Cheat Sheet: Firestore

Enable Firestore Native Mode

1. In the Google cloud console, click on **Navigation Menu | Firestore**
2. Click the **Select Native Mode** button
3. Select a location that matches where your other resources are running
4. Click the **Create Database** button

Access Firestore from Node.js

- In cloud shell, navigate to the **`internal`** folder and install Firestore node module:
  - **`npm install --save @google-cloud/firestore`**
- In the **`internal/server.js`**, require Firestore and initialize Firestore:

```
// bring in firestore
const Firestore = require("@google-cloud/firestore");

// initialize Firestore and set project id from env var
const firestore = new Firestore(
    {
        projectId: process.env.GOOGLE_CLOUD_PROJECT
    }
);
```

- Keep the **`internal/server.js`** file open, and make the edits on the next slides

Modify the POST /event Code as Follows:

```
app.post('/event', (req, res) => {
    // create a new object from the json data and add an id
    const ev = {
        title: req.body.title,
        description: req.body.description,
        id : mockEvents.events.length + 1
     }
     // this will create the Events collection if it does not exist
    firestore.collection("Events").add(ev).then(ret => {
        getEvents(req, res);
    });

});
```

Create a New Function to Retrieve Events

```
function getEvents(req, res) {
    firestore.collection("Events").get()
        .then((snapshot) => {
            if (!snapshot.empty) {
                const ret = { events: []};
                snapshot.docs.forEach(element => {
                    ret.events.push(element.data());
                }, this);
                console.log(ret);
                res.json(ret);
            } else {
                 res.json(mockEvents);
            }
        })
        .catch((err) => {
            console.error('Error getting events', err);
            res.json(mockEvents);
        });
};
```

Modify code for GET `/events` code as follows:

```
app.get('/events', (req, res) => {

    getEvents(req, res);

});
```

Deployment

- The internal server.js now needs an environment variable called `GOOGLE_CLOUD_PROJECT`
    - This variable stores the Google Cloud project ID
    - It is there by default in Cloud Shell
- To run your app in Kubernetes, you will need to add the environment variable for your internal deployment yaml:
    - Tip: look at the external deployment yaml for an env: example

```
env:
- name: GOOGLE_CLOUD_PROJECT
  value: <PROJECT_ID>
```

Deloitte.

ROITRAINING
MAXIMIZE YOUR TRAINING INVESTMENT

Likes

- If you have added likes, you will need the id of the event you are updating
- Firestore assigns unique ids automatically. You could make it available inside the returned object with code like this inside getEvents:

```
snapshot.docs.forEach(element => {
     //get data
     const el = element.data();
     //get internal firestore id and assign to object
     el.id = element.id;
     //add object to array
     ret.events.push(el);
}, this);
```