CAT 4

Alejandro Pérez Bueno

Dec 18, 2023

Table of Contents

Firebase preparation

Add project

We created a new project in firebase.google.com with Google Analytics on (see Figure 1a).

We then created a new database in test mode located in West Europe (see Figure 1b).

`user` collection

We created a new `user` collection with the following values from this *SQL* command:

```
INSERT INTO 'main'.'user'
  ('user_id', 'user_username', 'user_pwd', 'user_display_name')
  VALUES ('1', 'user1@uoc.com', '123456', 'Jane Doe')

INSERT INTO 'main'.'user'
  ('user_id', 'user_username', 'user_pwd', 'user_display_name')
  VALUES ('2', 'user2@uoc.com', '123456', 'John Doe')
```

See Figure 2a for an example of the setup screen corresponding to the first *SQL* command.

`seminar` collection

Now we will insert the three `seminar` entries from the `dbHelper` class:

```
INSERT INTO 'main'.'seminar'
  ('sem_id', 'sem_name', 'sem_duration') VALUES ('1', 'Dogs Agility Seminary','60')

INSERT INTO 'main'.'seminar'
  ('sem_id', 'sem_name', 'sem_duration') VALUES ('2', 'Medicine Seminary','40')

INSERT INTO 'main'.'seminar'
('sem_id', 'sem_name', 'sem_duration') VALUES ('3', 'AI Seminary','30')"
```

See Figure 2b for an example of the setup screen corresponding to the first *SQL* command.

`user_seminar` collection

Yet again, we will insert four `user_seminar` entries from the `dbHelper` class:

```
INSERT INTO 'main'.'user_seminar'
  ('usersem_user_id', 'usersem_seminar_id') VALUES ('1', '1')
```

```
INSERT INTO 'main'.'user_seminar'
  ('usersem_user_id', 'usersem_seminar_id') VALUES ('1', '3')

INSERT INTO 'main'.'user_seminar'
  ('usersem_user_id', 'usersem_seminar_id') VALUES ('2', '2')

INSERT INTO 'main'.'user_seminar'
  ('usersem_user_id', 'usersem_seminar_id') VALUES ('2', '3')
```

See Figure 2c for an example of the setup screen corresponding to the first *SQL* command.

`Add app button`

Note

Answered in project folder (see `build.gradle`).

Back-end programming

Info

This is a theoretical question

Where is the data located, compared to previous CATs? Has data access become faster or slower?

In this CAT, data is stored on the Firestore cloud database, whereas in CAT 3 it was stored locally. Data access is slower now, because there can be all kinds of delays in the network or in the cloud service, whereas storing the information locally is typically faster.

When method `loginAsync` ends, does it know at that time whether the login attempt is correct or not?

The `loginAsync` method does not know at that time whether the login attempt is correct or not, the result of the login is determined asynchronously and passed to the listener when it becomes available.

What is the role of listener that is passed as parameter to the `loginAsync` method?

The listener receives a Result object containing either a User object if the login was successful or an error message if the login failed.

Loading a user's seminars

`usersem_seminar_id`

Note

Answered in project folder (see `DataSourceFirebase.kt`).

create Seminary class instances

Note

Answered in project folder (see `DataSourceFirebase.kt`).

Saving images in Storage

Create the `item` collection with Auto-Id

Yet again, we will insert `item` entries from the `dbHelper` class. Here is an example of one of them:

```
INSERT INTO 'main'.'item'
  ('item_id', 'item_type', 'item_title', 'item_description', 'item_sem_id', 'item_imageref')
  VALUES ('1', '2', 'Obstacles', 'It consists of a an exposition about the obstacles available in
  Agility competitions and how to teach the dog to accomplish the tests.', '1', '')
```

See Figure 2d for an example of the setup screen corresponding to the first *SQL* command.

Note that in the screenshot the `image_path` is filled, it should be empty.

Add the image file

See Figure 3a, Figure 3b and Figure 3c to see the uploaded images and the corresponding rules enabled.

Populate the `selectItemsAsync` method

Note

Answered in project folder (see `DataSourceFirebase.kt`).

Add the code to upload the image

Note

Answered in project folder (see `ItemsAdapter.kt`).

Modify the detail fragment

Note

Answered in project folder (see `DetailFragment.kt`).

Display a user's requests

Add to Firestone the request collection with Auto-ID

Once again, we will insert `request` entries from the `dbHelper` class. Here is an example of one of them:

```
INSERT INTO 'main'.'request'
  ('request_id', 'request_type', 'request_title', 'request_description', 'request_user_id')
  VALUES ('1', '1', 'Location', 'Where is the seminar going to be placed?', '1')
```

See Figure 2e for an example of the setup screen corresponding to the first *SQL* command.

Load the list of user requests

Note

Answered in project folder (see `DataSourceFirebase.kt`).

Inserting new requests

Implement the `getNewRequestId` function

Note

Answered in project folder (see `DataSourceFirebase.kt`).

In what order should we perform these operations?

- Insert the image into Storage
- Get a new `request_id`

- Insert an item into the request collection from Firestore

Info

This is a theoretical question

The correct order will be:

1. Get a new `request_id`
2. Insert the image into the Firebase Storage
3. Insert an item into the request collection from Firestore

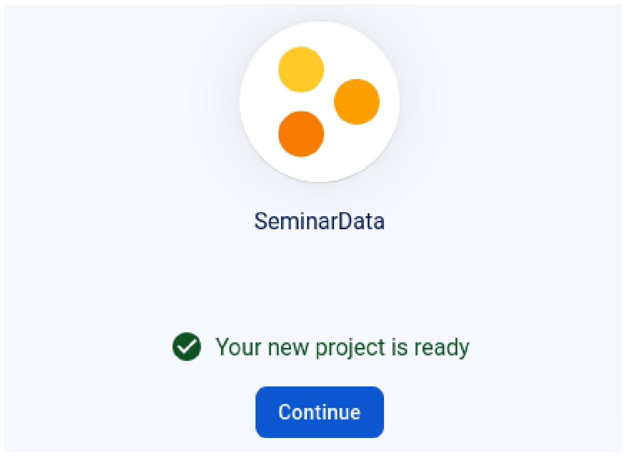
Insert the new request into the request collection

(WIP)

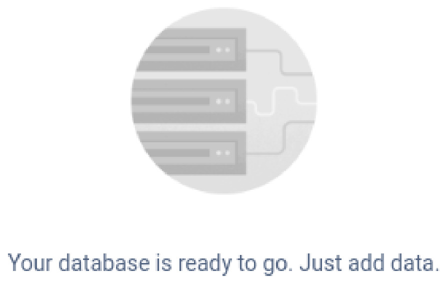
Create an item and add it to the inmemory model

(WIP)

Annexes

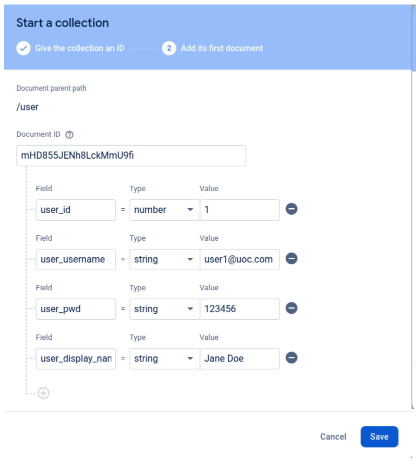


(a) `SeminarData` created



(b) Firestore database created

Figure 1: SeminarData Setup



Start a collection

Give the collection an ID — 2 Add its first document

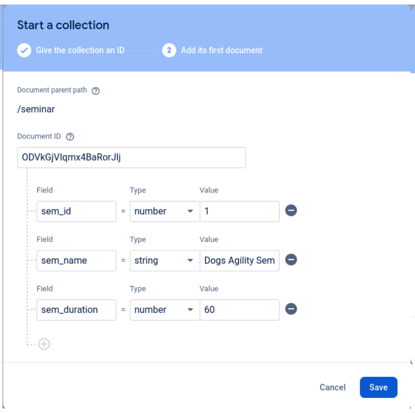
Document parent path: /user

Document ID: mHD855JENh8LckMmU9fi

Field	Type	Value
user_id	number	1
user_username	string	user1@uoc.com
user_pwd	string	123456
user_display_nan	string	Jane Doe

Cancel Save

(a) `user` collection



Start a collection

Give the collection an ID — 2 Add its first document

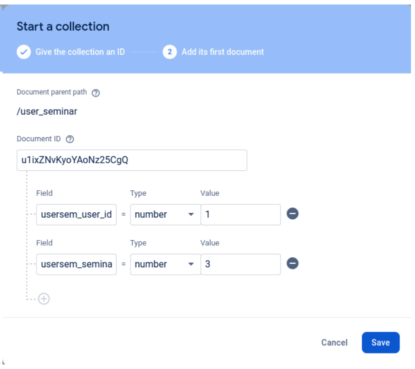
Document parent path: /seminar

Document ID: ODVkGjVlqmx4BaRorJlj

Field	Type	Value
sem_id	number	1
sem_name	string	Dogs Agility Sem
sem_duration	number	60

Cancel Save

(b) `seminar` collection



Start a collection

Give the collection an ID — 2 Add its first document

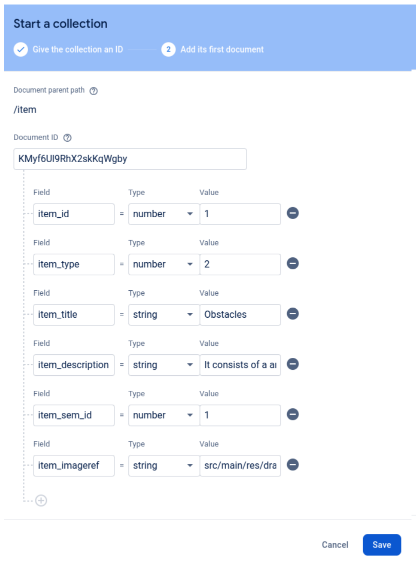
Document parent path: /user_seminar

Document ID: u1ixZNvKyoYAoNz25CgQ

Field	Type	Value
usersem_user_id	number	1
usersem_semina	number	3

Cancel Save

(c) `user_seminar` collection



Start a collection

Give the collection an ID — 2 Add its first document

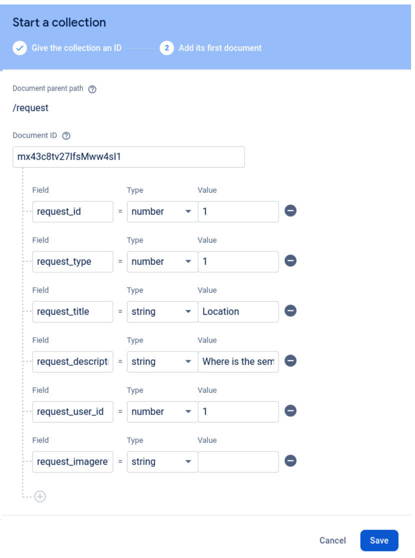
Document parent path: /item

Document ID: KMyf6Ul9RhX2skKqWgby

Field	Type	Value
item_id	number	1
item_type	number	2
item_title	string	Obstacles
item_description	string	It consists of a ai
item_sem_id	number	1
item_imageref	string	src/main/res/dra

Cancel Save

(d) `item` collection



Start a collection

Give the collection an ID — 2 Add its first document

Document parent path: /request

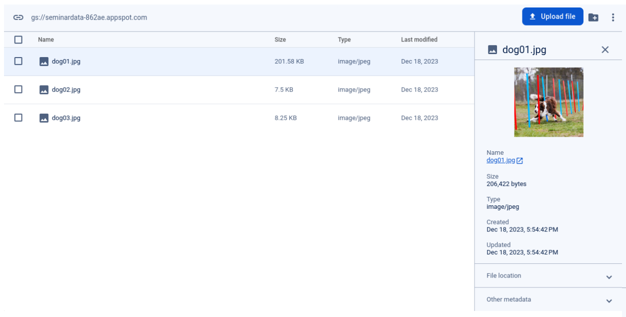
Document ID: mx43c8tv27IfsMww4sl1

Field	Type	Value
request_id	number	1
request_type	number	1
request_title	string	Location
request_descript	string	Where is the sem
request_user_id	number	1
request_imagere	string	

Cancel Save

(e) `request` collection

Figure 2: Collections setup



gs://seminardata-862ae.appspot.com

Name	Size	Type	Last modified
dog01.jpg	201.58 KB	image/jpeg	Dec 18, 2023
dog02.jpg	7.5 KB	image/jpeg	Dec 18, 2023
dog03.jpg	8.25 KB	image/jpeg	Dec 18, 2023

dog01.jpg

Name
dog01.jpg

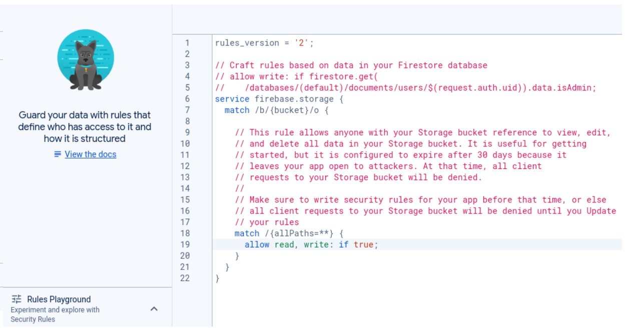
Size
206,422 bytes

Type
image/jpeg

Created
Dec 18, 2023, 5:54:42 PM

Updated
Dec 18, 2023, 5:54:42 PM

(a) `dog01.jpg`



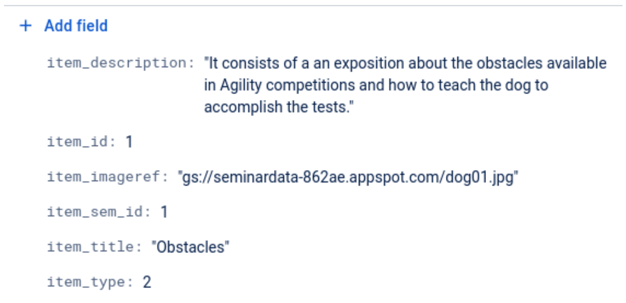
Guard your data with rules that define who has access to it and how it is structured

```
rules_version = '2';

// Craft rules based on data in your Firestore database
// allow write: if firestore.get(
//    /databases/(default)/documents/users/$(request.auth.uid)).data.isAdmin;
service firebase.storage {
  match /b/{bucket}/o {

    // This rule allows anyone with your Storage bucket reference to view, edit,
    // and delete all data in your Storage bucket. It is useful for getting
    // started, but it is configured to expire after 30 days because it
    // leaves your app open to attackers. At that time, all client
    // requests to your Storage bucket will be denied.
    //
    // Make sure to write security rules for your app before that time, or else
    // all client requests to your Storage bucket will be denied until you Update
    // your rules
    match /{allPaths=**} {
      allow read, write: if true;
    }
  }
}
```

(b) rules



```
item_description: "It consists of a an exposition about the obstacles available
                  in Agility competitions and how to teach the dog to
                  accomplish the tests."
item_id: 1
item_imageref: "gs://seminardata-862ae.appspot.com/dog01.jpg"
item_sem_id: 1
item_title: "Obstacles"
item_type: 2
```

(c) New ref to `dog01.jpg`

Figure 3: Firebase Storage