**Getting started with Cosmos DB**

Azure Cosmos DB is a globally distributed multi-model database. One of the supported APIs is the Graph (Gremlin) API, which provides a graph data model with [Gremlin query/traversals](https://tinkerpop.apache.org/gremlin.html). In this Hands on Lab you will learn how to use the Azure Cosmos DB with the Graph API to store and access data from a Java application.

**Prerequisite:**

Below are the prerequisite to get started with the below Hands on Lab (HoL). Please ensure you machine is having all the required software, If not download from the given links.

Eclipse : download from <https://www.eclipse.org/downloads/> Version is Oxygen

Jdk 1.8 : download from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Github client: https://git-for-windows.github.io/

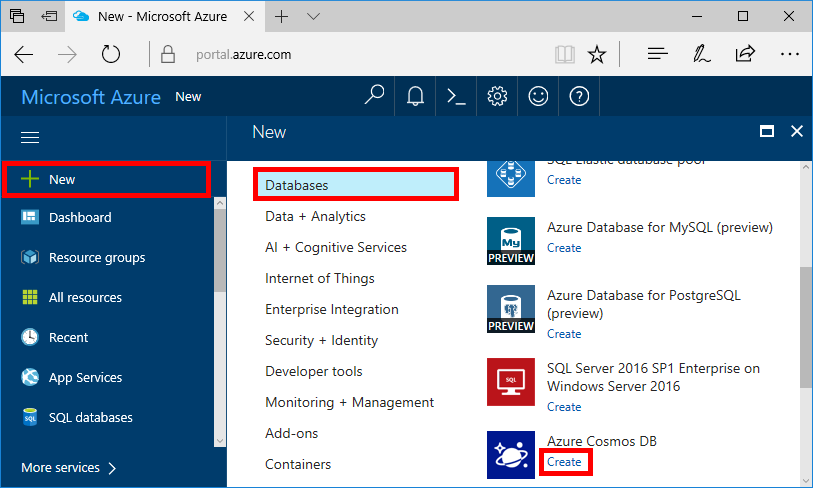
Maven : <https://maven.apache.org/download.cgi>

Azure subscription

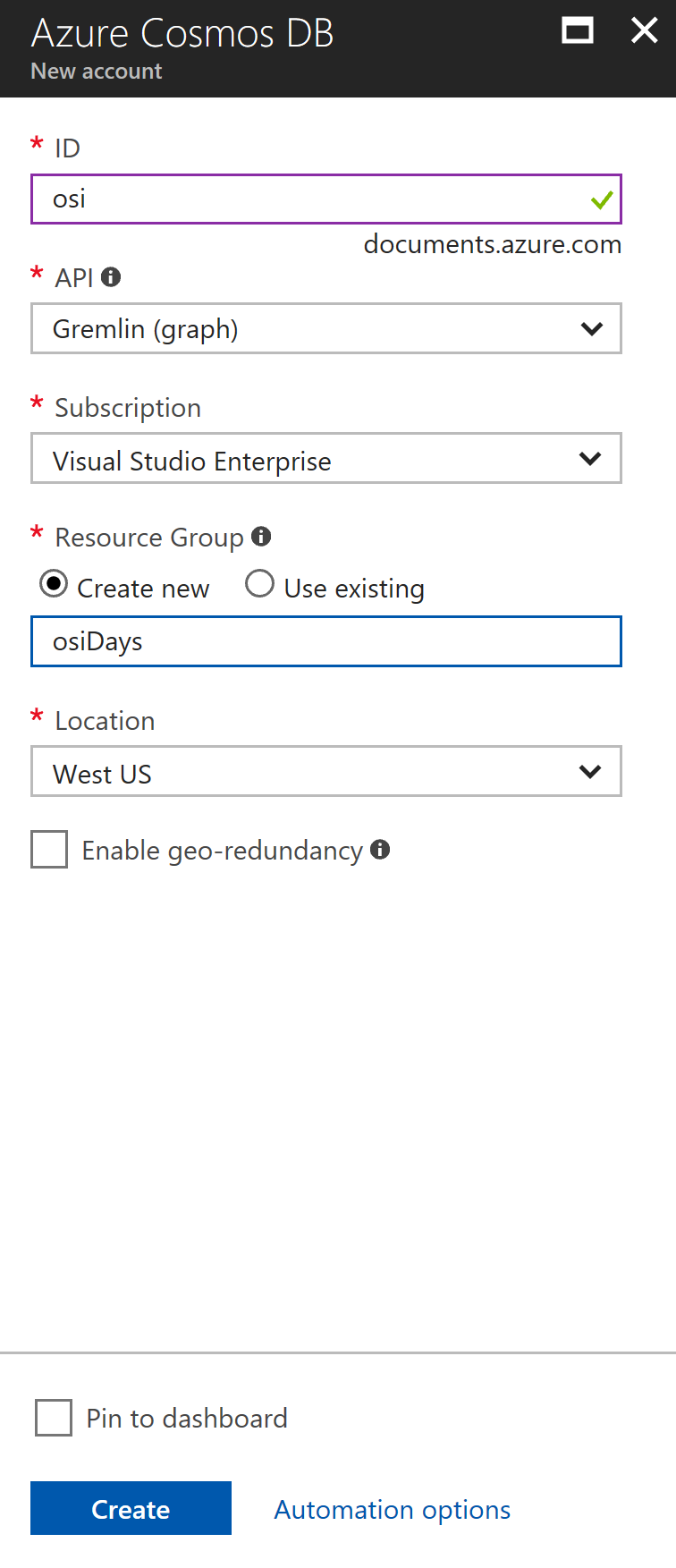
**Lets Start**

Before you can create a graph database, you need to create a Gremlin (Graph) database account with Azure Cosmos DB.

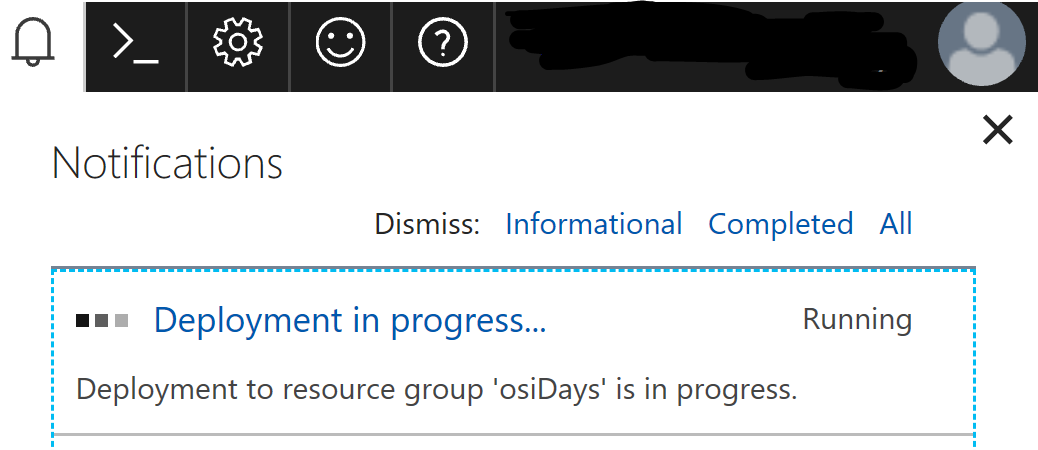
1. In a new window, sign in to the [Azure portal](https://portal.azure.com/).
2. In the left pane, select **New** > **Databases** > **Azure Cosmos DB** > **Create**.



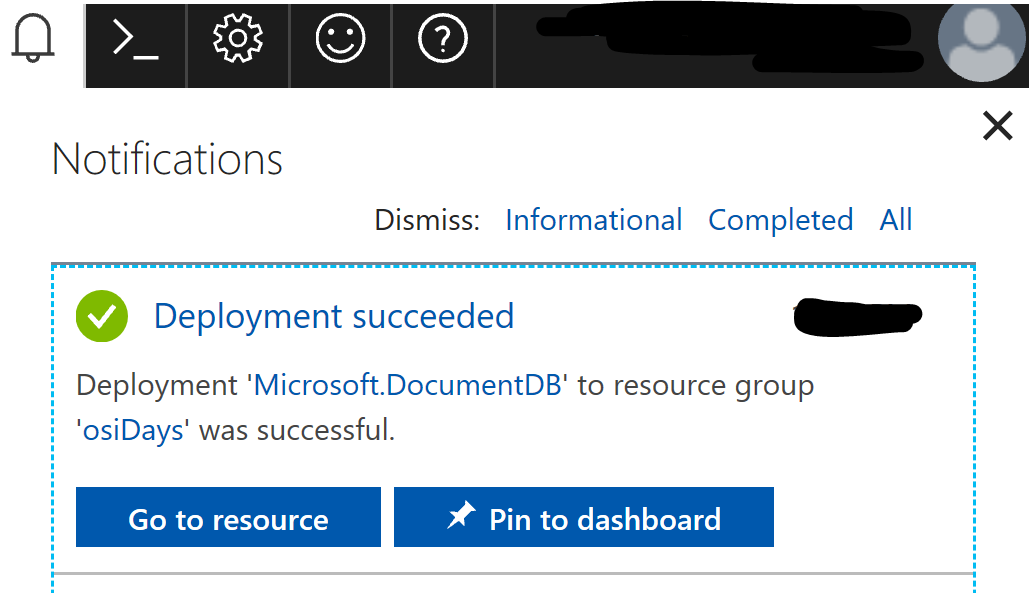
1. Under **New account**, specify the configuration that you want for this Azure Cosmos DB account. You need to specify a **unique** **ID** for your database, as in below screenshot we have given osi.



1. Select **Create** to create the account.
2. On the toolbar, select the **Notifications** icon to monitor the deployment process.



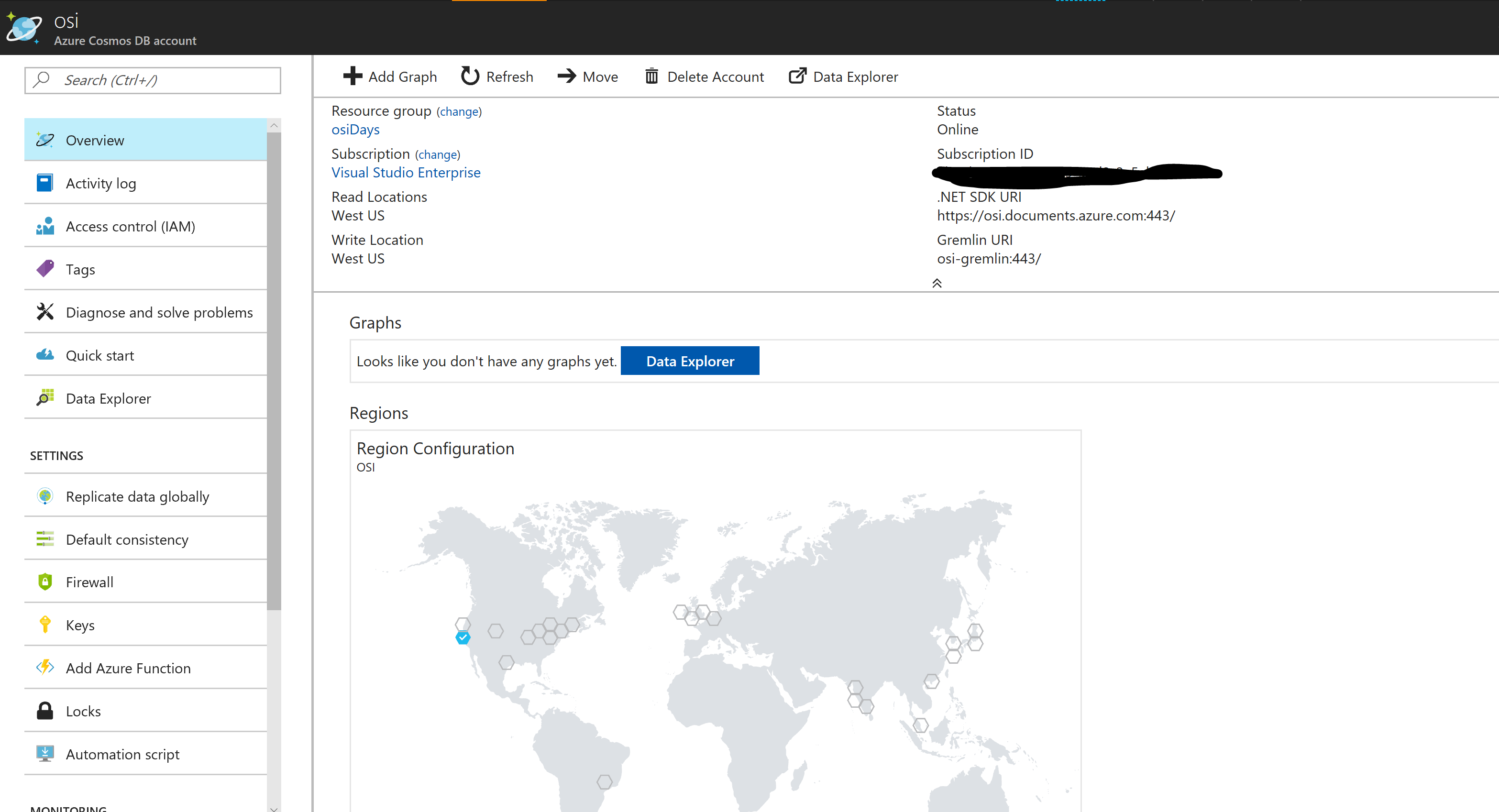
1. When the **Notifications** window indicates the deployment succeeded, close the window. Open the new account from the **All resources** tile on the **Dashboard** or directly from **Go to resource** button in notification as shown below



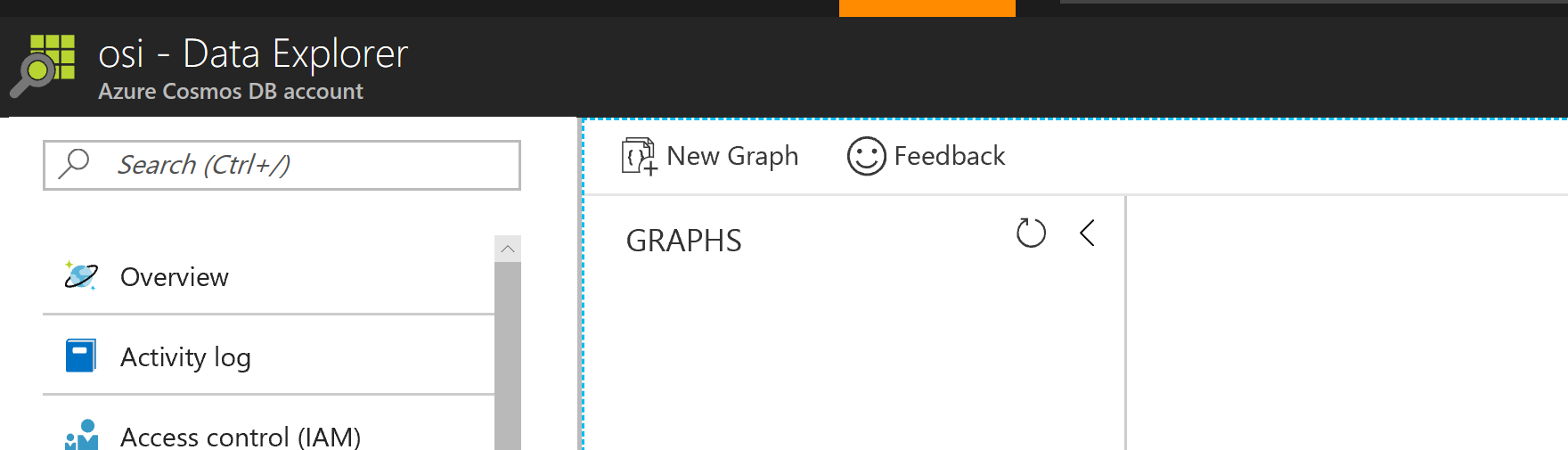
Add a graph

You can now use the Data Explorer tool in the Azure portal to create a graph database.

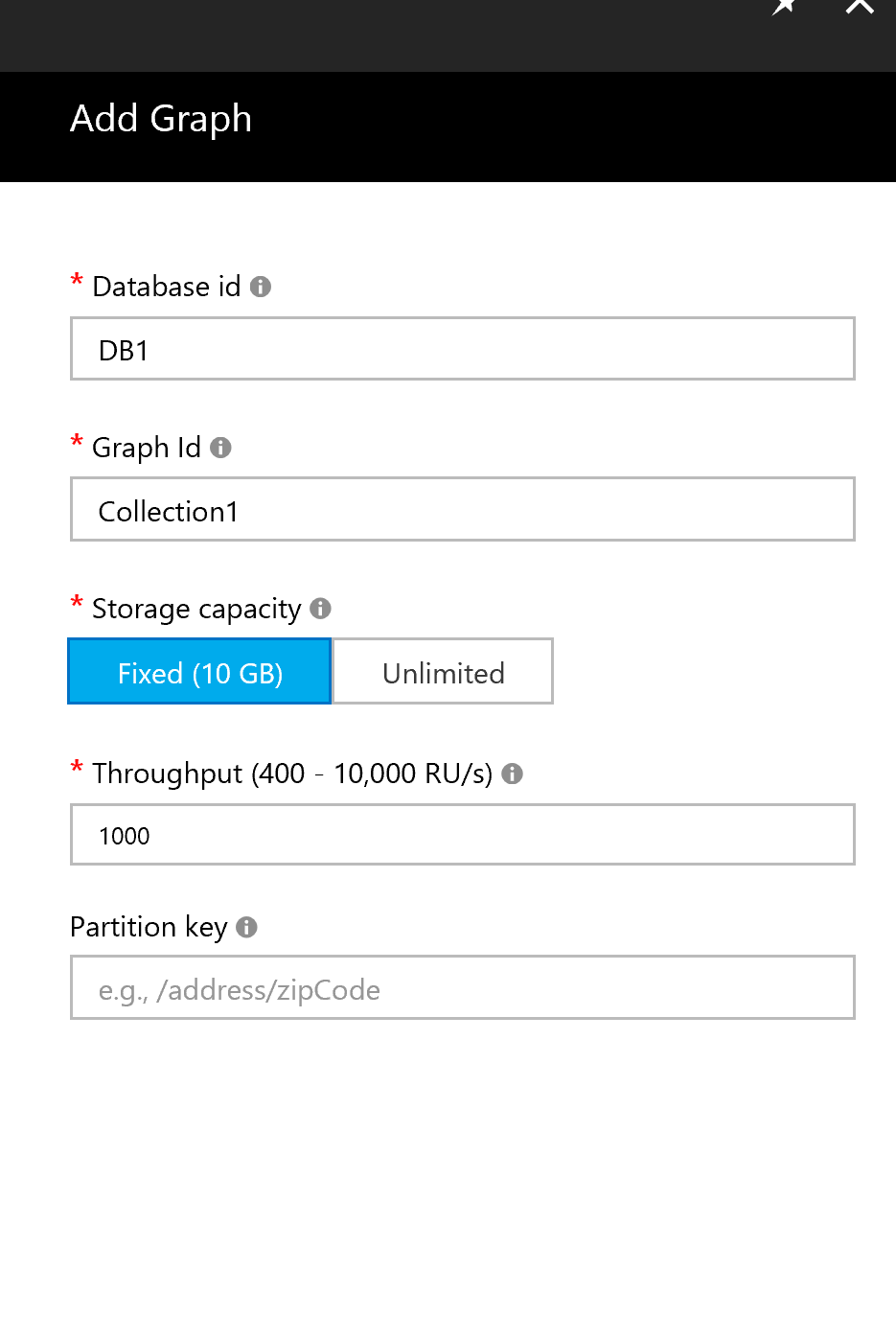
1. Open your **Azure Cosmos DB account** like osi in this case



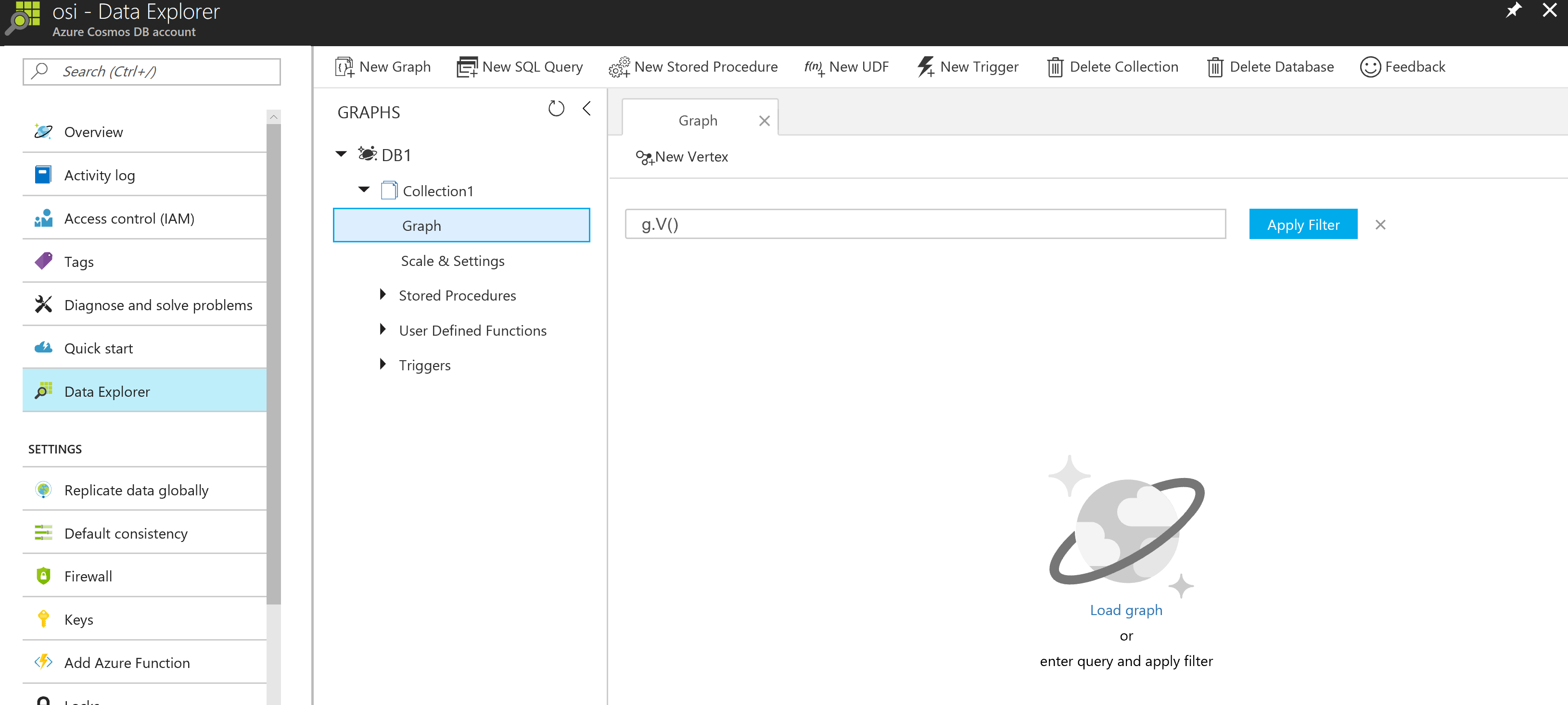
1. Click on **Data Explorer** button
2. Click on **New Graph**



1. Enter all **database ID , Graph ID** and click on **OK**



1. It will create a blank graph database as shown below.



## Clone the sample application

Now let's clone a graph app from github, set the connection string, and run it. You see how easy it is to work with data programmatically. +

1. Open a git terminal window, such as git bash, and cd to a working directory.
2. Run the following command to clone the sample repository.

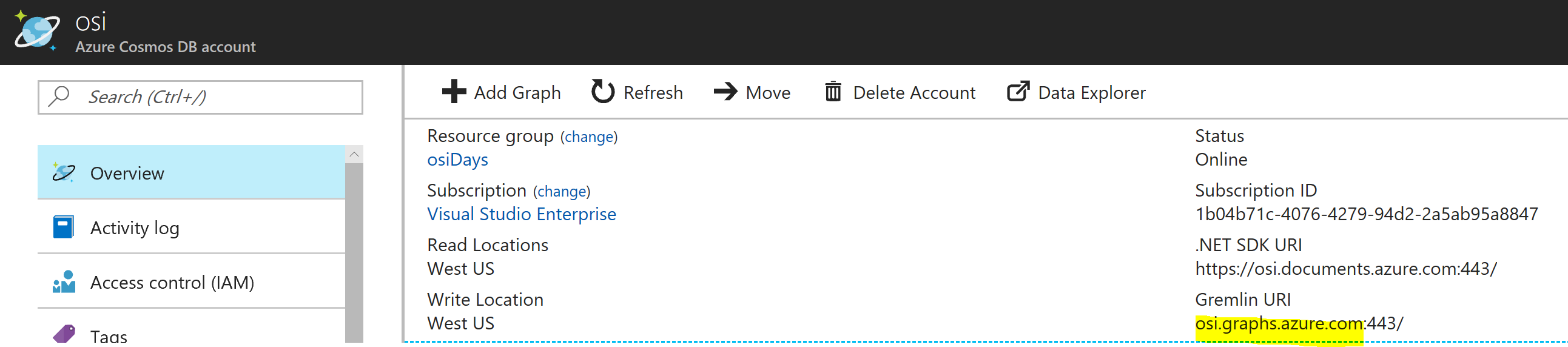
git clone https://github.com/rawatsudhir1/OSI2017.git

1. Your source code exist under ACDB folder.

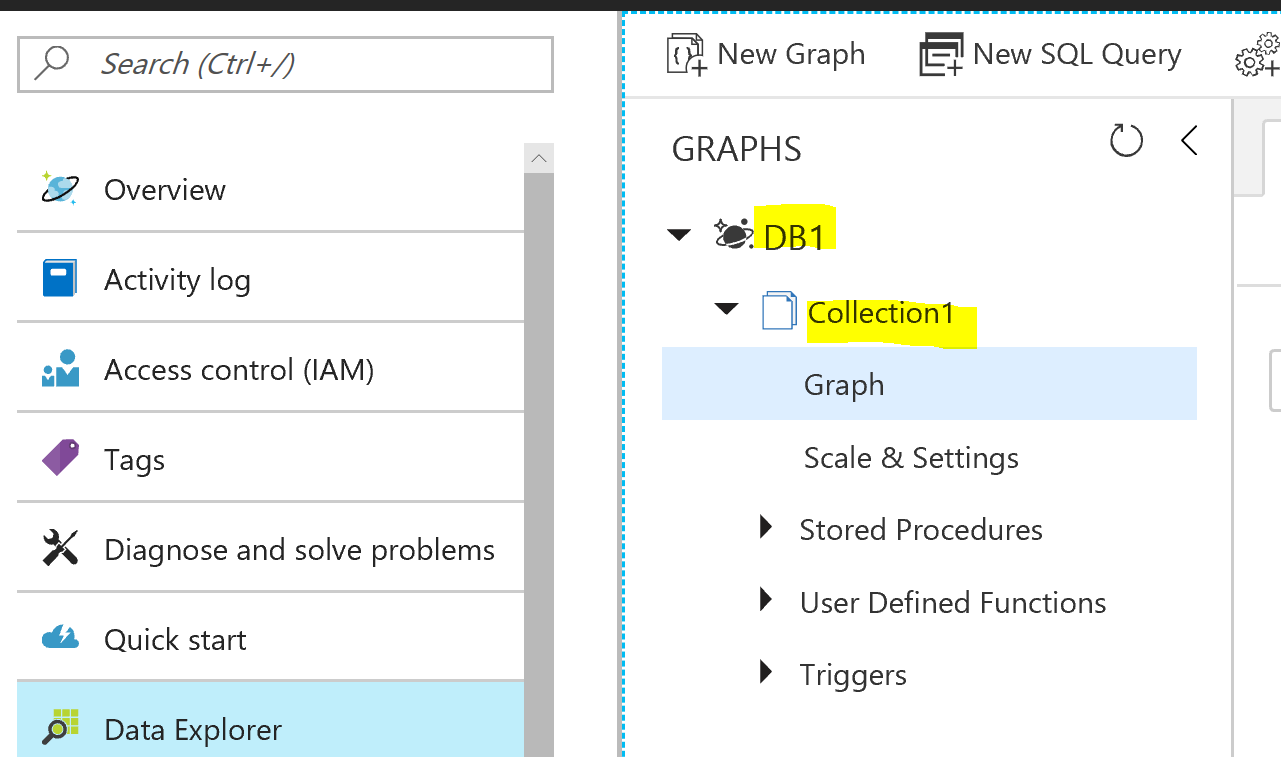
## Update your connection string

1. Open the src/remote.yaml file.
2. Fill in your hosts, username, and password values in the src/remote.yaml file. The rest of the settings do not need to be changed.

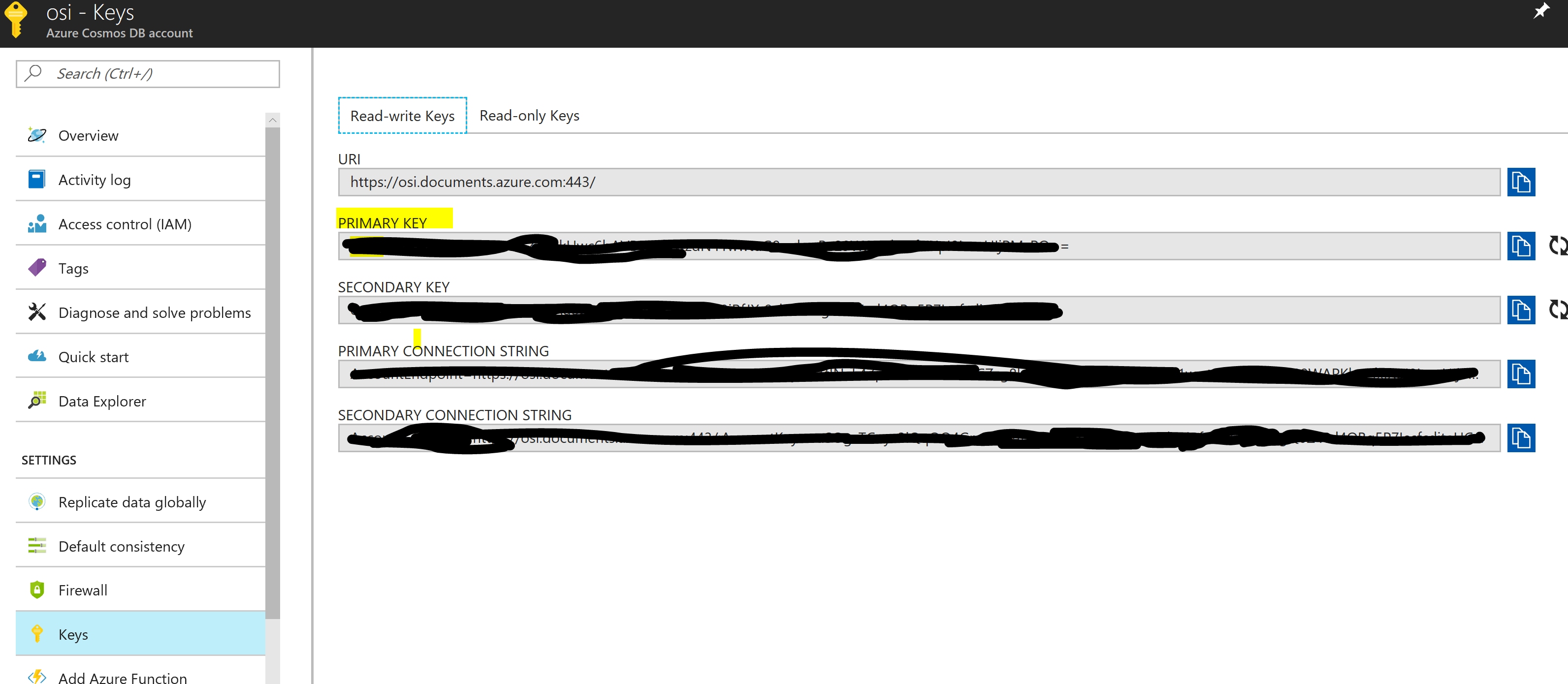
**Value of Hosts:**



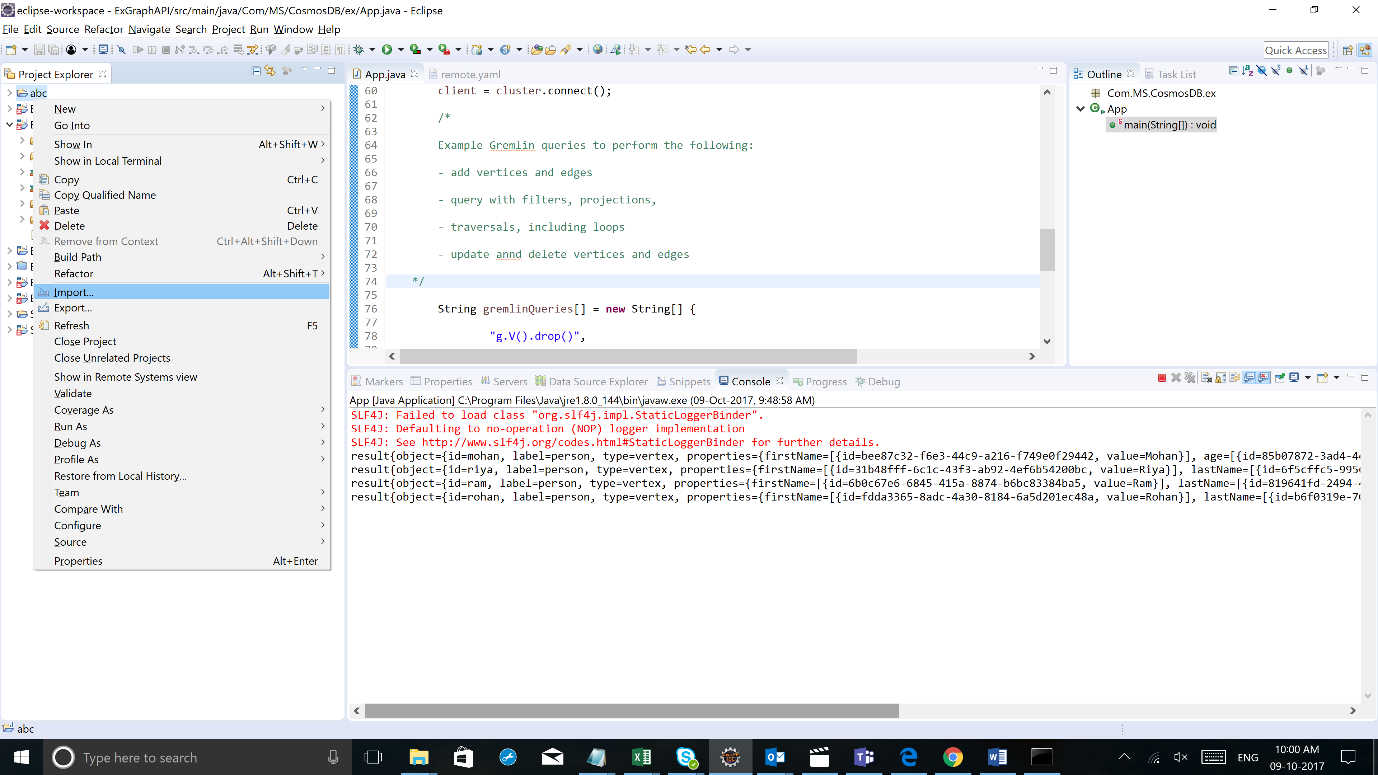
**Value of Username:** We will use **database ID** and **Graph ID** value that we had used during creation of graph DB. That you can get by clicking on **Data Explorer.** SO our final user name will be /dbs/<**database ID**>/colls/ **<Graph ID>.** In this use case it will be /dbs/DB1/colls/Collection1. Please update based of your configuration.

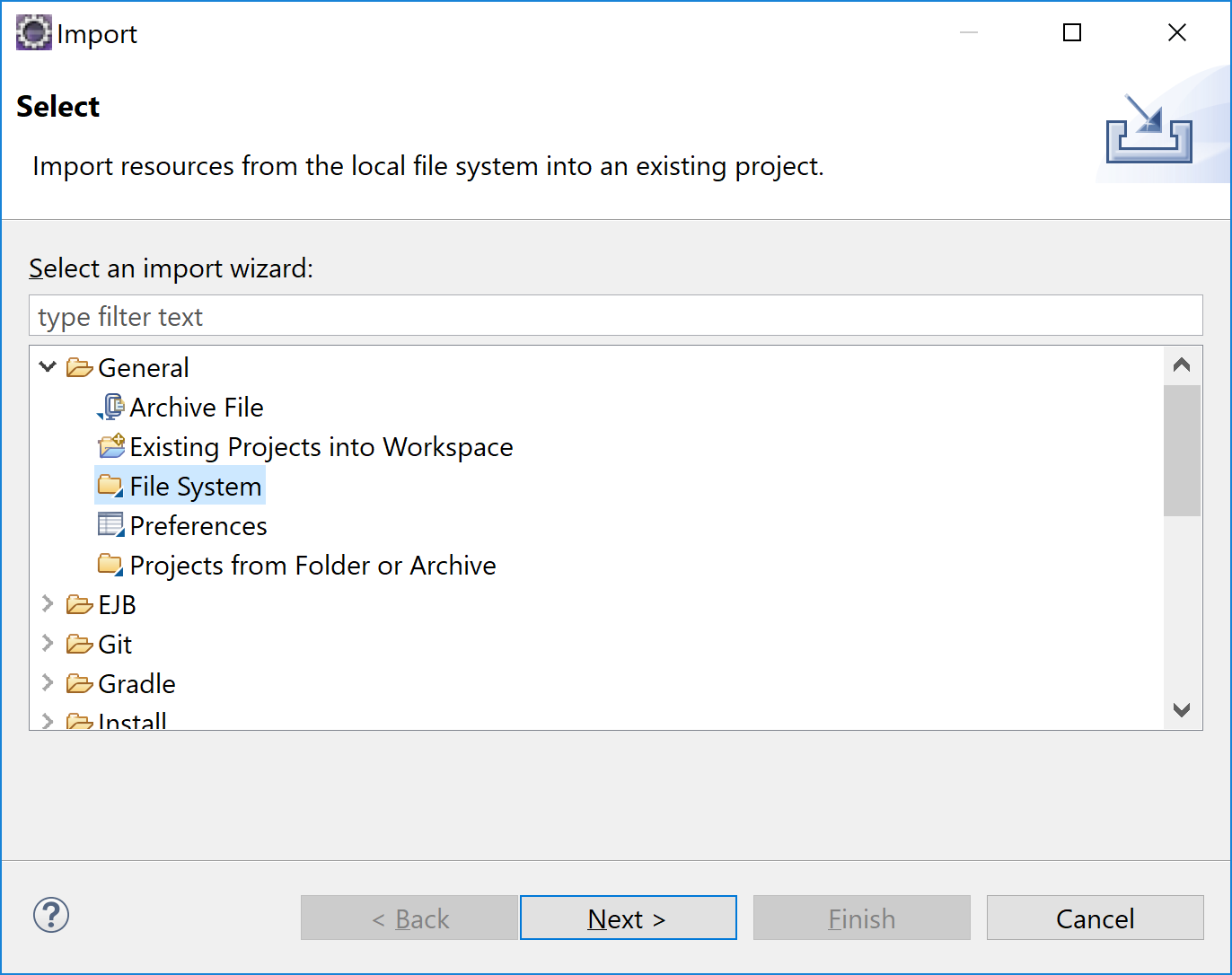


Value of password: Go to **Keys** and copy **Primary key**

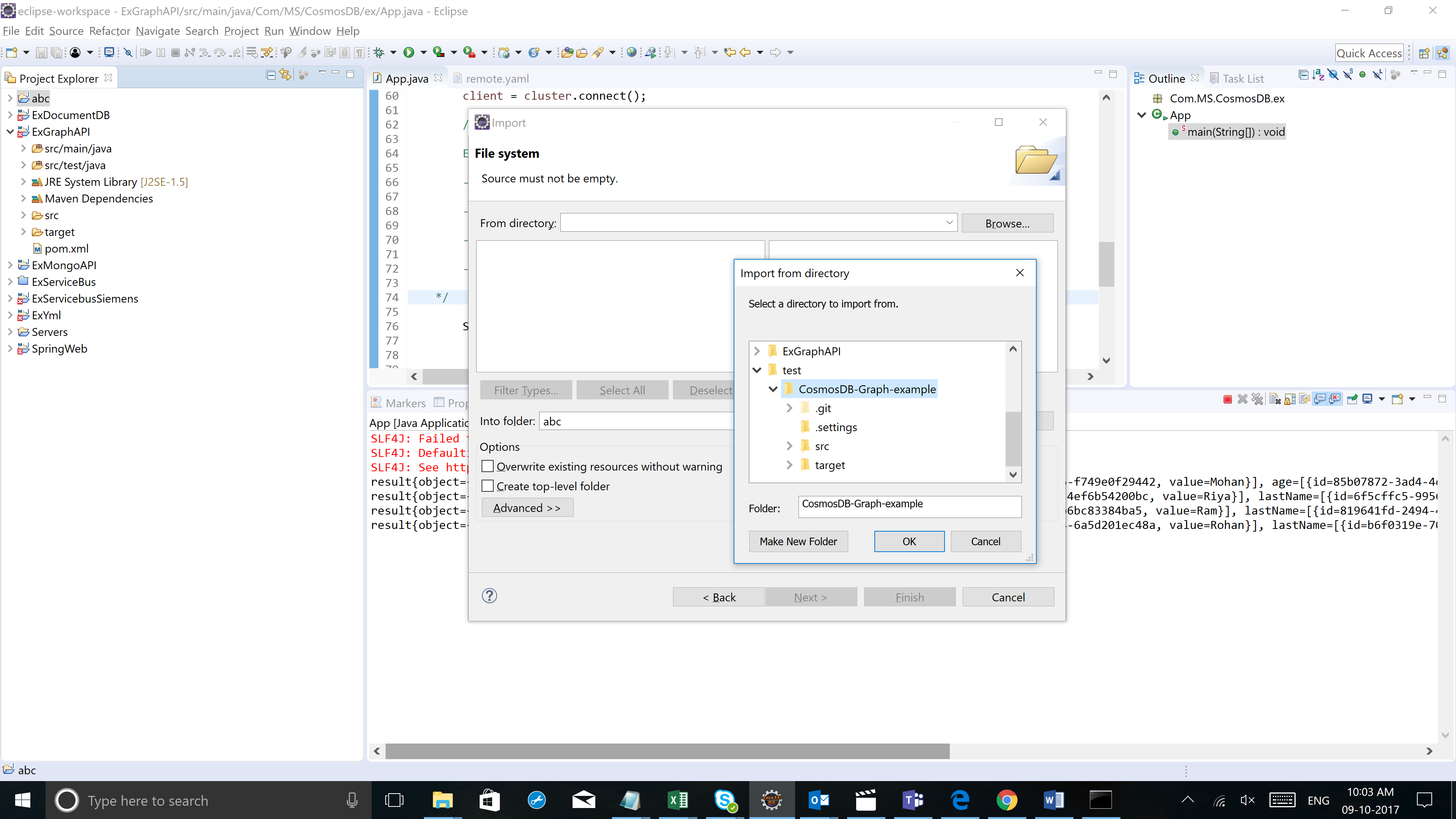


## Run your code

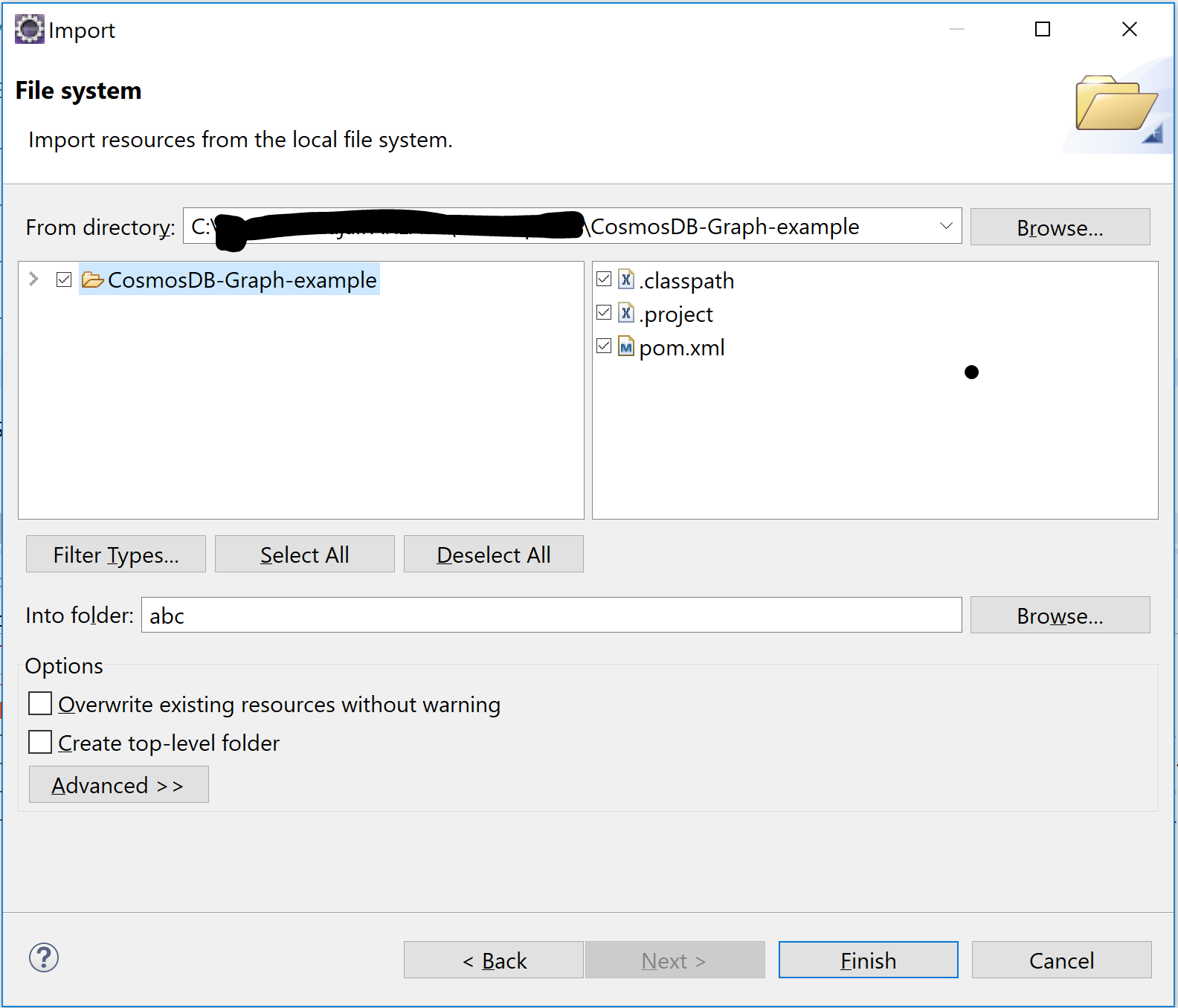
* 1. Create a new **java project** in Eclipse.
  2. Import code from where you did Git clone.
     1. Right click on project click **Import**
     2. Select **File Folder**



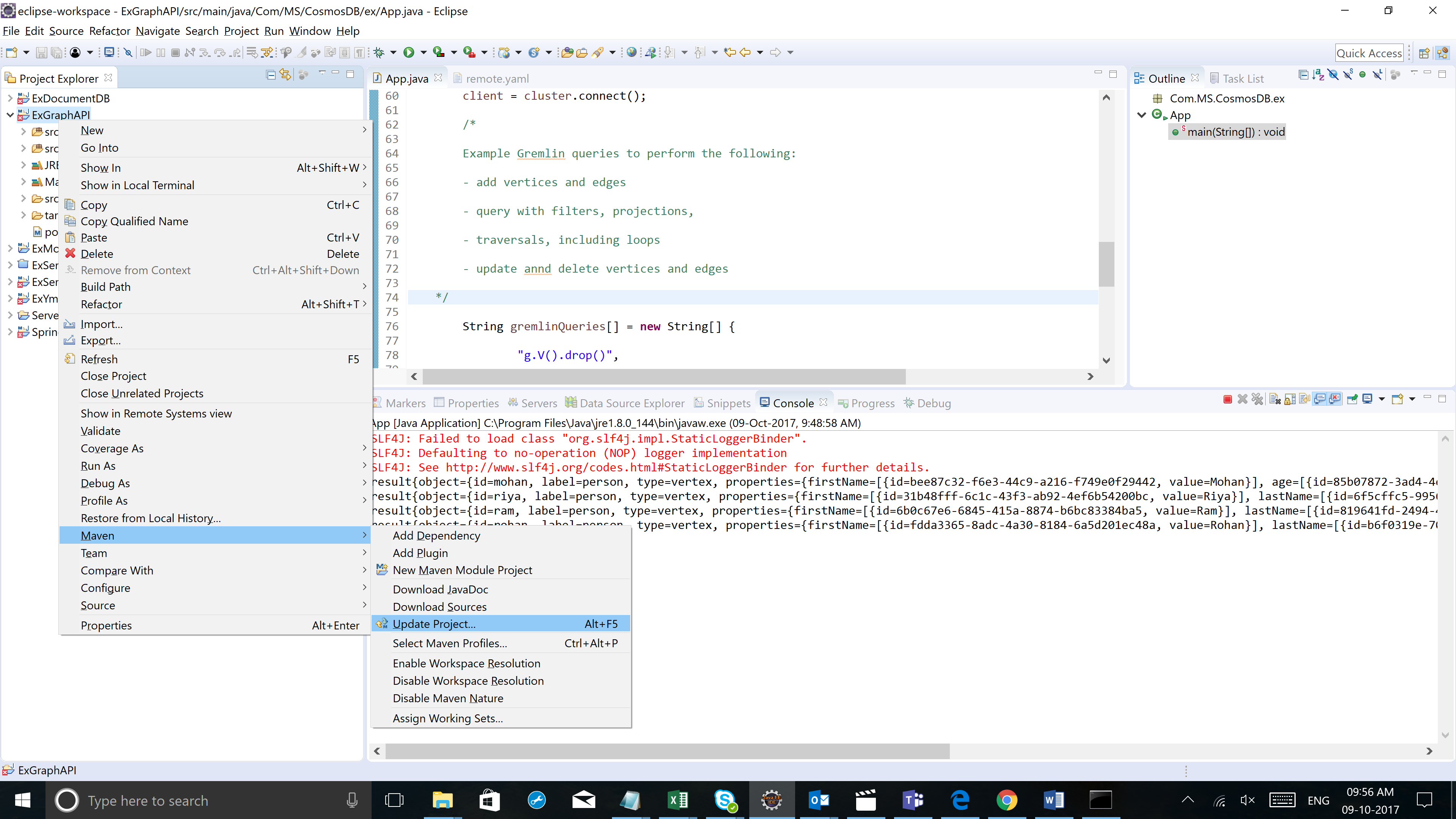
* + 1. Browse directory you did git clone



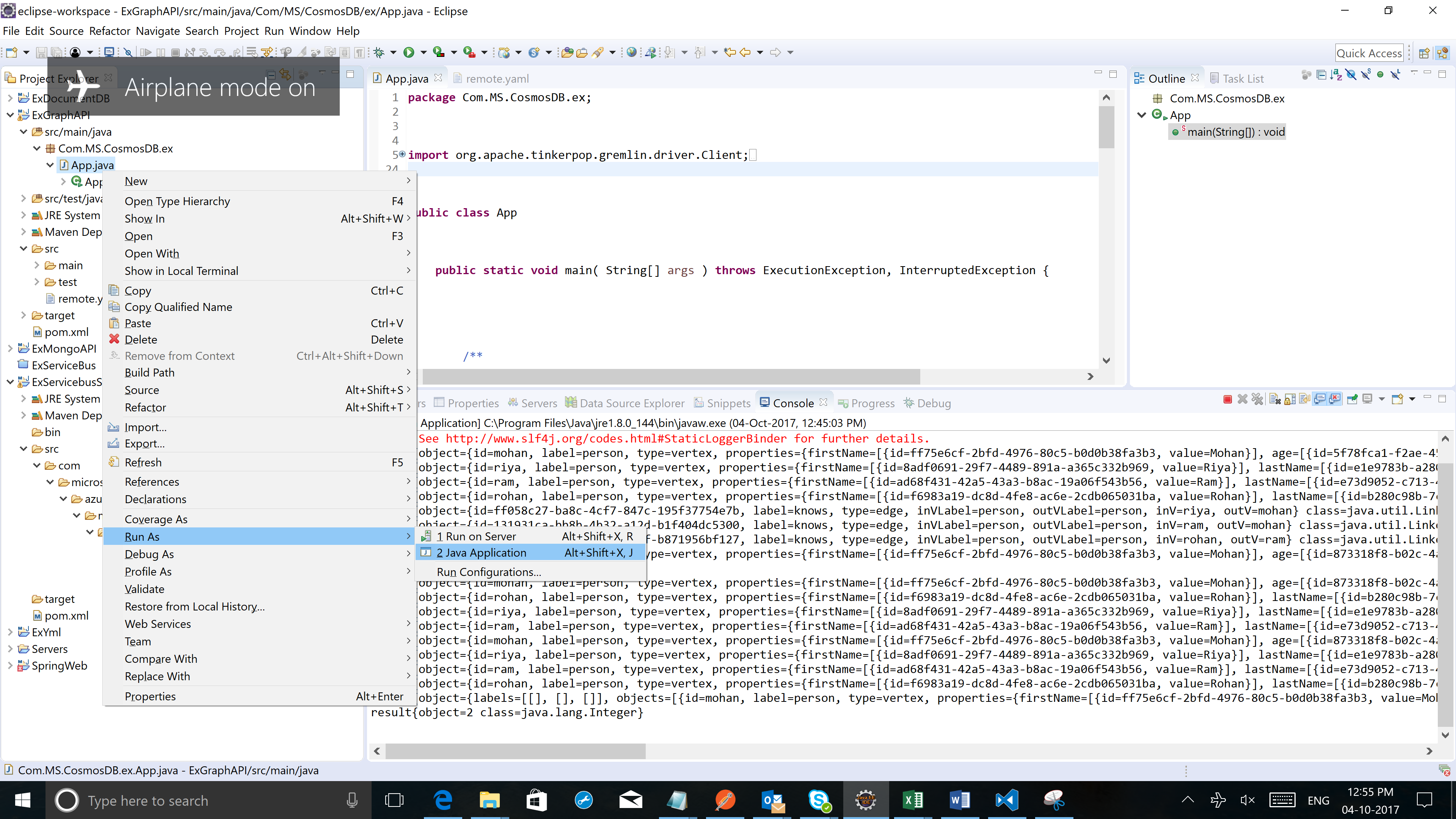
* + 1. Click on OK
    2. Select the folder with name CosmosDB-Graph-Example and click on Finish



* 1. Right click on project ->**Maven**-> **update project**



* 1. Right click App.java ->Run As -> Java Application



## Review data

1. Go to Azure Portal
2. Open Azure Cosmos DB Account
3. Select Data Explorer and Click on **Apply Filter** button

## Update Code

1. Go back to eclipse
2. Go to App.java and Replace

String gremlinQueries[] = **new** String[] {

"g.V().drop()",

"g.addV('person').property('id', 'mohan').property('firstName', 'Mohan').property('age', 44)",

"g.addV('person').property('id', 'riya').property('firstName', 'Riya').property('lastName', 'Kumar').property('age', 39)",

"g.addV('person').property('id', 'ram').property('firstName', 'Ram').property('lastName', 'singh')",

"g.addV('person').property('id', 'rohan').property('firstName', 'Rohan').property('lastName', 'Das')" };

With

String gremlinQueries[] = **new** String[] {

"g.V().drop()",

"g.addV('person').property('id', 'mohan').property('firstName', 'Mohan').property('age', 44)",

"g.addV('person').property('id', 'riya').property('firstName', 'Riya').property('lastName', 'Kumar').property('age', 39)",

"g.addV('person').property('id', 'ram').property('firstName', 'Ram').property('lastName', 'singh')",

"g.addV('person').property('id', 'rohan').property('firstName', 'Rohan').property('lastName', 'Das')",

"g.V('mohan').addE('knows').to(g.V('riya'))",

"g.V('mohan').addE('knows').to(g.V('ram'))",

"g.V('ram').addE('knows').to(g.V('rohan'))",

"g.V('mohan').property('age', 44)",

"g.V().count()",

"g.V().hasLabel('person').has('age', gt(40))",

"g.V().hasLabel('person').order().by('firstName', decr)",

"g.V('mohan').outE('knows').inV().hasLabel('person')",

"g.V('mohan').outE('knows').inV().hasLabel('person').outE('knows').inV().hasLabel('person')",

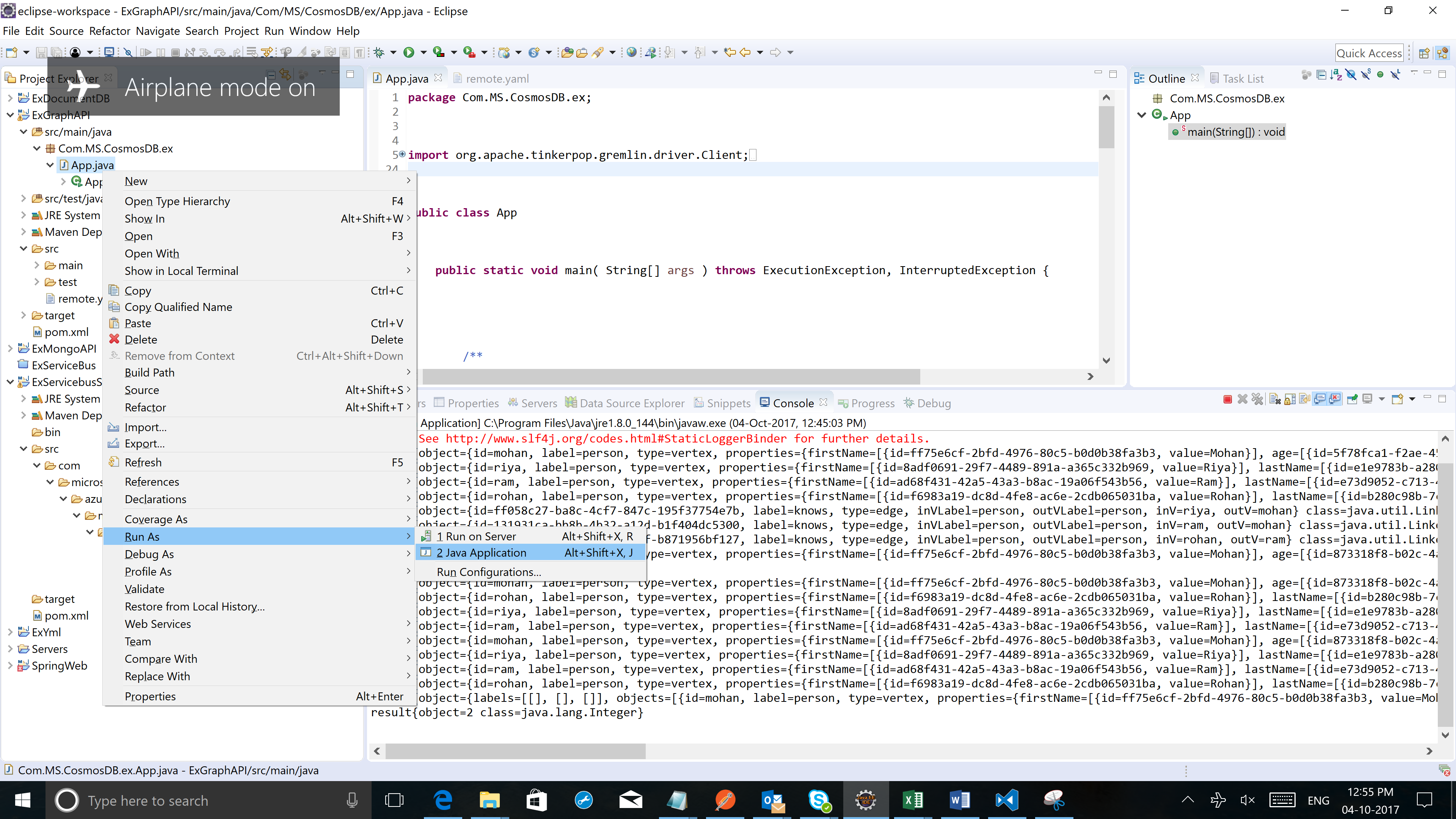
"g.V('mohan').repeat(out()).until(has('id', 'rohan')).path()",

"g.V('mohan').outE('knows').where(inV().has('id', 'riya')).drop()",

"g.E().count()",

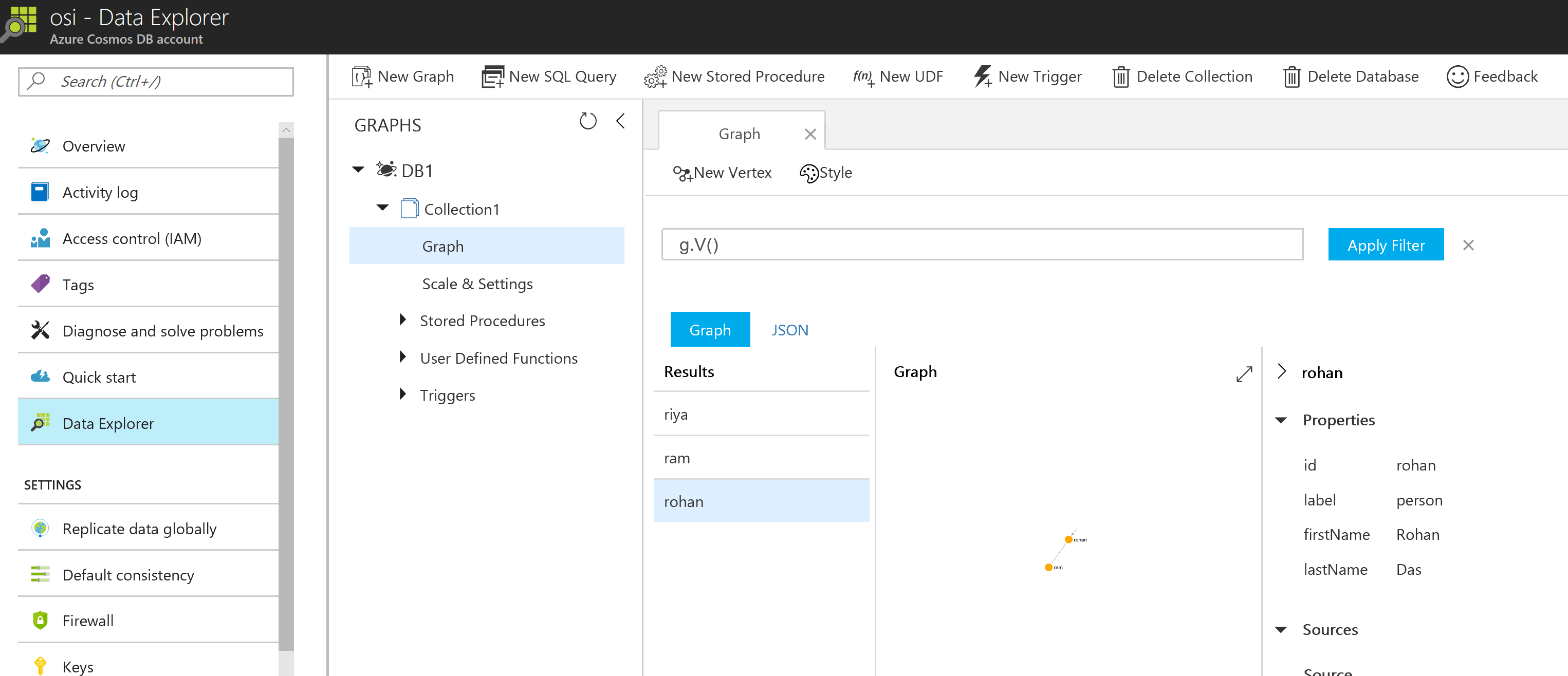
"g.V('mohan').drop()" };

1. Right click App.java ->Run As -> Java Application



## Review data

1. Go to Azure Portal
2. Open Azure Cosmos DB Account
3. Select Data Explorer and Click on **Apply Filter** button



**Clean up resources**

If you're not going to continue to use this app, delete all resources created by this quickstart in the Azure portal with the following steps:

1. From the left-hand menu in the Azure portal, click **Resource groups** and then click the name of the resource you created.
2. On your resource group page, click **Delete**, type the name of the resource to delete in the text box, and then click **Delete**.