**App Dev: Storing Application Data in Cloud Datastore v1.1**

**Overview**

In this lab, you will review the case study application, an online Quiz. You will store application data for the Quiz application in Cloud Datastore.

The Quiz application skeleton has already been written for you. You will clone a repository that contains the skeleton using Google Cloud Shell, review the code using the Cloud Shell editor, and view it using the Cloud Shell web preview feature.

Then you will modify the code that stores data to use Cloud Datastore.

**Objectives**

In this lab, you will learn how to perform the following tasks:

* Harness Cloud Shell as your development environment.
* Integrate Cloud Datastore into a NodeJS application.

**Previewing the Case Study Application**

In this section, you will access Cloud Shell, clone the git repository containing the Quiz application, and run the application.

**Clone source code in Cloud Shell**

1. On the **Cloud Platform Console**, click **Activate Google Cloud Shell**.
2. If a dialog box appears, click **Start Cloud Shell**.

I checked which project I was on with the command

***gcloud config list***

if I wasn’t on that project I will run the command

***gcloud config set project PROJECT\_ID***

To set the cloud to the project I want to work on

1. To clone the repository for the class, execute the following command:

***git clone*** [***https://github.com/GoogleCloudPlatform/training-data-analyst***](https://github.com/GoogleCloudPlatform/training-data-analyst)

**Configure and run the case study application**

1. To change the working directory, execute the following command:

***cd ~/training-data-analyst/courses/developingapps/nodejs/datastore/start***

1. I created an environment variable to store my GCP project ID (I replaced the $DEVSHELL\_PROJECT\_ID with my Project ID)

***export GCLOUD\_PROJECT=$DEVSHELL\_PROJECT\_ID***

1. Then I ran

***npm install***

To install all dependencies and

***npm start***

To start my server

1. Open the code on the cloud editor and made changes to the code to enable it to store data into datastore, saved all my changes and went back to the cloud shell

This code is need to import the Datastore module

**const config = require('../config');**

**//Load the @google-cloud/datastore module**

**const Datastore = require('@google-cloud/datastore');**

**TODO: Create a Datastore client object, ds**

**// Datastore should be used via the projectId property.**

**// The projectId is retrieved from the config module. This // module retrieves the project ID from the GCLOUD\_PROJECT // environment variable.**

**const ds = Datastore({**

**projectId: config.get('GCLOUD\_PROJECT')**

**});**

**// TODO: Declare a constant named kind**

**//The Datastore key is the equivalent of a primary key in a // relational database.**

**// Specify the kind, and let Datastore generate a unique // numeric id**

**const kind = 'Question';**

**// The create({quiz, author, title, answer1, answer2,**

**// answer3, answer4, correctAnswer}) function uses a**

**// destructuring assignment to extract properties from the form data passed to the function**

**function create({ quiz, author, title, answer1, answer2,**

**answer3, answer4, correctAnswer }) {**

**// TODO: Declare the entity key,**

**// with a Datastore generated id**

**const key = ds.key(kind);**

**// END TODO**

**// TODO: Declare the entity object, with the key and data**

**const entity = {**

**key,**

**// The entity's members are represented in a data property.**

**// This is an array where each element represents one**

**// member in the entity. Each element is an object with a // name and a value**

**data: [**

**{ name: 'quiz', value: quiz },**

**{ name: 'author', value: author },**

**{ name: 'title', value: title },**

**{ name: 'answer1', value: answer1 },**

**{ name: 'answer2', value: answer2 },**

**{ name: 'answer3', value: answer3 },**

**{ name: 'answer4', value: answer4 },**

**{ name: 'correctAnswer', value: correctAnswer },**

**]**

**};**

**// END TODO**

**// TODO: Save the entity, return a promise**

**// The ds.save(...) method returns a Promise to the**

**// caller, as it runs asynchronously.**

**return ds.save(entity);**

}

1. We need to app engine to running before we can have access to datastore, so I ran the command

***gcloud app create --region "us-central"***

This create my vm in app engine in us-central

After I ran to restart my server

***npm start***

and previewed the web application on the port it was running.

I tested if it was working by imputing some data and saving, finally I checked the datastore in the console, under entities I could see the data there.

Then the bonus quiz I wrote the code to query the datastore to return a JSON file of all the question while excluding the answers

**function list(quiz = 'gcp') {**

**const q = ds.createQuery([kind])**

**.filter('quiz', '=', quiz);**

**const p = ds.runQuery(q);**

**return p.then(([results, { moreResults, endCursor }]) => {**

**const questions = results.map(item => {**

**item.id = item[Datastore.KEY].id;**

**delete item.correctAnswer;**

**return item;**

**});**

**return {**

**questions,**

**nextPageToken:**

**moreResults != 'NO\_MORE\_RESULTS' ? endCursor : false**

**};**

**});**

**}**