON array as the response body.

#### Review

In this exercise, you built a proxy tier between your Azure Search account and any developers who wish to make search queries.

### Exercise 4: Create new table entities by using Azure Logic Apps

#### Task 1: Create a Logic Apps resource

1. In the left navigation pane of the portal, select **+ Create a resource**.
2. At the top of the **New** blade, locate the **Search the Marketplace** field.
3. In the search field, enter **Logic** and press Enter.
4. In the **Everything** search results blade, select the **Logic App** result.
5. In the **Logic App** blade, select **Create**.
6. In the **Logic App** blade, perform the following actions:
   1. In the **Name** field, enter **prodworkflow[*your name in lowercase*]**.
   2. Leave the **Subscription** field set to its default value.
   3. In the **Resource group** section, select **Use existing** and then select the **MultiTierService** option from the list.
   4. In the **Location** list, select **East US**.
   5. In the **Log Analytics** section, select **Off**.
   6. Select **Create**.
7. Wait for the creation task to complete before you move forward with this lab.

#### Task 2: Create a trigger for Logic Apps workflow

1. In the left navigation pane of the portal, select **Resource groups**.
2. In the **Resource groups** blade, select the **MultiTierService** resource group that you created earlier in this lab.
3. In the **MultiTierService** blade, select the **prodworkflow\*** logic app that you created earlier in this lab.
4. In the **Logic Apps Designer** blade, select the **Blank Logic App** template.
5. In the **Designer** area, perform the following actions to add a **HTTP Trigger**:
   1. In the **Search connectors and triggers** field, enter **HTTP**.
   2. In the **Triggers** result list, select **When a HTTP request is received**.
   3. Select **Use sample payload to generate schema**.
   4. In the **Enter or paste a sample JSON payload** window, enter the following JSON object:

* {   
   "id": "",  
   "manufacturer": "",  
   "price": 0.00,  
   "quantity": 0,  
   "name": ""  
  }

1. Select **Done**.
2. Observe the schema in the **Request Body JSON Schema** field. This schema is built by Azure automatically based on the JSON content that you entered in the previous step.

#### Task 3: Build a connector for Azure Storage

1. In the **Designer** area, select **+ New step**.
2. In the **Designer** area, perform the following actions to add an **Insert or Replace Entity Action**:
   1. In the **Search connectors and triggers** field, enter **Table**.
   2. In the category list, select **Azure Table Storage**.
   3. In the **Actions** result list, select **Insert or Replace Entity**.
   4. In the **Connection Name** field, enter **tableconnection**.
   5. In the **Storage Account** section, select the **prodstorage\*** Storage account that you created earlier in this lab.
   6. Select **Create**.
   7. Wait for the connector resource to finish creating.
   * **Note**: These resources take one to five minutes to create.
3. In the **Table** list, select **products**.
4. On the right side of the **Partition Key** field, in the **Dynamic content** pane, within the **When a HTTP request is received** category, select **manufacturer**.
5. On the right side of the **Row Key** field, in the **Dynamic content** pane, within the **When a HTTP request is received** category, select **id**.
6. On the right side of the **Entity** field, in the **Dynamic content** pane, within the **When a HTTP request is received** category, select **Body**.

#### Task 4: Build a HTTP response action

1. In the **Designer** area, select **+ New step**.
2. In the **Designer** area, perform the following actions to add a **Response Action**:
   1. In the **Search connectors and triggers** field, enter **Response**.
   2. In the **Actions** result list, select **Response**.
   3. In the **Status Code** field, enter **201**.
   4. On the right side of the **Body** field, in the **Dynamic content** pane, within the **Insert of Replace Entity** category, select **Body**.

#### Task 5: Retrieve a HTTP trigger POST URL

1. In the **Designer** area, select **Save**.
2. After the workflow is saved, the **HTTP POST URL** field in the **When a HTTP request is received** trigger will be updated with a new URL that you’ll need to start this workflow. Copy the URL in the **HTTP POST URL** field. You will use this URL later in this lab.

* **Note**: This is a very long URL because it includes the URL with the SAS token. Make sure that you copy the entire URL.

#### Task 6: Validate that logic app results are indexed

1. At the top of the portal, select the **Cloud Shell** icon to open a new shell instance.

* **Note**: The **Cloud Shell** icon is represented by a greater than symbol and underscore character.

1. If this is your first time opening the **Cloud Shell** by using your subscription, a **Welcome to Azure Cloud Shell** wizard will display how to configure **Cloud Shell** for first-time usage. Perform the following actions:
   1. When offered a choice between **Bash** or **PowerShell**, select **Bash**.
   2. A dialog box prompts you to create a new Storage account to begin by using the shell. Accept the default settings and select **Create storage**.
   3. Wait for the **Cloud Shell** to finish its first-time setup procedures before you move forward with the lab.
   * **Note**: If you the configuration options for the **Cloud Shell** do not appear, this is most likely because you are using an existing subscription with this course’s labs. The labs are written from the perspective that you are using a new subscription.
2. At the bottom of the portal, in the **Cloud Shell** command prompt, enter the following partial **CURL** command to issue a **HTTP POST** request to the Logic Apps instance and then press **Enter**:

* curl \  
  --header "Content-Type: application/json" \  
  --data '{"id":"6","manufacturer":"VEHTOP","price":750,"quantity":6,"name":"car roof rack"}' \

1. Next, enter the logic app’s **HTTP POST URL** that you copied earlier in this lab, ensuring that you place the URL within **quotation marks** so that the URL characters are not escaped. Press **Enter** to execute the command.

* **Note**: For example, if the URL is https://prod.eastus.logic.azure.com:443/workflows/test/triggers/invoke?api-version=2016&sig=3, then you would insert “https://prod.eastus.logic.azure.com:443/workflows/test/triggers/manual?invoke?api-version=2016&sig=3”. If you don’t include the quotation marks, you will get an error message indicating that the SAS token has been truncated and is required to issue the request. This occurs because the query string separator, **&,** is truncated if not enclosed in quotation marks.

1. In the left navigation pane of the portal, select **Resource groups**.
2. In the **Resource groups** blade, select the **MultiTierService** resource group that you created earlier in this lab.
3. In the **MultiTierService** blade, select the **prodsearch\*** Search service that you created earlier in this lab.
4. In the **Search Service** blade, select the **Indexers** tab.
5. In the **Indexers** tab, select the **tableindexer** indexer that you created earlier in this lab.
6. In the **Indexer** blade, perform the following actions:
   1. Select **Run**.
   2. When prompted for confirmation, select **Yes**.
   3. Close the **Indexer** blade.
7. Wait for the indexer to finish running, and then select **Refresh** at the top of the blade.
8. Back in the **Search Service** blade, select **Search Explorer**.
9. In the **Search Explorer** blade, select **Search**.
10. Observe the results of a search for all documents.

* **Note**: At this point, you will notice a sixth document in your index representing the new document that was inserted by the logic app.

1. In the left navigation pane of the portal, select **Resource groups**.
2. In the **Resource groups** blade, select the **MultiTierService** resource group that you created earlier in this lab.
3. In the **MultiTierService** blade, select the **prodapi\*** API Management account that you created earlier in this lab.
4. In the **API Management Service** blade, on the left side of the blade, in the **API Management** section, select **APIs** .
5. In the **APIs** section, select **Search API**.
6. In the **Design** tab, select the **Test** tab.
7. Select the **List All Documents** operation.
8. In the **List All Documents** section, select **Send**.
9. Observe the results of the API request.

* **Note**: Observe that there are six documents now instead of five.

#### Review

In this exercise, you created a logic app that takes a HTTP request and then persists the JSON body of the request as a new Azure Storage table entity.

### Exercise 5: Clean up subscription

#### Task 1: Open Cloud Shell

1. At the top of the portal, select the **Cloud Shell** icon to open a new shell instance.
2. At the bottom of the portal, in the **Cloud Shell** command prompt, type the following command and press Enter to list all resource groups in the subscription:

* az group list

1. Type the following command and press Enter to view a list of possible commands to delete a resource group:

* az group delete --help

#### Task 2: Delete resource groups

1. Type the following command and press Enter to delete the **MultiTierService** resource group:

* az group delete --name MultiTierService --no-wait --yes

1. Close the **Cloud Shell** pane at the bottom of the portal.

#### Task 3: Close active applications

1. Close the currently running **Microsoft Edge** application.
2. Close the currently running **Microsoft Azure Storage Explorer** application.

#### Review

In this exercise, you cleaned up your subscription by removing the **resource groups** used in this lab.