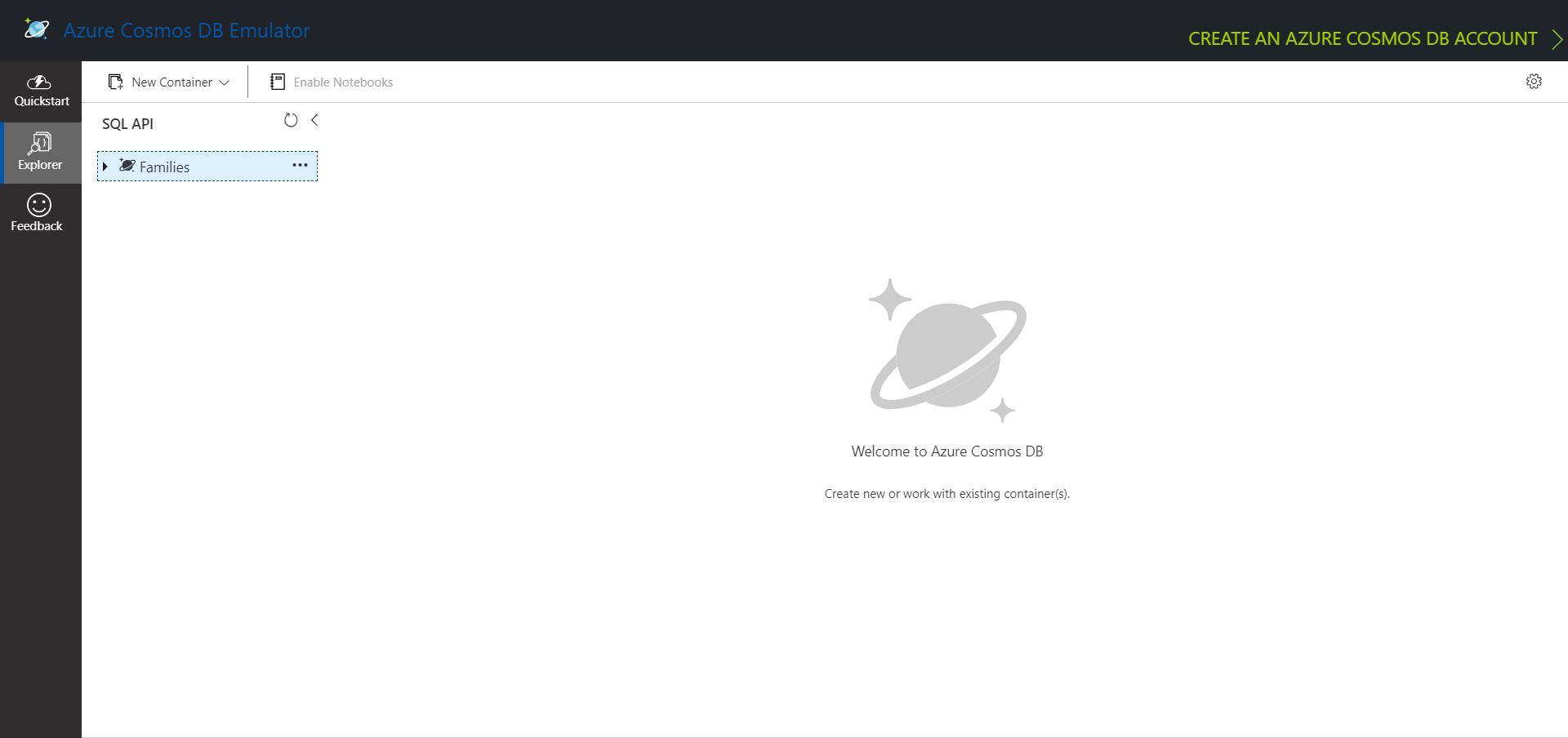
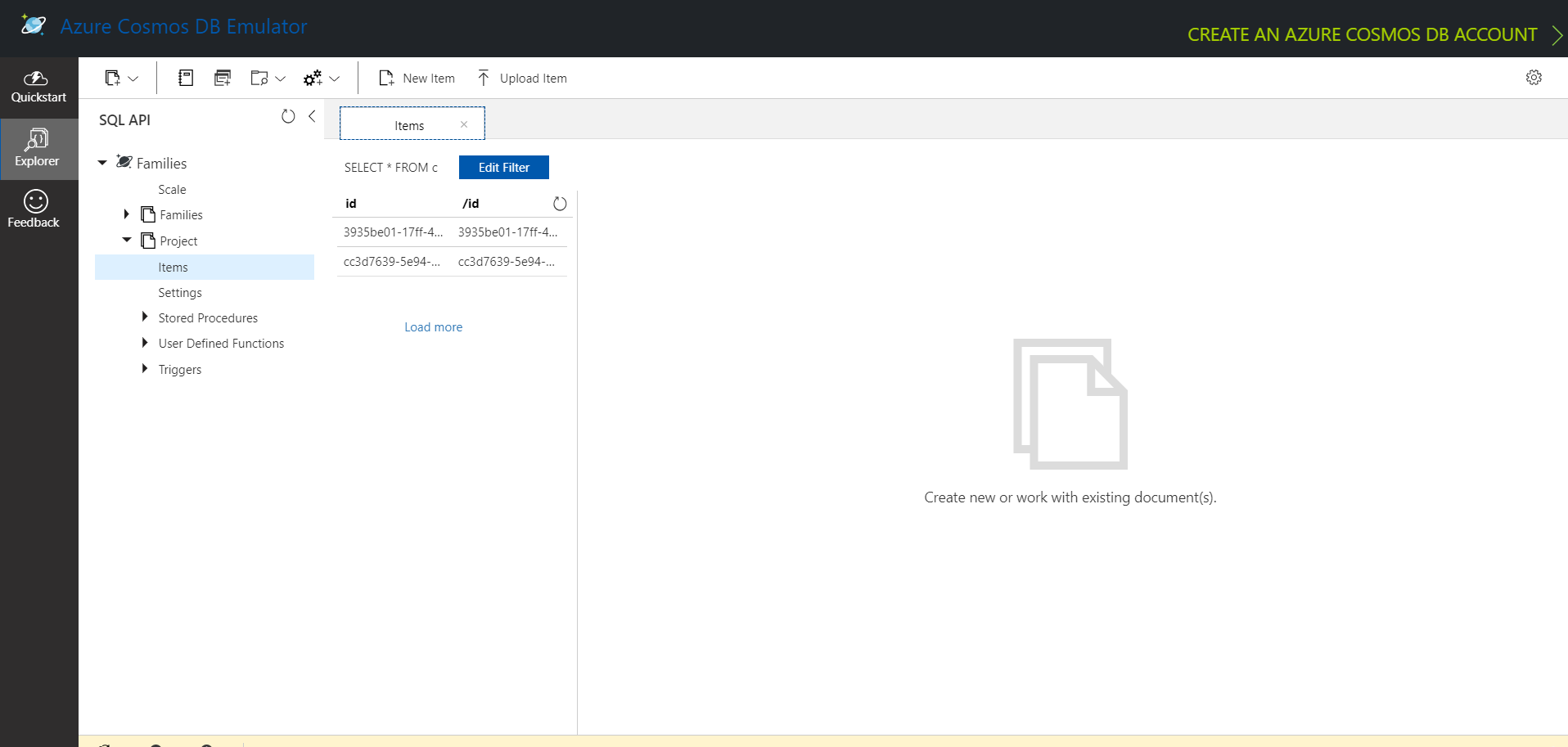
**Azure Cosmos Db Emulator**

**Database Setup:**

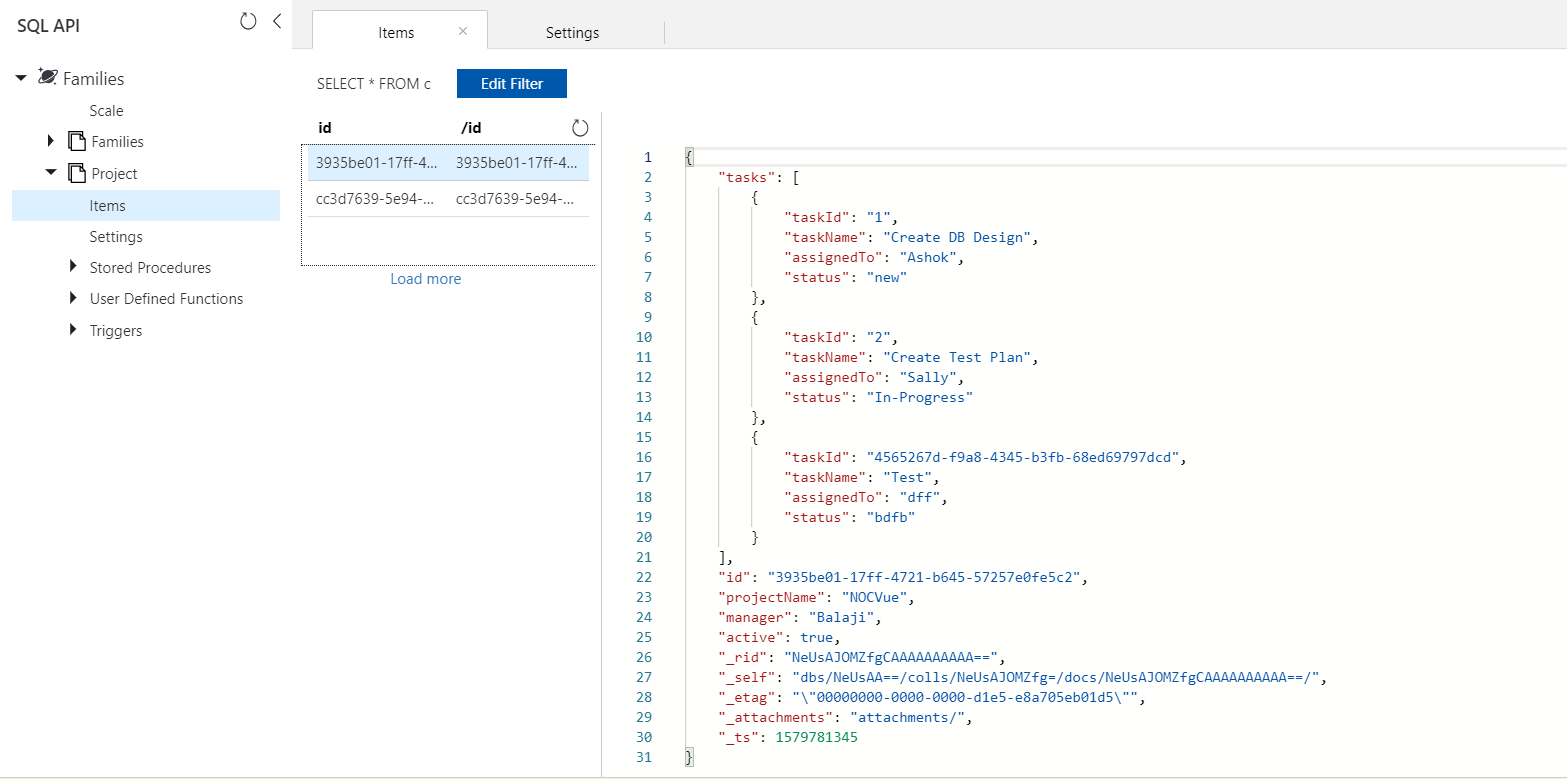
* Download and install Azure Cosmos DB Emulator.
* Start the Emulator, it will launch the below screen in browser.



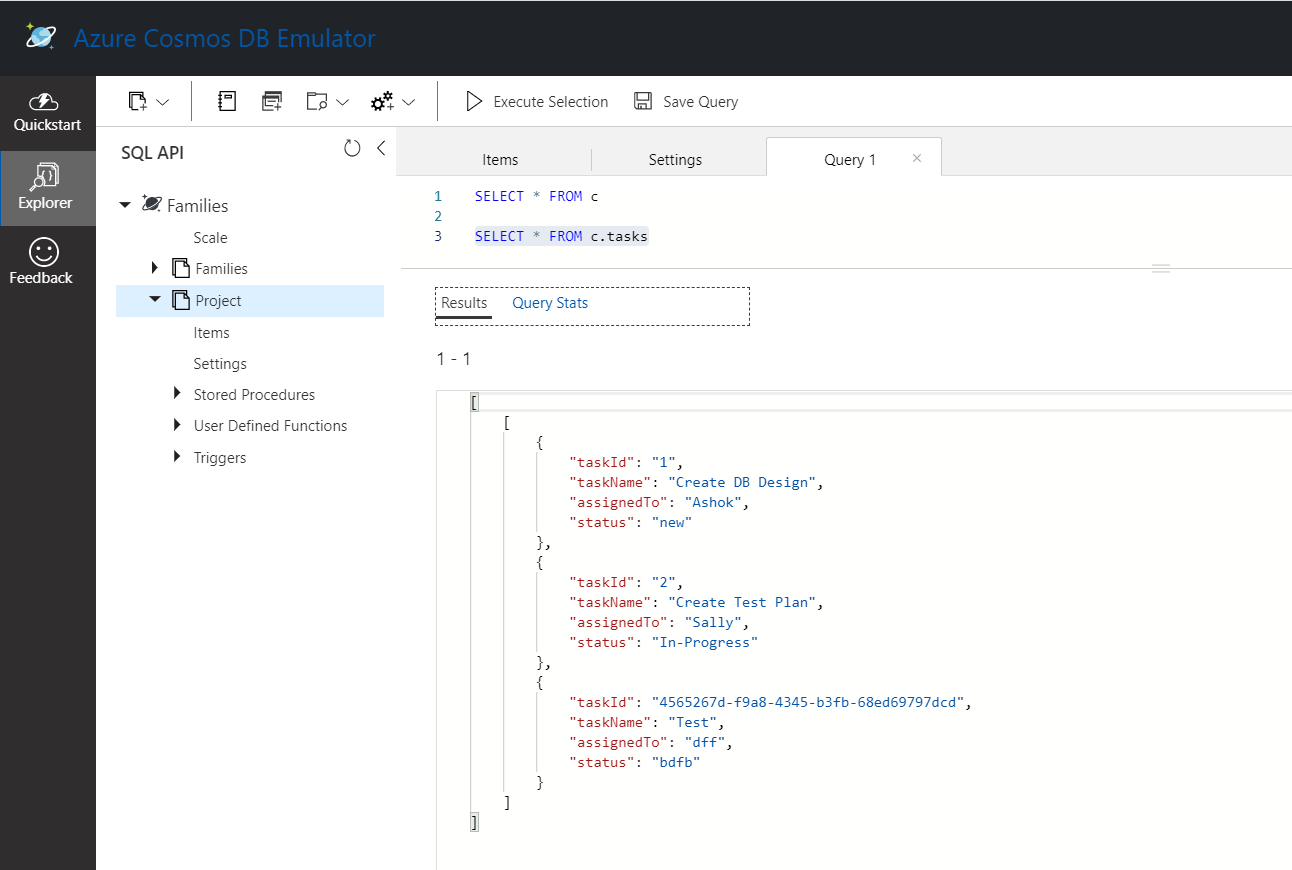
* Emulator supports only SQL and Mongo APIs, we will use SQL API for our technical spike.
* Create a Database in SQL API, in our spike it is **Families**.
* Create a Container[**Project**] inside Database as shown below,



* Add a Partition Key [**/Id** in our case] while creating the Container.
* Add sample data as shown below.



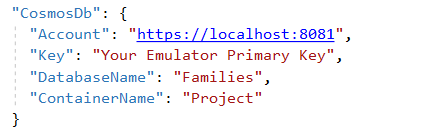
* Open a new SQL Query tab and we can write and execute our queries.



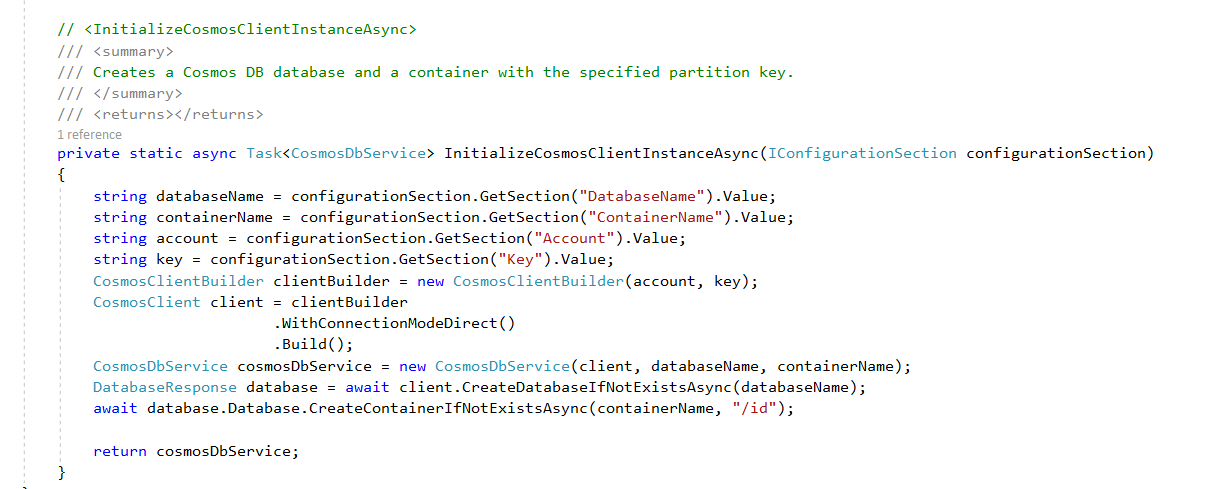
**Accessing Cosmos DB Data in .NetCore Application:**

**Library: Microsoft.Azure.Cosmos**

* Create a .NetCore application and add the above library using Nuget Package manager.
* Add the Cosmos Db Connection Configurations in appsettings.json file.



* Add the below code in startup.cs to get the DB & Container details if already there else to create the DB and container with the specified partition key

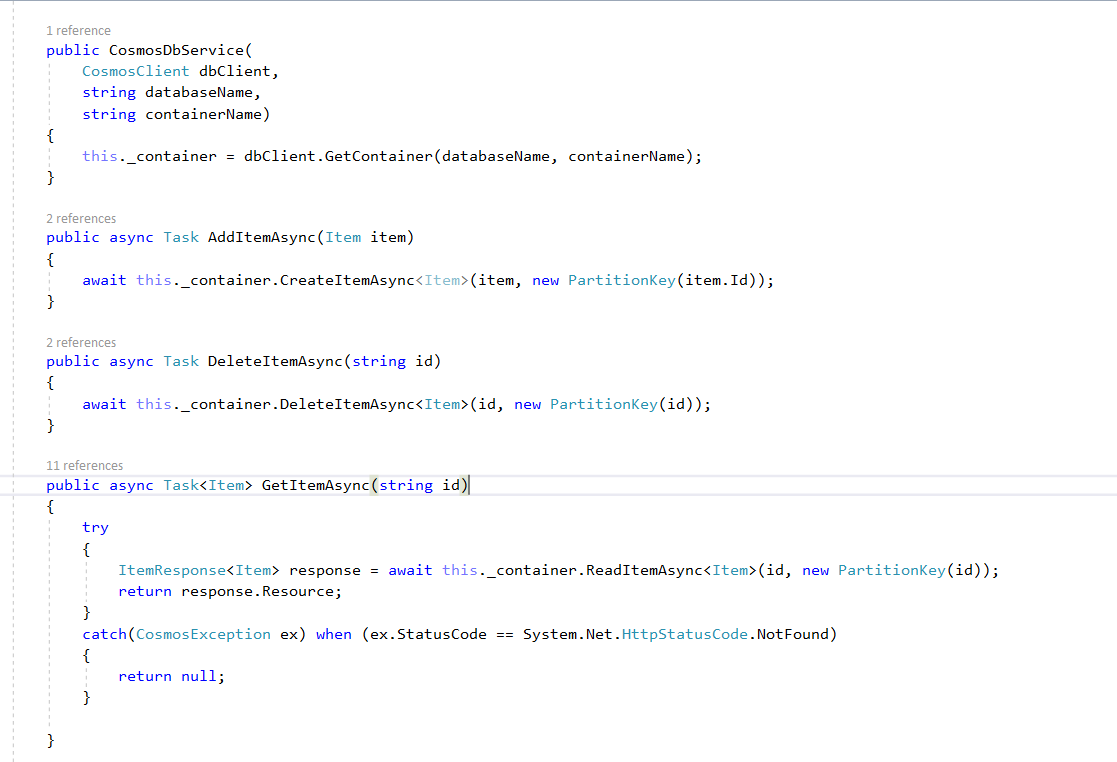


**Case:**

* Create an application to fetch all the project details from the DB.
* Add CRUD Operations for each Project.
* On click of a particular project, fetch all the tasks of that project.
* Add CRUD Operations for each Task.

**Code Usage:**

* Add Models, View & Controller classes to access and represent the **Project** Container details from Cosmos DB
* Create a DataLayer service Class to add the logic to create access and update the Project and Task Details.





**Note:** ORM used – **CosmosClient** from Library **Microsoft.Azure.Cosmos**

* Build and Launch the application.

