Azure Lab 1 will discuss about below Azure Components:

**IoT Hub**

**IoT Device Client**

**Azure WebApp**

**Azure Service Bus**

**Azure Function**

**Cosmos DB**

**SignalR**

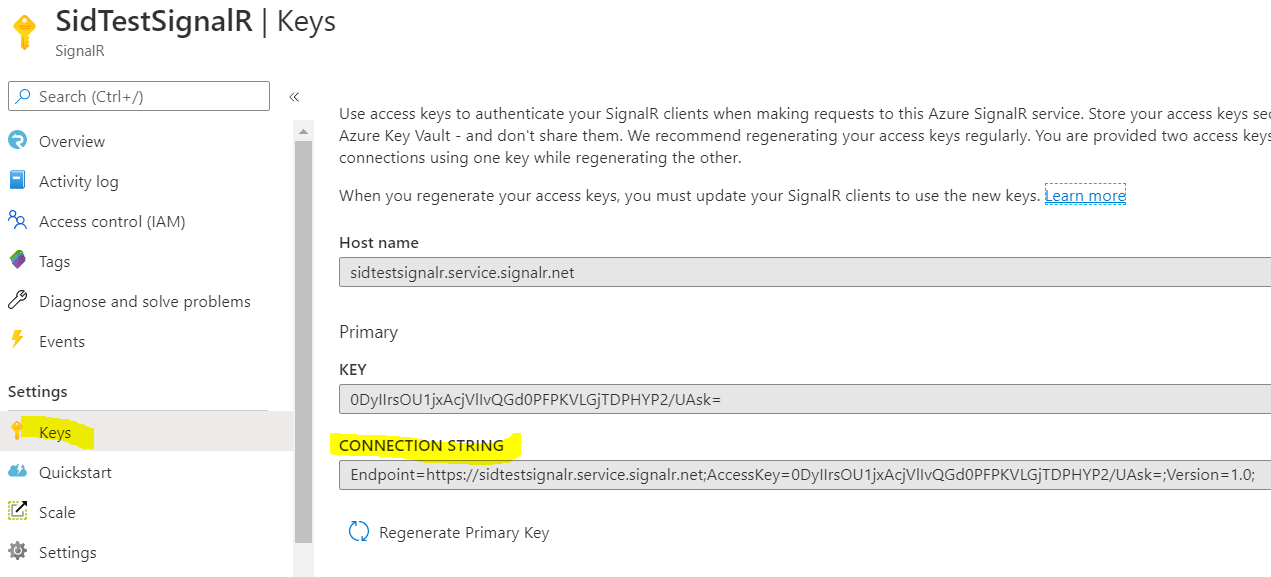
Through 3 Paths:

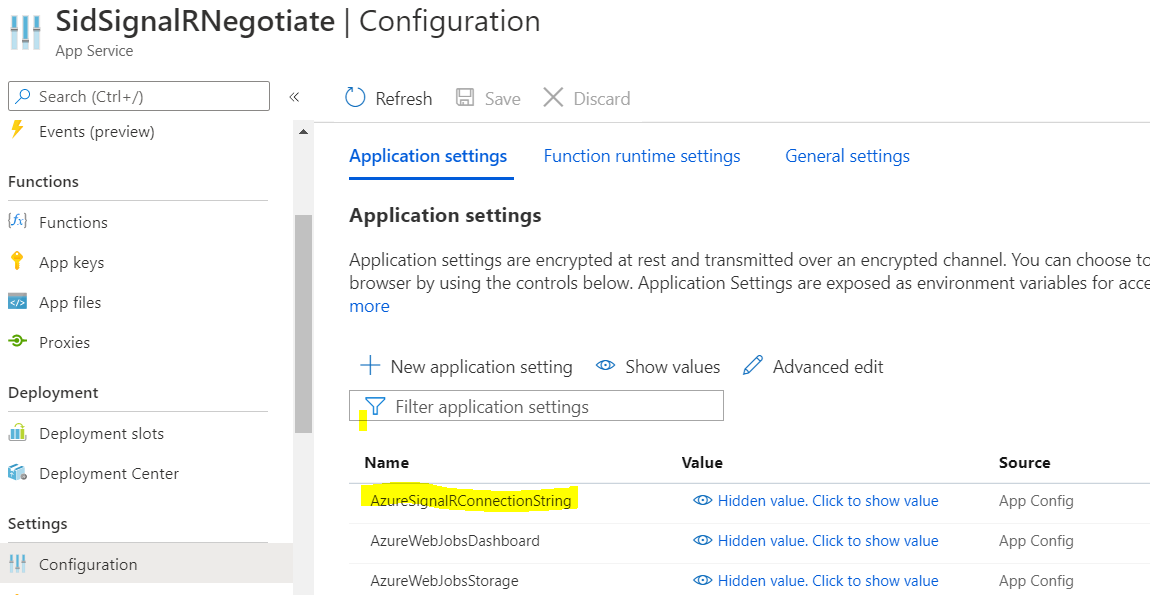
1. **Process Order - Client Panel to Admin Panel** which includes (Azure WebApp, Azure Service Bus, Azure Function, Cosmos DB and SignalR)
2. **Update Order Status - Admin Panel to Client Panel** (Azure WebApp, Azure Service Bus, Azure Function, Cosmos DB and SignalR)
3. **Process Order - IoT Device to Admin Panel** (IoT Hub, IoT Device Client, Azure WebApp, Azure Service Bus, Azure Function, Cosmos DB and SignalR)

**You can get complete code here xxxx. Download and start experiencing live learning by following below steps.**

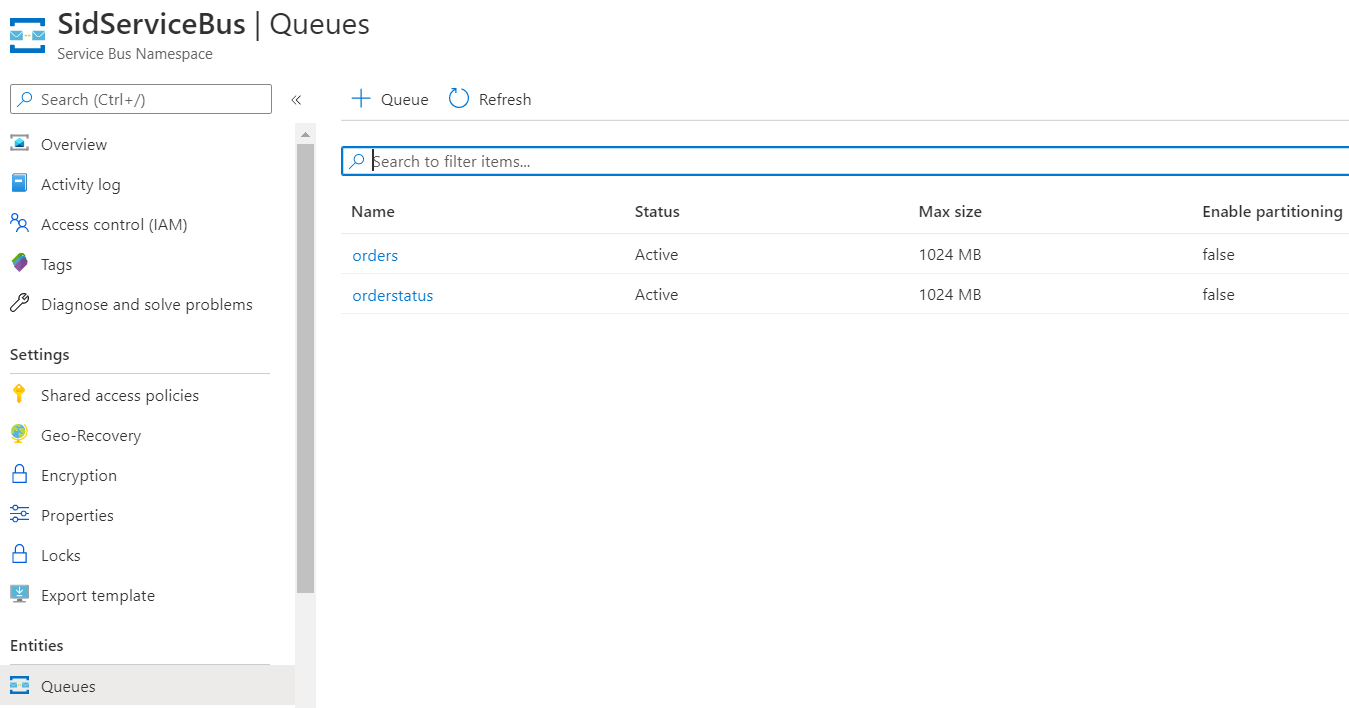
**Path 1 (Client Panel to Admin Panel) Setup:**

1. Create SignalR service on Azure portal.
2. Deploy SignalRNegotiate Function on Azure from your visual studio IDE.
3. Add Application Setting for “AzureSignalRConnectionString” as per below images:



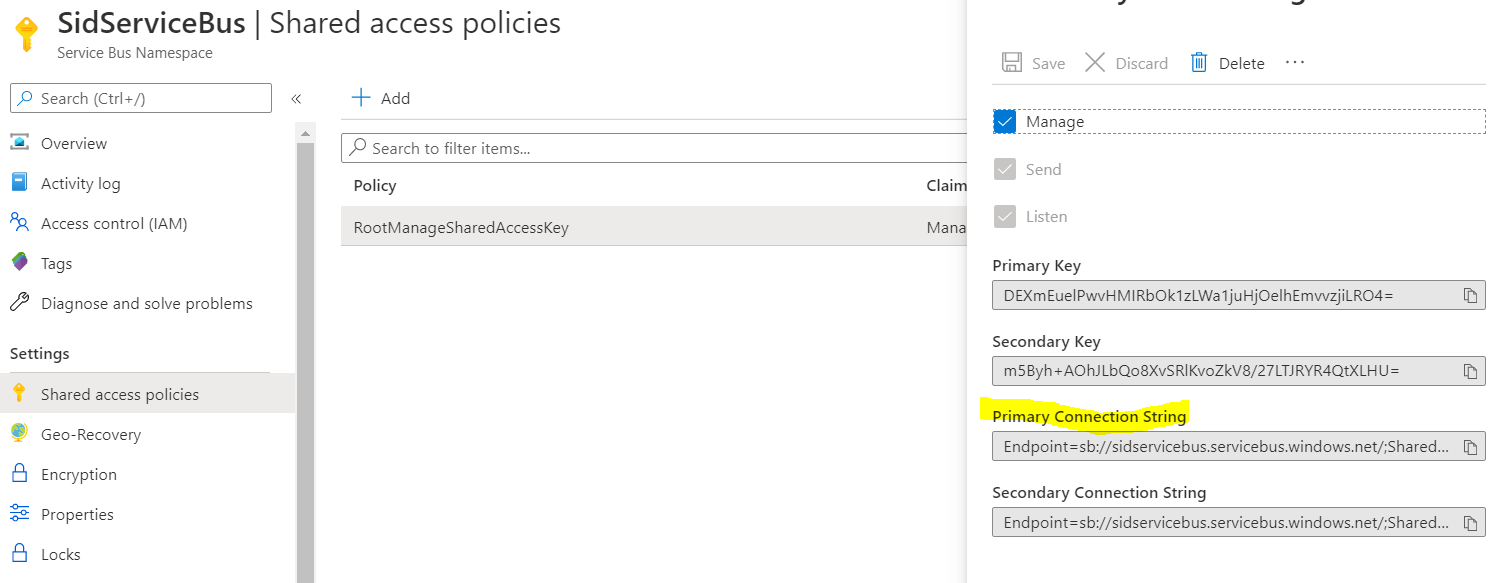


1. Create Azure Service Bus
2. Add Queue => orders and orderstatus



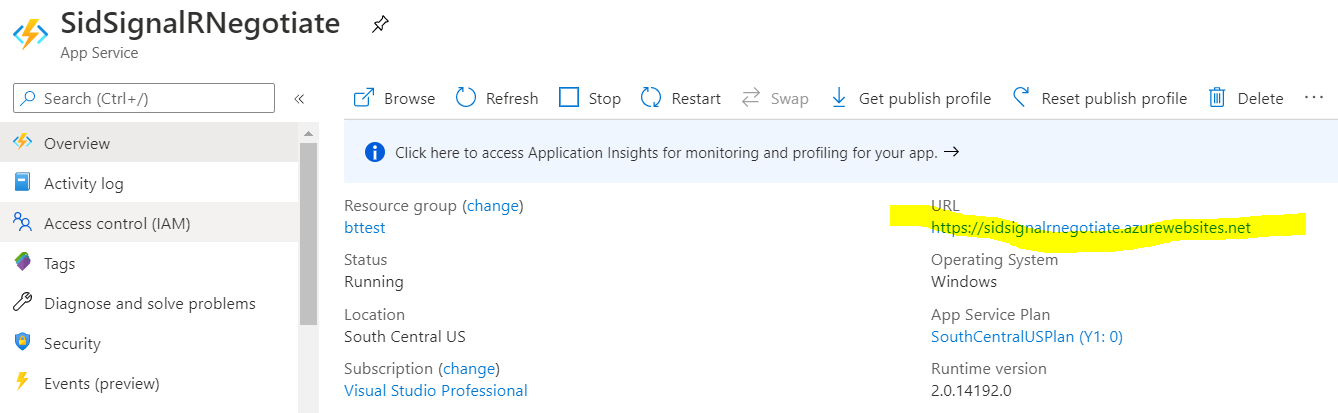
1. Open OrderDemo =>Controller=> HomeController.cs

Change service bus connection string in PlaceOrder function



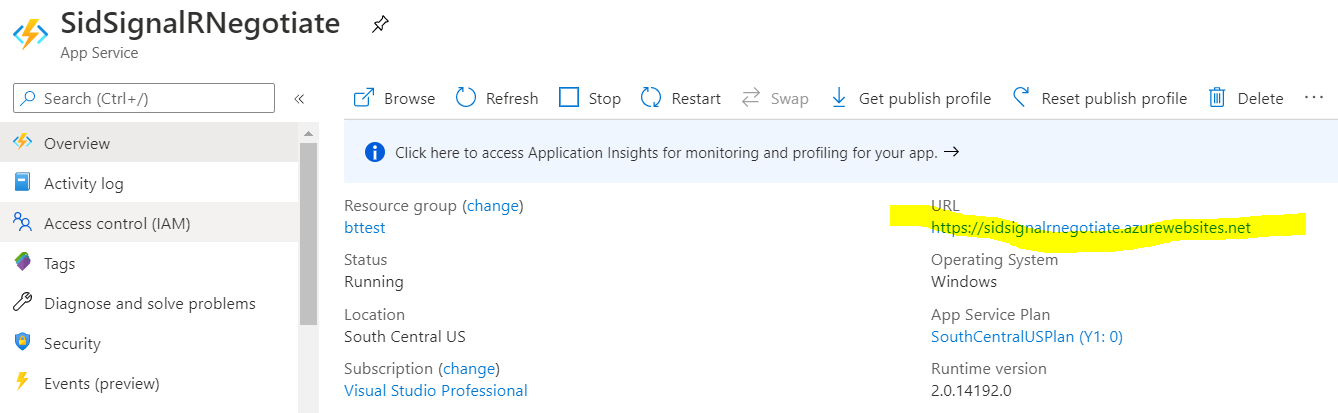
1. Open OrderDemo=>Views=>Home=>View.cshtml

Change .withUrl("https://sidsignalrnegotiate.azurewebsites.net/api/") highlighted url with SiganlR Negotiate Function Hostname.

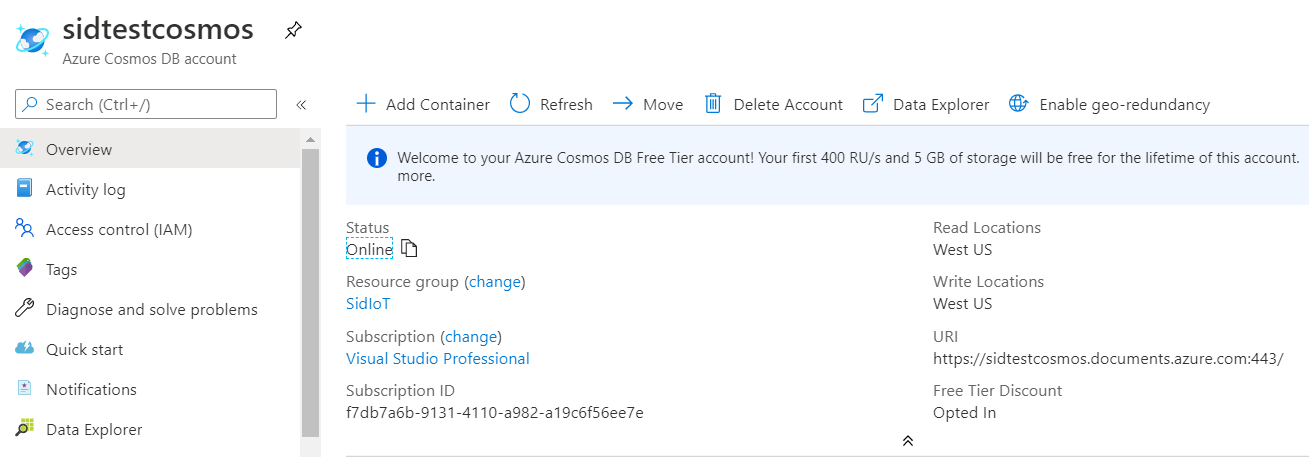


1. Open Admin=>Views=>Home=>OrderList.cshtml

Change .withUrl("https://sidsignalrnegotiate.azurewebsites.net/api/") highlighted url with SiganlR Negotiate Function Hostname.

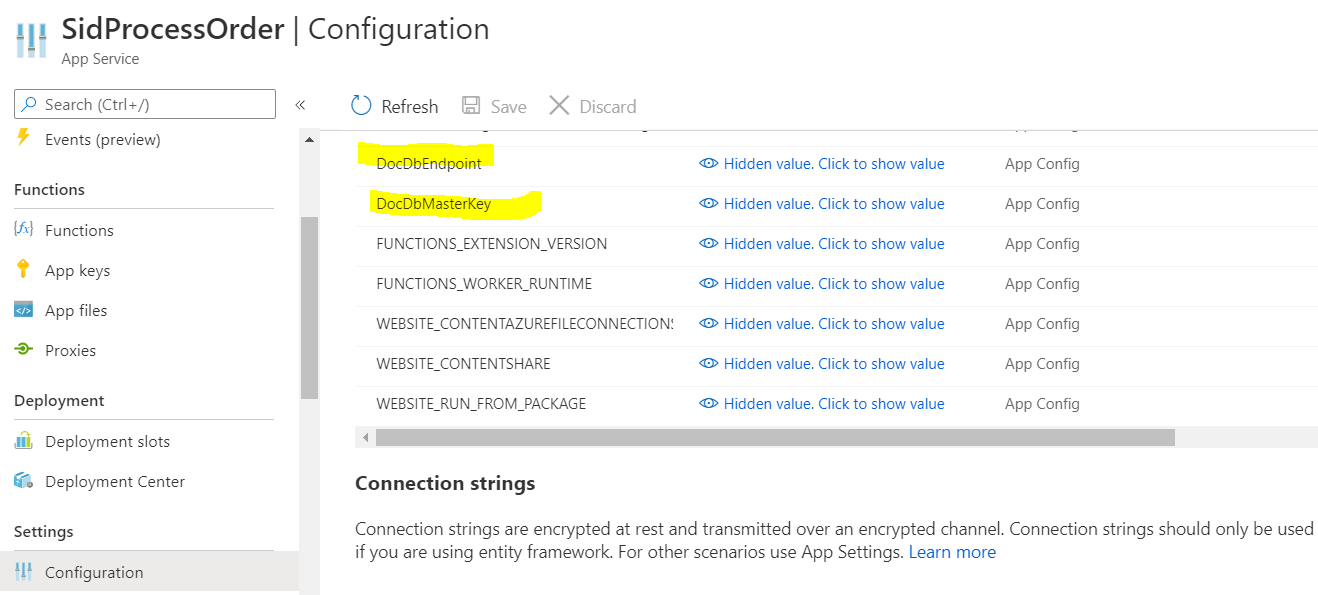


1. Create Azure CosmosDB.

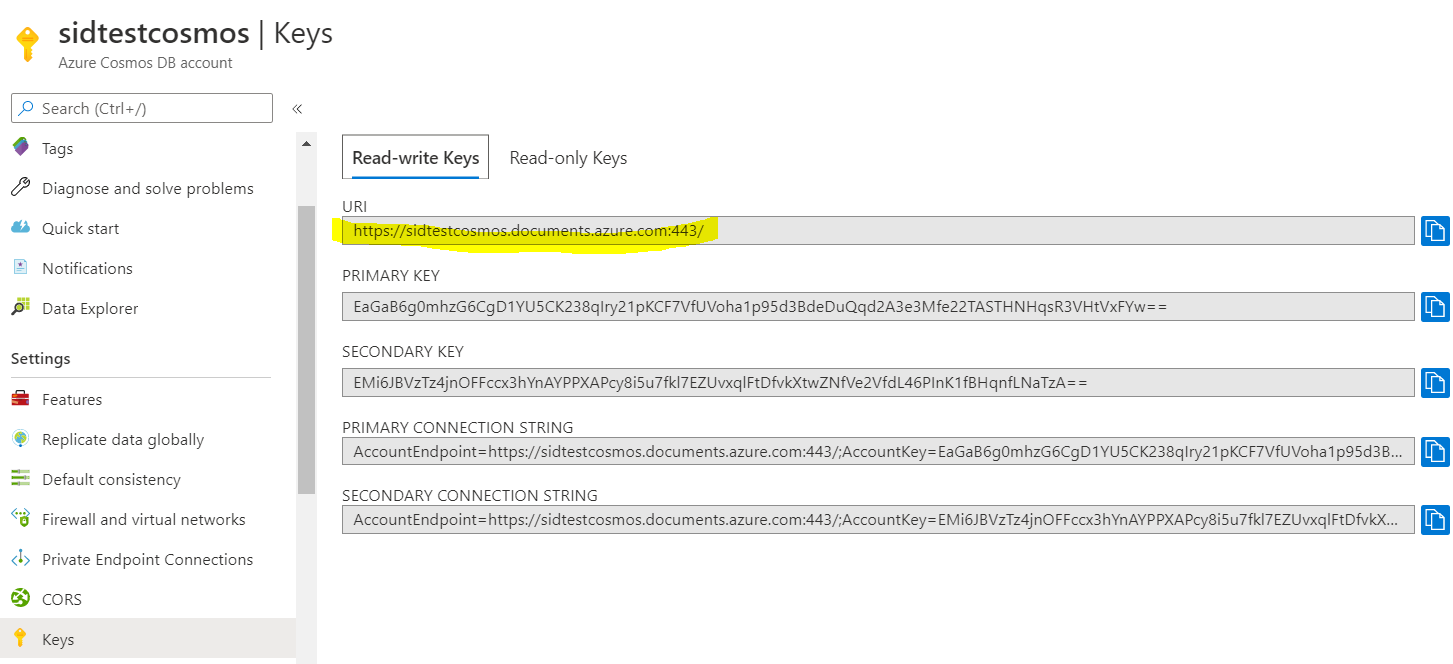


1. Publish ProcessingOrder Function app on Azure portal.

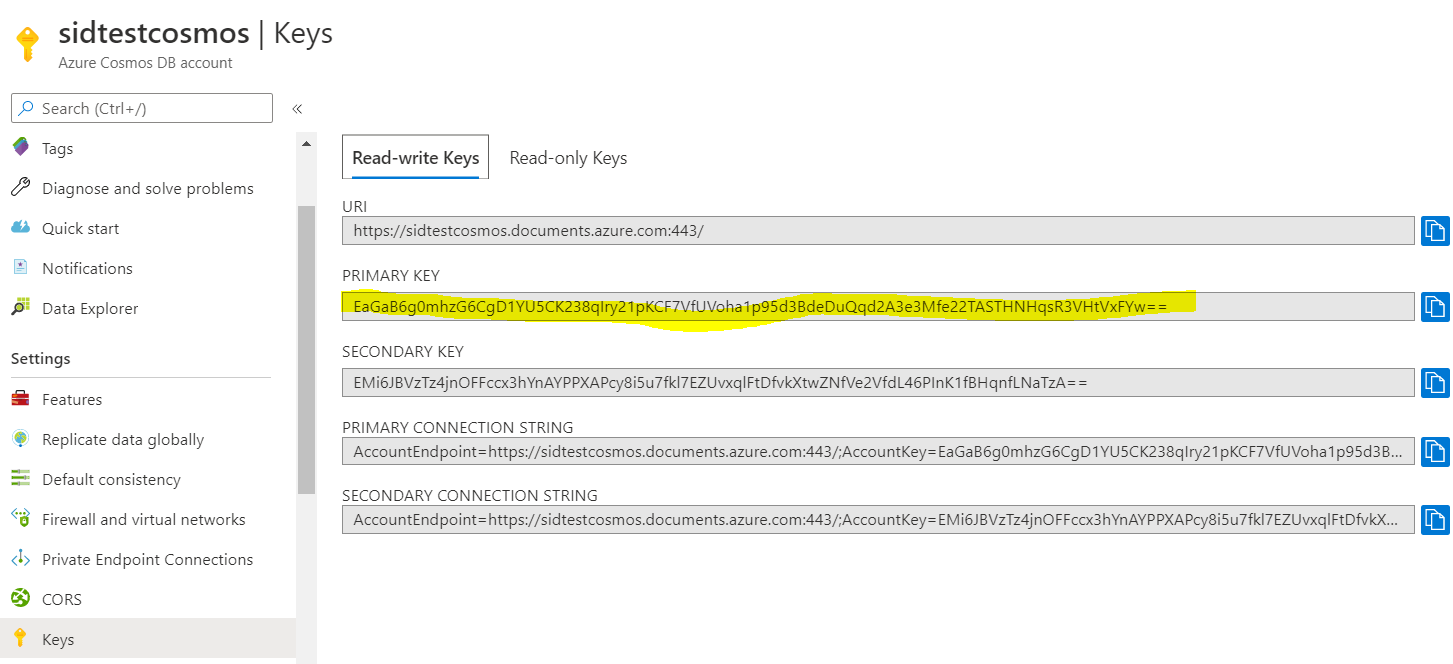
Add below 3 application settings in this function app as per below image:



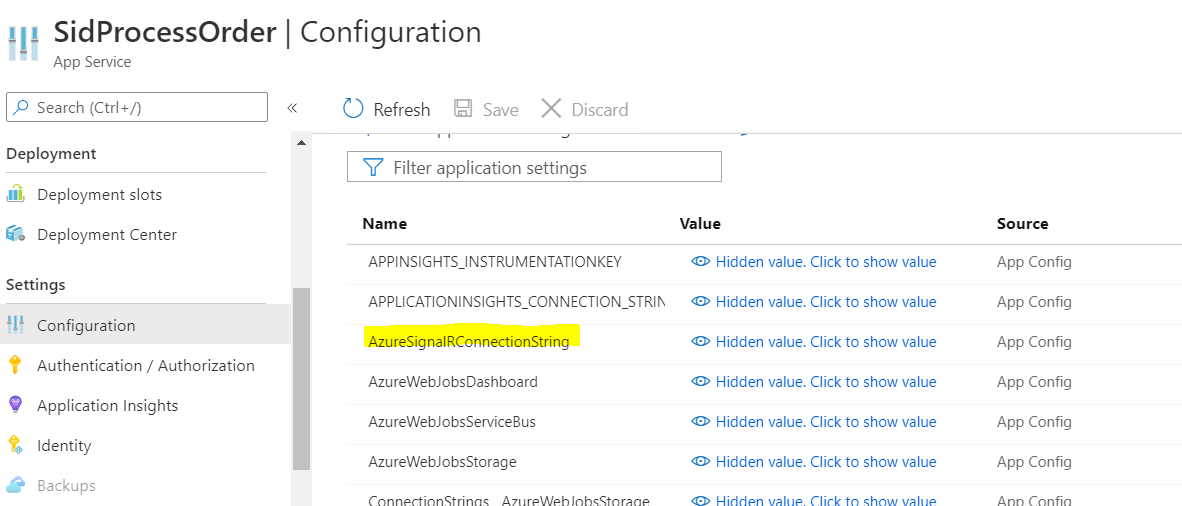
DocDbEndPoint :



DocDbMasterKey:

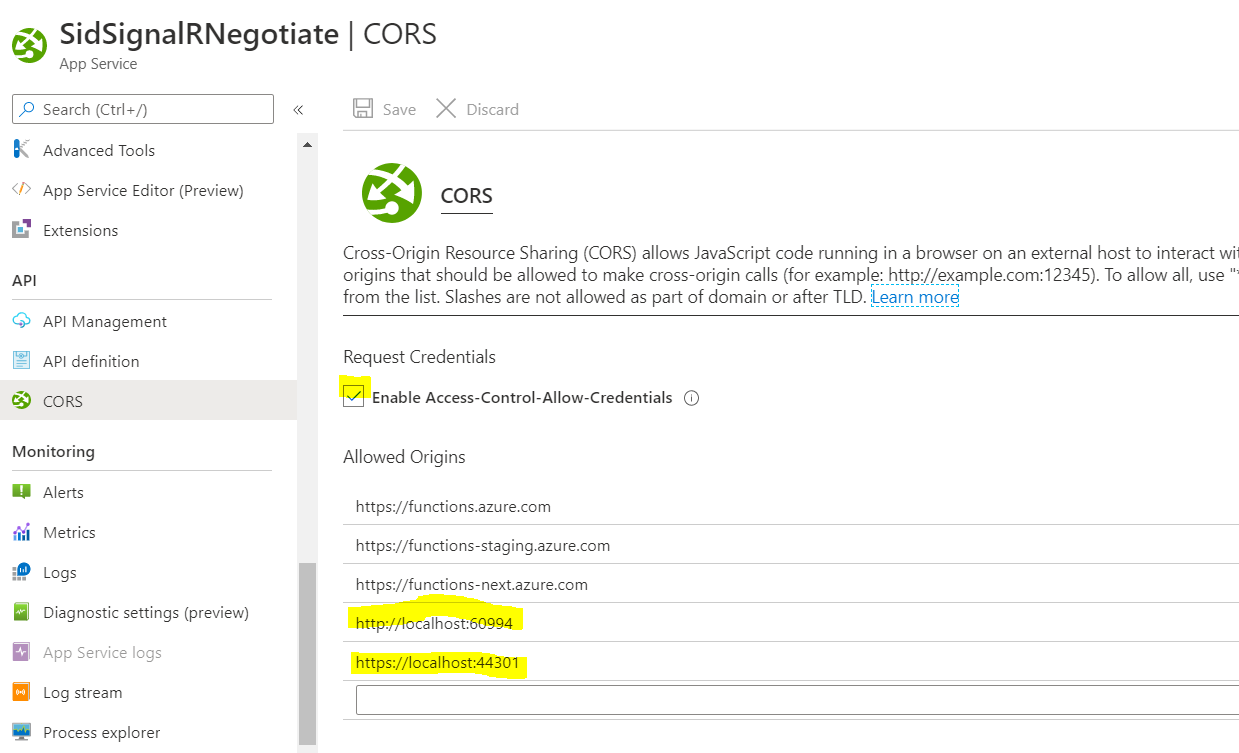


AzureSignalRConnectionString same like point 3.

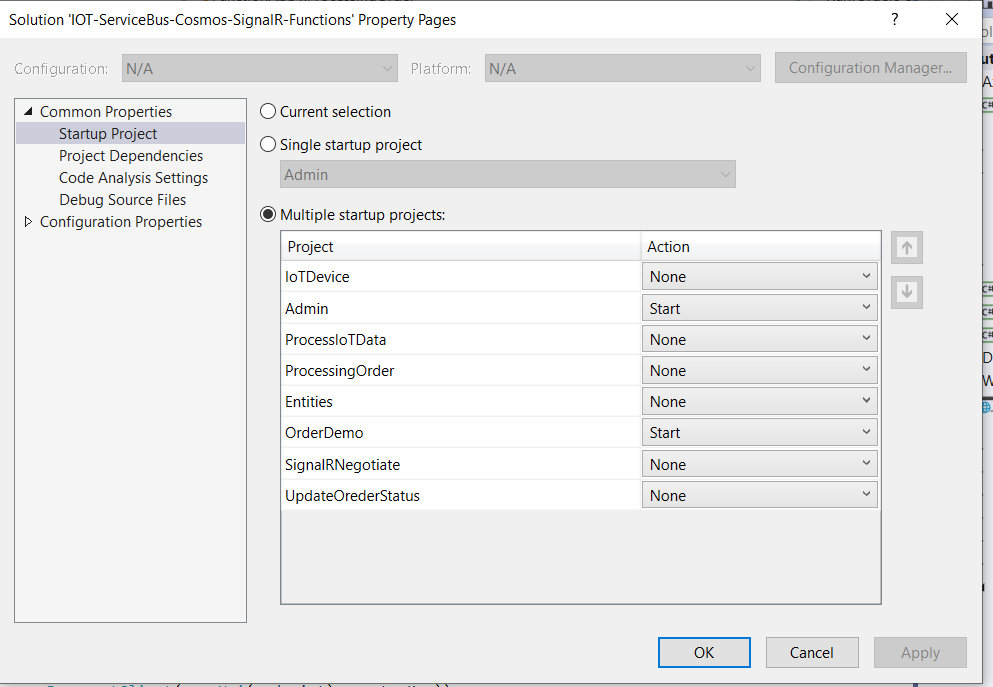


1. In SignalRNegotiate Function app enable and add below settings:

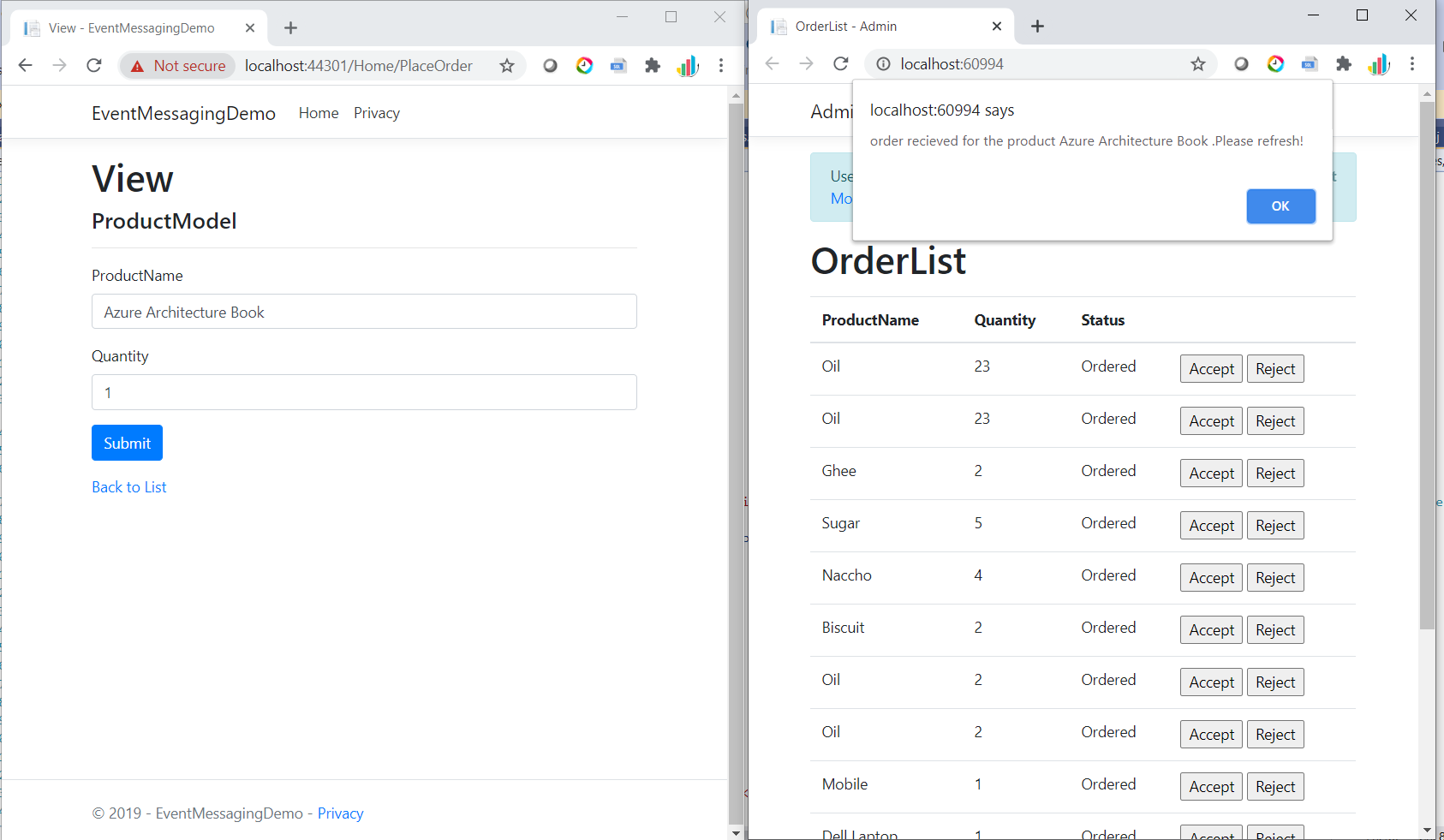
Currently our Admin and Client Panel both are running on localhost so we have added localhost URL for live applications we can change URL accordingly.



1. Set Admin & Order Demo as a Multiple Startup Project like below:



1. Run Application and add Product through Client Panel and magic happen you got notification through SignalR on Admin Panel.



**Path 2 (Admin Panel to Client Panel)** **Setup:**

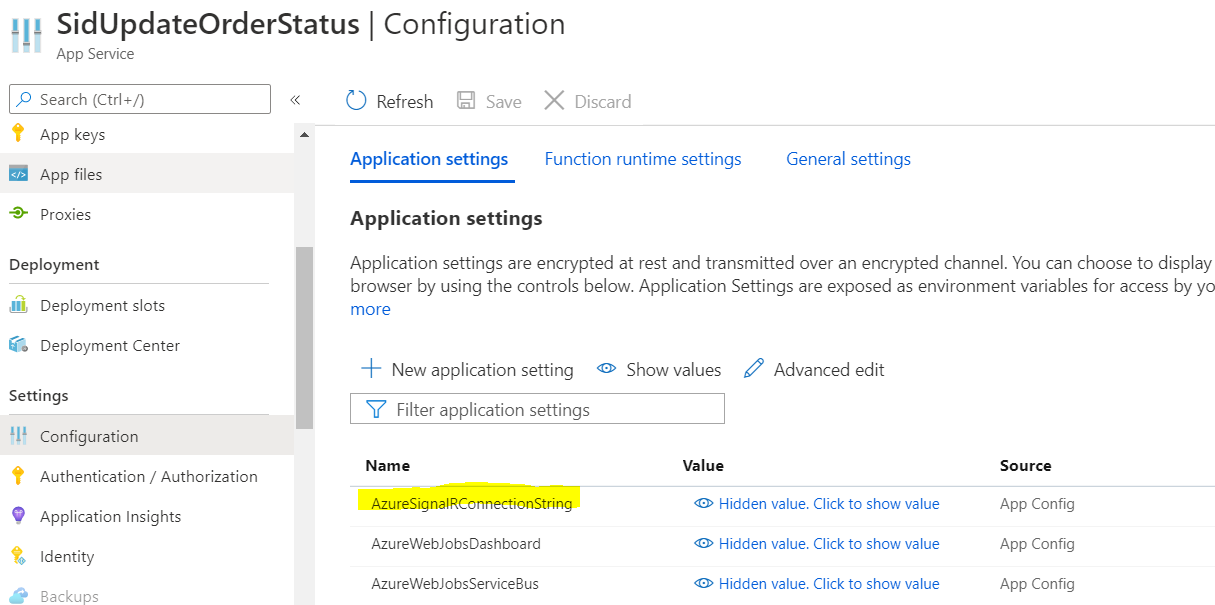
1. Open Admin=>Controller=>HomeController.cs

In ChangeOrderStatus function update endpoint and masterkey variable of your CosmosDB

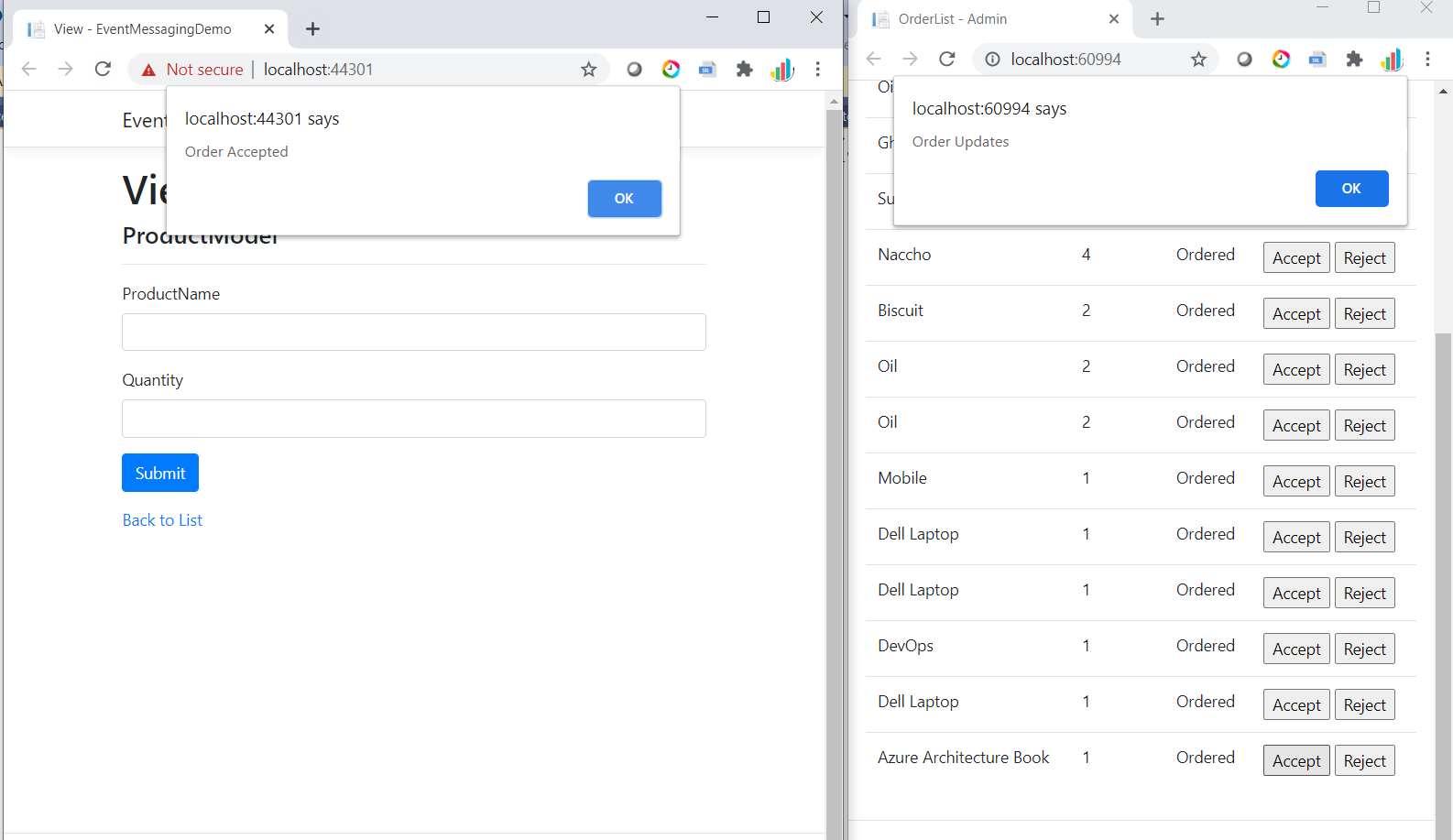
In UpdateServiceBus function add azure service bus connection string.

1. Publish UpdateOrderStatus Function App on Azure portal.

Add AzureSignalRConnectionString same as point 3.

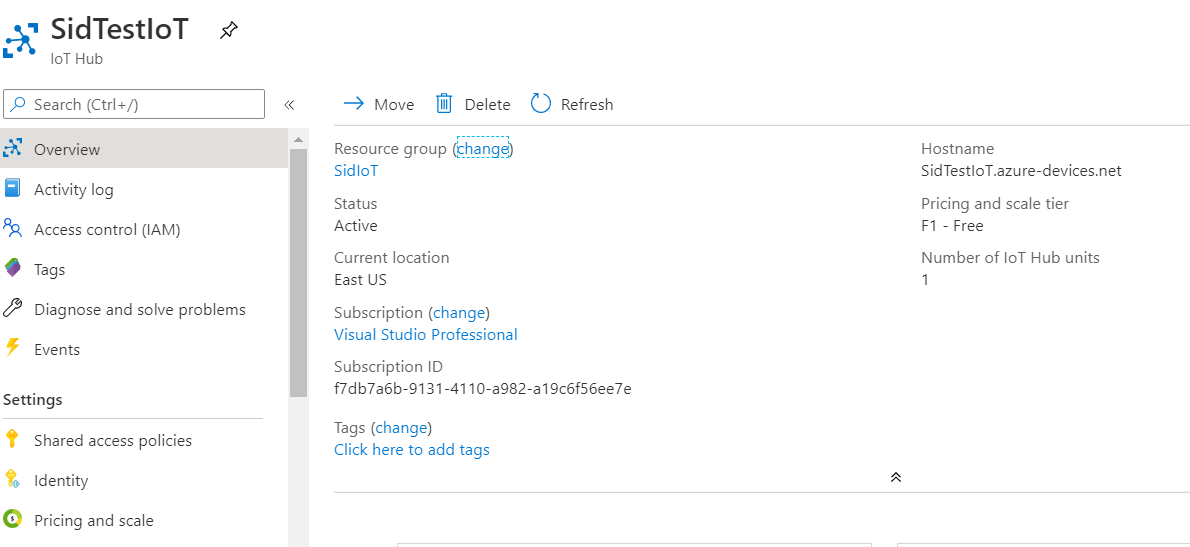


1. Run Admin & OrderDemo Application. On Admin panel Accept\Reject Order which notification through our process comes on OrderDemo application as per below.

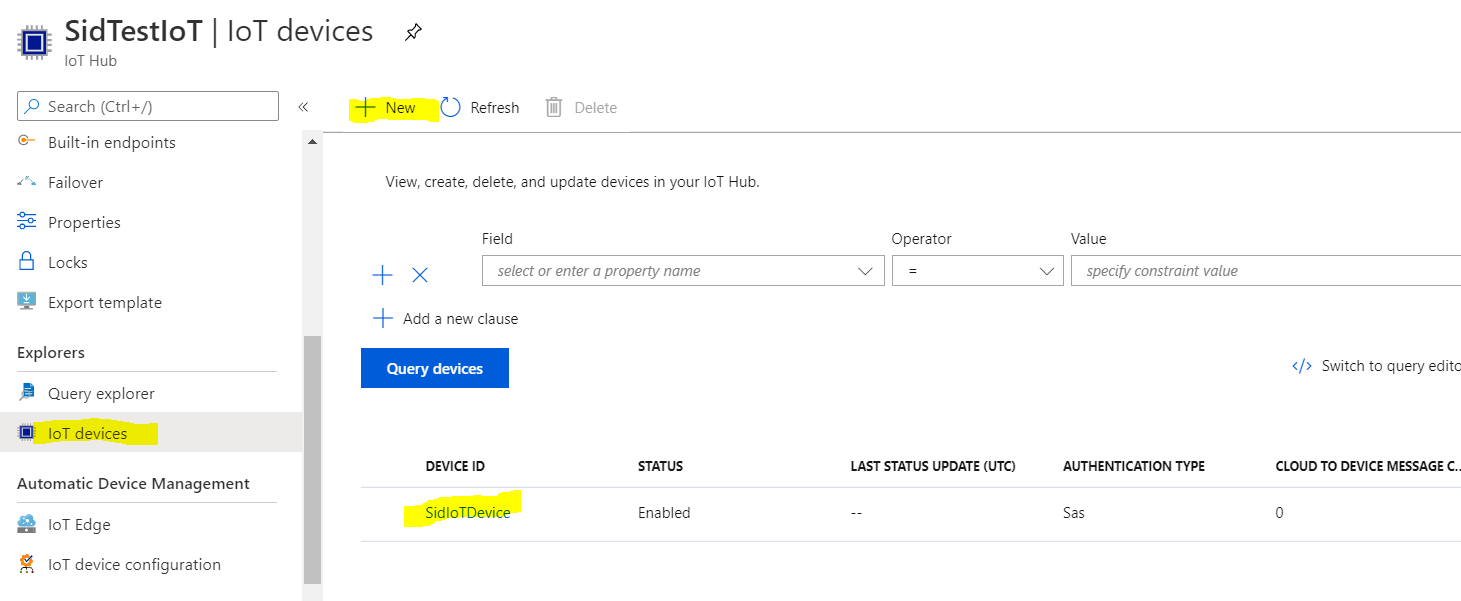


**Path 3 (IoT Device to Admin Panel) Setup:**

1. Create IoT Hub.



1. Add IoT Device in IoT hub component

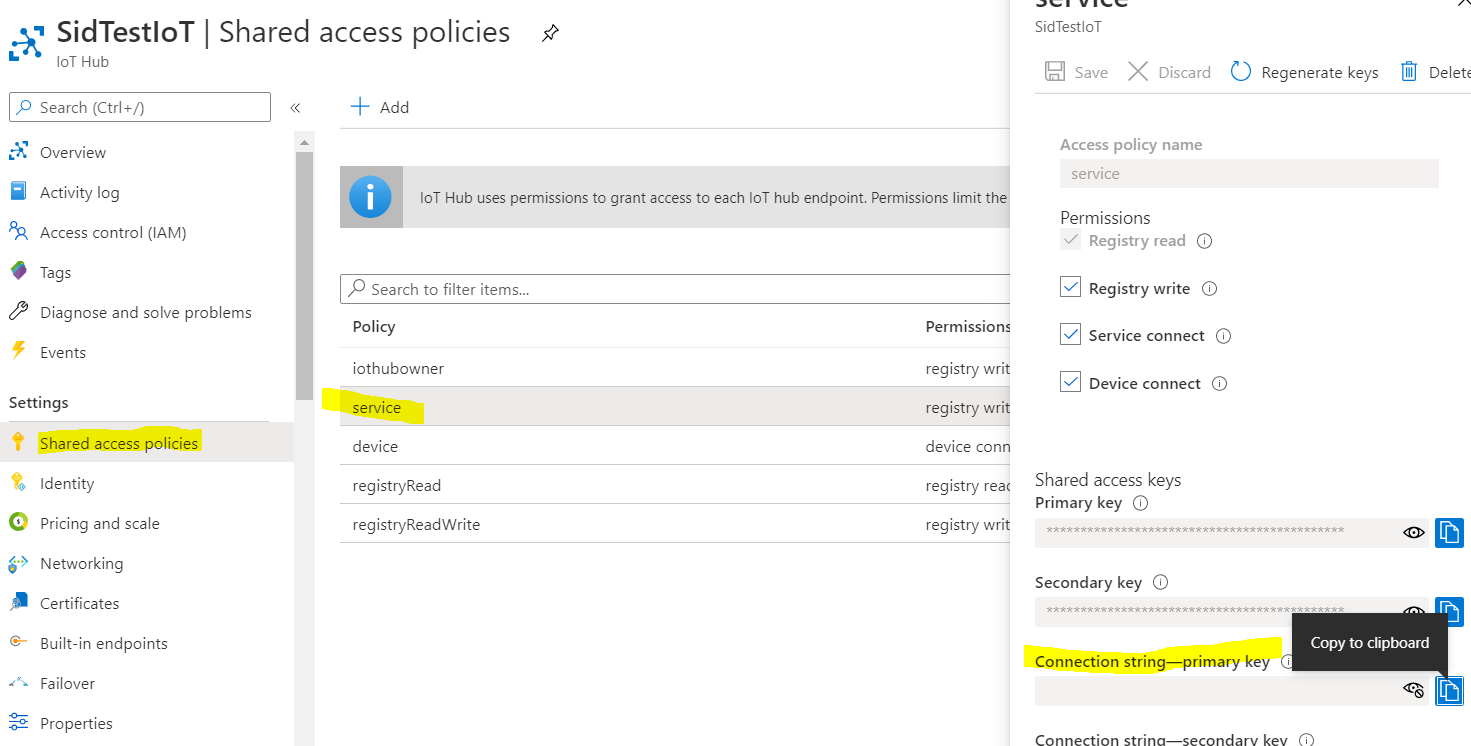


1. Open IoTDevice ConsoleApp => Program.cs

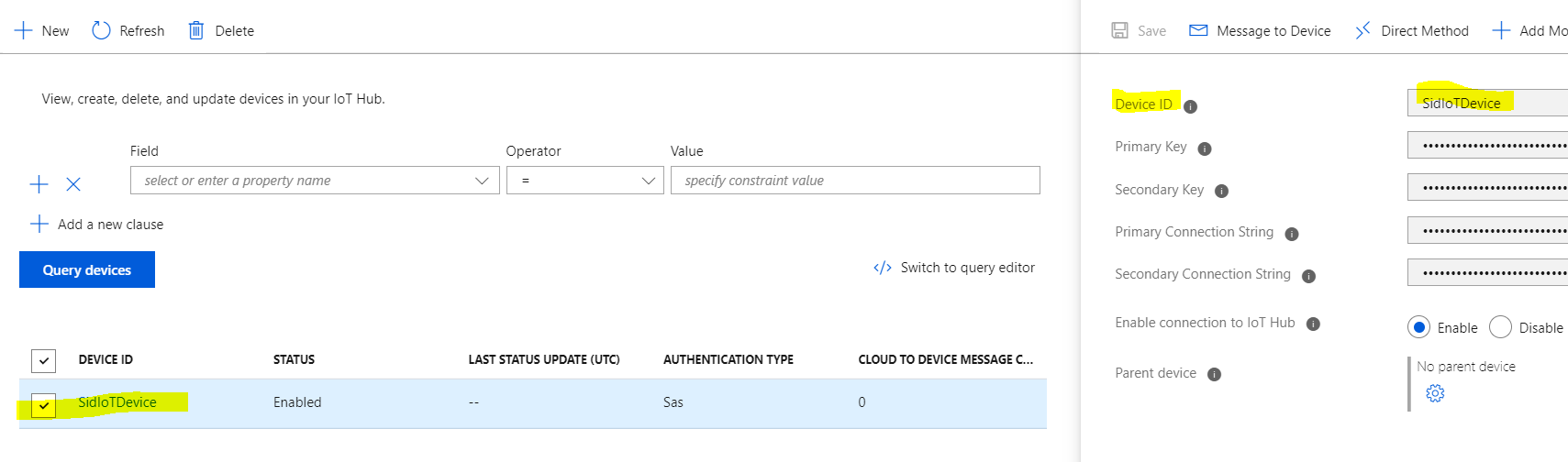
In Main function change connection as per below image from IoT Hub.

var conn = "HostName=SidTestIoT.azure-devices.net;SharedAccessKeyName=service;SharedAccessKey=VKOYPRPIAnmcAptRG2F/012KPvuE4LgddUS9kHT/2D8=;DeviceId=SidIoTDevice";

Note \* - Please add DeviceId in connection string at the end.



DeviceId:



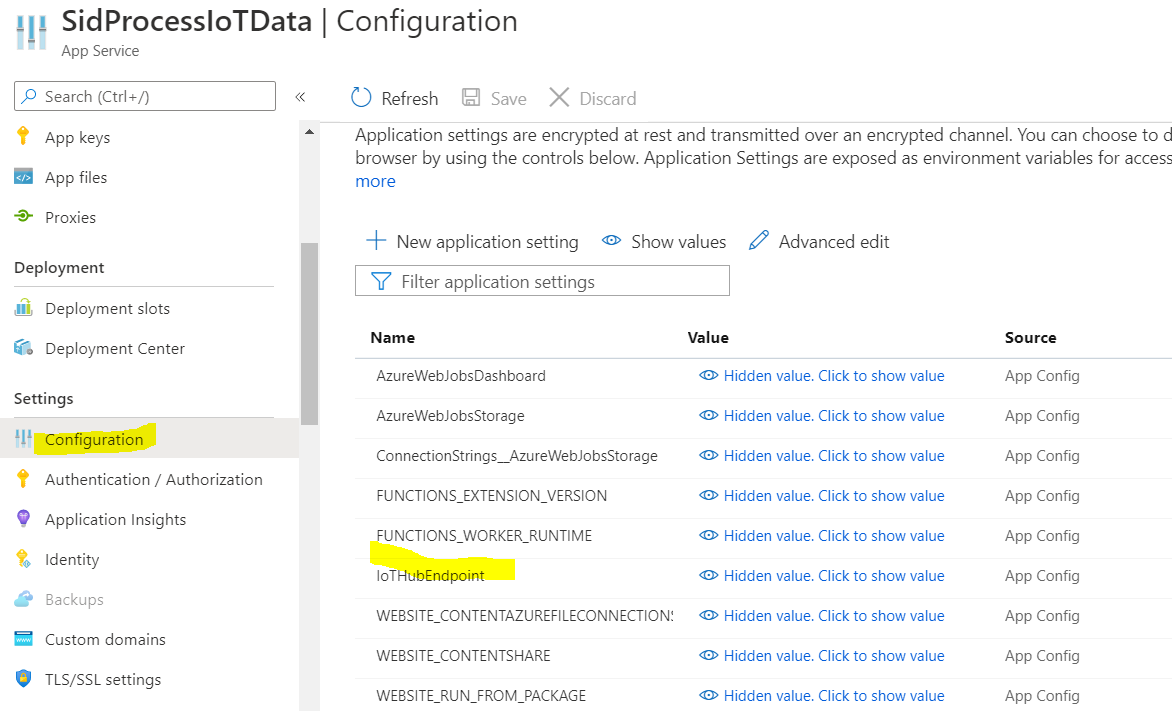
1. Open ProcessIoTData Function App =>ProcessIoTData.cs

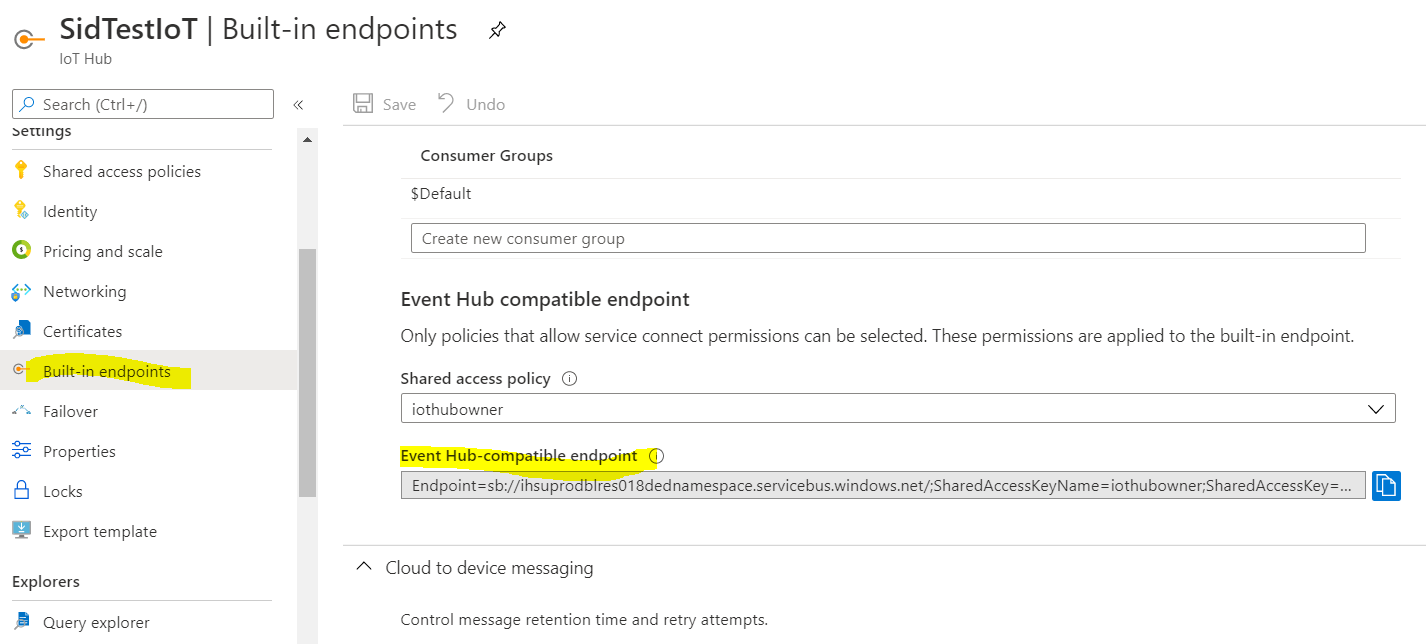
Update your Azure Service bus connection string.

1. Publish ProcessIotData Function App on Azure portal.

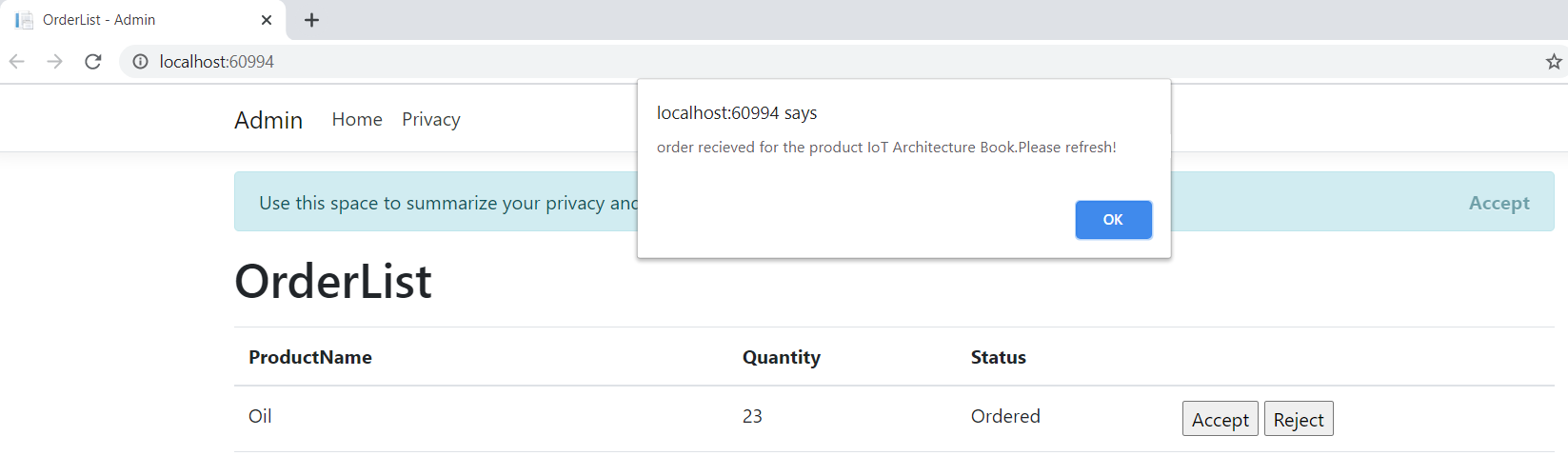
Add below Application settings in that.

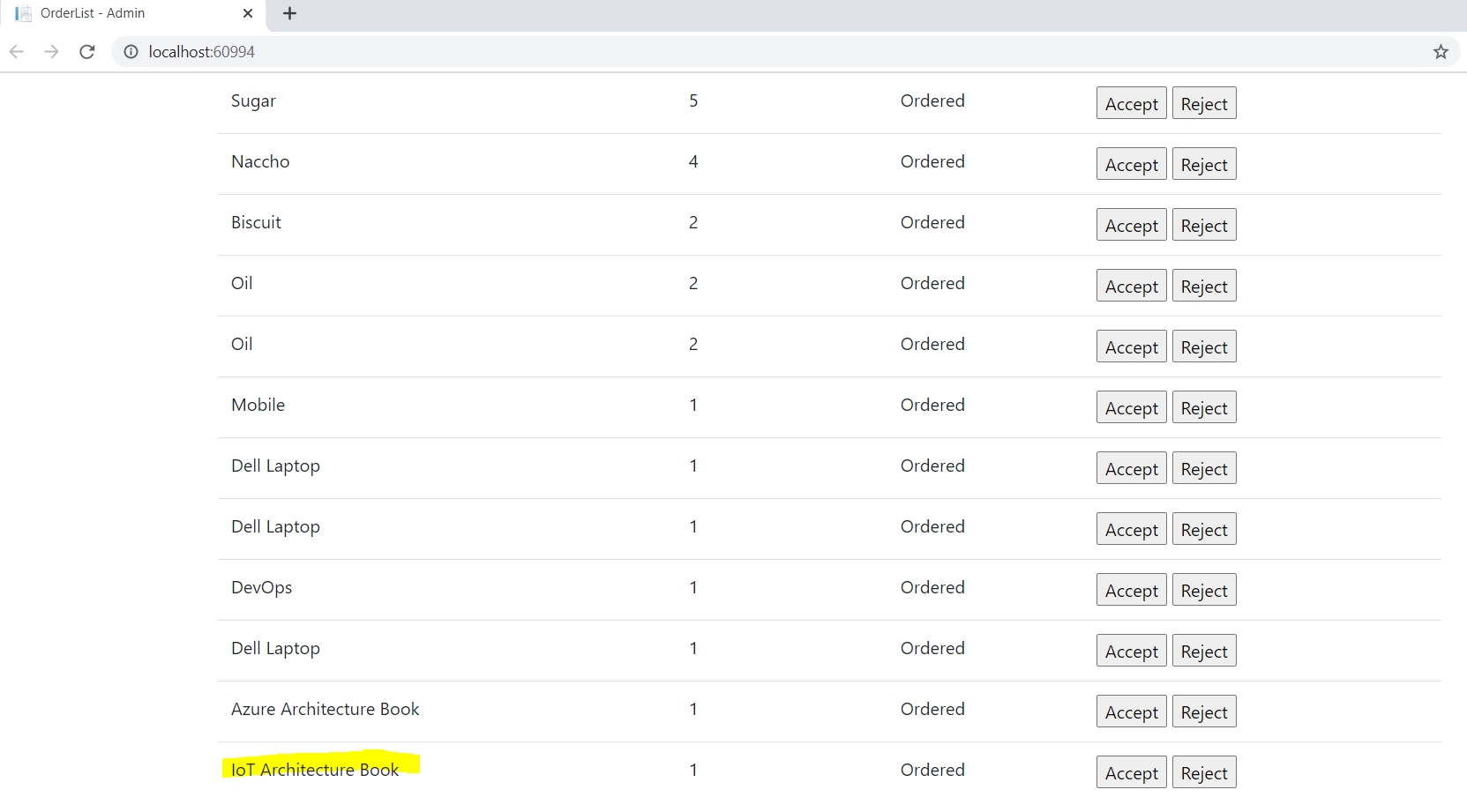
IoTHubEndPoint :





1. Run IoTDevice Console Application that placed order and data float through IoTHub to CosmosDb and AdminPanel.





Hope that article helps you to understand Azure components (IoT Hub, IoT Device Client, Azure WebApp, Azure Service Bus, Azure Function, Cosmos DB and SignalR) integration and configuration with working example.

You can get complete code here xxxx.

In Next Azure Lab we will discuss about other new and exciting Azure components.