MOBILE DEVELOPMENT 2

W8- Firebase **Realtime** Database & **REST** API

CADT IDT

🥇 Course Objectives 🥇

- ✓ Understand how to interact with **Firebase Realtime Database** using **REST API**
- ✓ Perform **CRUD** (Create, Read, Update, Delete) operations using **HTTP requests**

- ✓ Handle **JSON serialization** and **deserialization**
- ✓ Handle **Async states** and **cache** optimizations

EXPLAIN

A Firebase **Real Time Database…**

Stores and sync data with a NoSQL cloud database

MORE INFO

- ✓ **Cloud-hosted** database
- ✓ **NoSQL** database
- ✓ Stores and syncs data in **JSON format**
- ✓ Access either with **REST API** or **Firebase SDK**

Let's jump to Firebase

EXPLAIN

A Firebase **Real Time Database…**

Is Document-oriented database, following JSON syntax

JSON Document

KEY VALUE *(another object)*

```
{
  "users": {
    "user1": {
      "name": "Alice",
      "age": 25,
      "email": "alice@example.com"
    },
    "user2": {
      "name": "Bob",
      "age": 30,
      "email": "bob@example.com"
    }
  }
}
```

KEY VALUE *(a string)*

- A **JSON document starts** with
  - either an object ({})
  - or an array ([]).
- An object is a collection of **key-value pairs**, where:
  - The key is always a string.
  - The value can be
    - another object
    - an array
    - a string, a number, a boolean, or null.

EXPLAIN

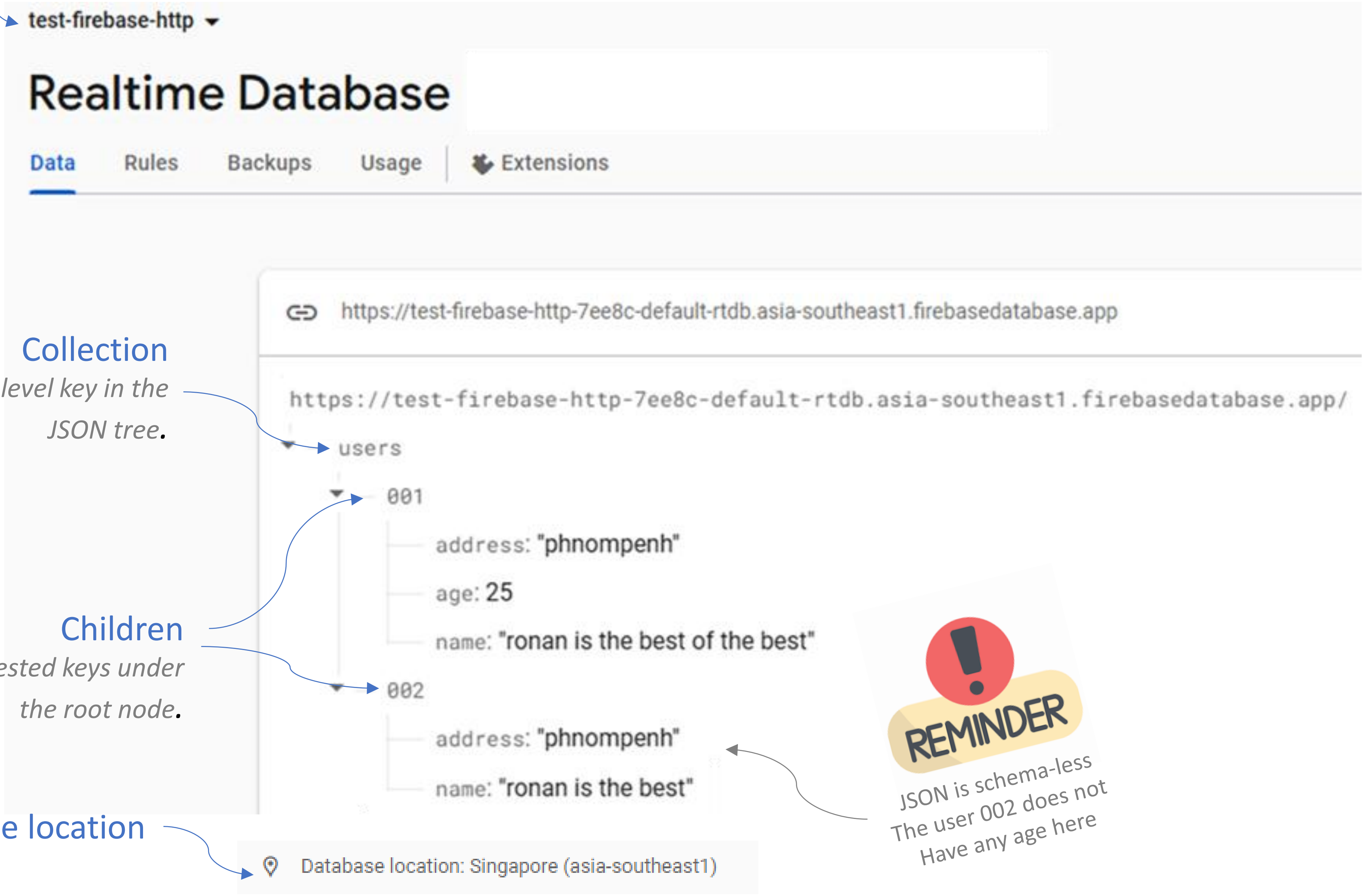
Your first **Firebase database**

Database name

Collection
top-level key in the JSON tree.

Children
Nested keys under the root node.

Database location



test-firebase-http

Realtime Database

Data Rules Backups Usage Extensions

https://test-firebase-http-7ee8c-default-rtdb.asia-southeast1.firebasedatabase.app

```
https://test-firebase-http-7ee8c-default-rtdb.asia-southeast1.firebasedatabase.app/
  users
    001
      address: "phnompenh"
      age: 25
      name: "ronan is the best of the best"
    002
      address: "phnompenh"
      name: "ronan is the best"
```

Database location: Singapore (asia-southeast1)

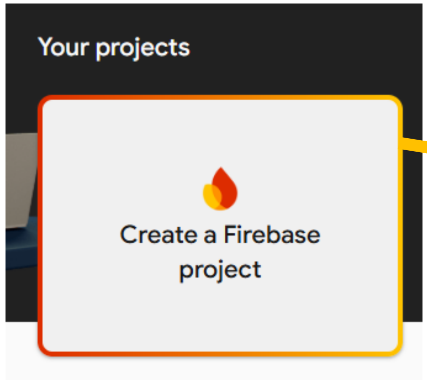
JSON is schema-less
The user 002 does not
Have any age here

ACTIVITY

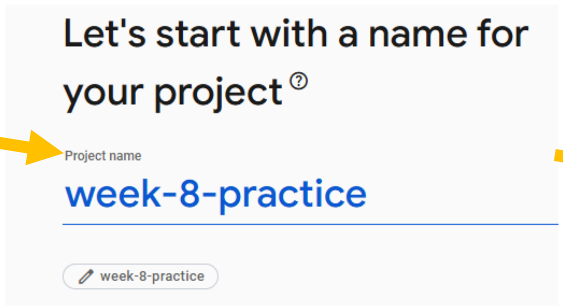
10 MIN

Your first **Firebase database**

Connect to Firebase console and create your first real time database


Your projects

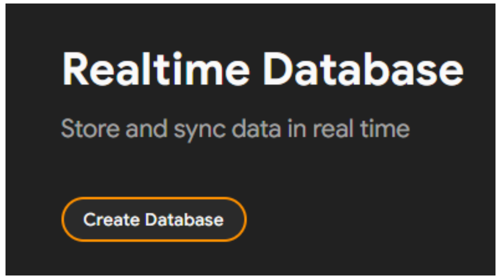
Create a Firebase project


Let's start with a name for your project

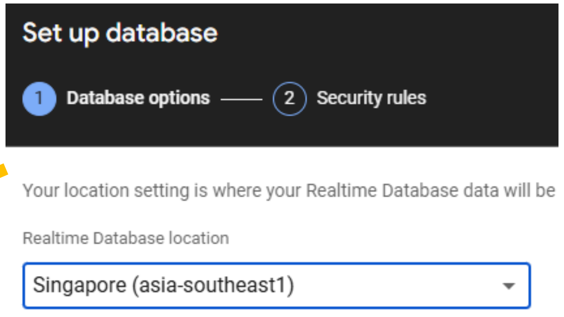
Project name

week-8-practice

week-8-practice

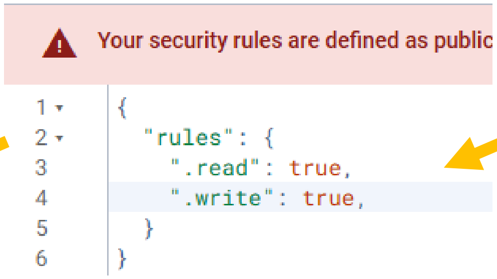

Realtime Database

Store and sync data in real time

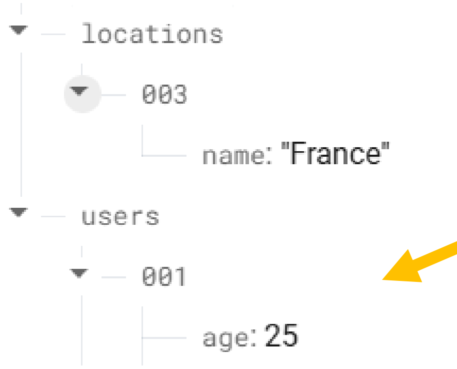

Set up database

Your location setting is where your Realtime Database data will be

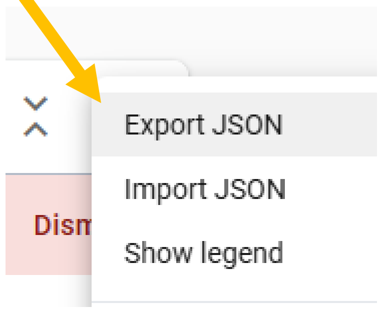
Realtime Database location: Singapore (asia-southeast1)


Your security rules are defined as public

```
{
  "rules": {
    ".read": true,
    ".write": true,
  }
}
```



```
locations
  003
    name: "France"
users
  001
    age: 25
```



Create a project with a real time database

- Go to Firebase Console.
- Click on "Create a Project".
- Enter a Project Name and click Continue

- In the Firebase Console, select your project.
- In the left sidebar, go to Build → Realtime Database.
- Click "Create Database".

- Choose a region (e.g., asia-southeast1).
- Select Start in Test Mode

Configure Database Rules

- In the Realtime Database tab, go to the Rules section.
- Change the rules to allow public access

Populate the database using the console

- Create 2 collections
- Add 2 children on each collection with appropriate attributes

Export the database to JSON

- Click on Export on the contextual menu
- Observe how the JSON reflect the database

EXPLAIN

MORE INFO

MORE INFO

Firebase SDK vs REST API

Two approaches to connect to a firebase Realtime database

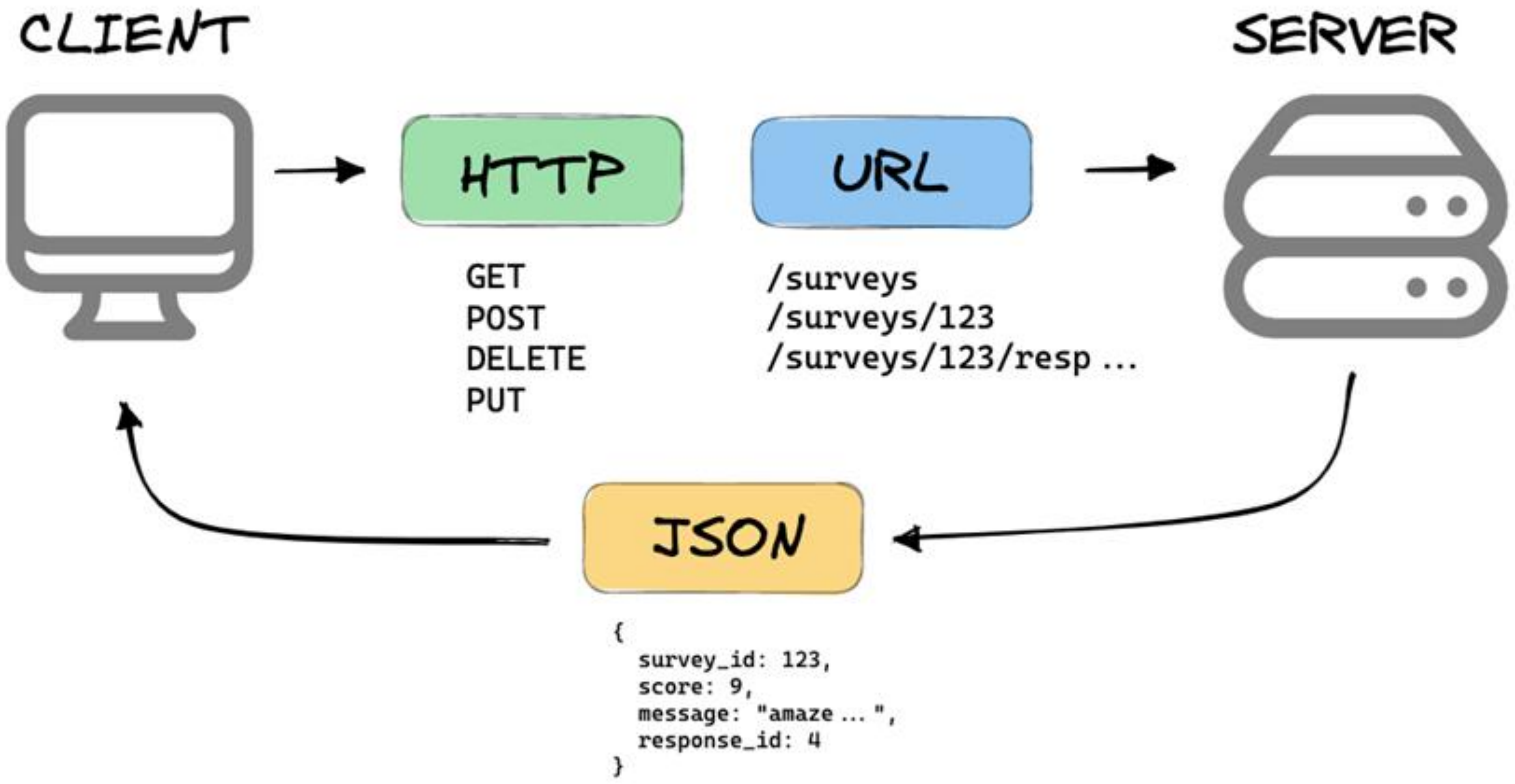
What we Are using today

Feature	Firebase SDK	REST API
Setup	Requires Firebase libraries	Works via HTTP requests
Authentication	Uses Firebase Authentication	Requires manual token handling
Real-Time Sync	Built-in WebSocket connection	Requires polling or long polling
Ease of Use	Provides direct database bindings	Requires RESTful HTTP methods
Offline Support	Yes	No (unless handled manually)

EXPLAIN

What is a **REST API** ?

A REST API is a standard used between **Clients** who want to access information from the web from **Servers** who have access to that information.



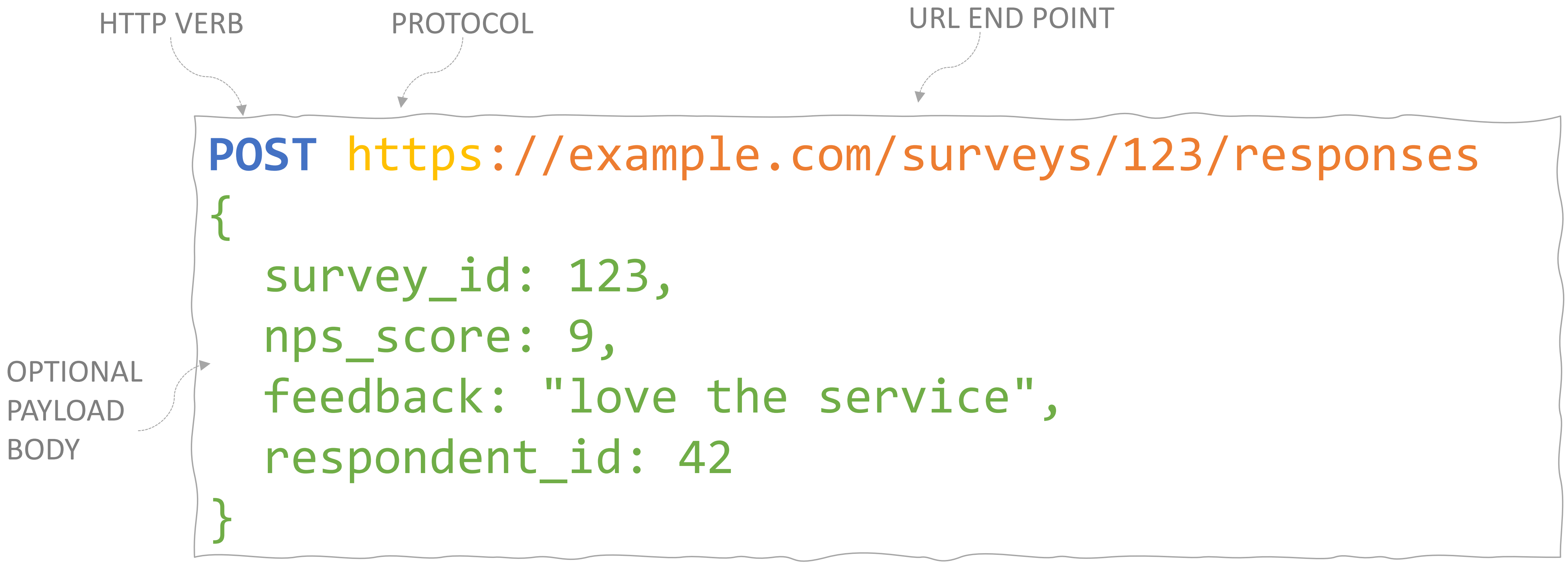
```
{
  survey_id: 123,
  score: 9,
  message: "amaze ... ",
  response_id: 4
}
```

EXPLAIN

Taxonomy of a **RESTful** API

MORE INFO

3 components : **URL endpoint**, **HTTP verb,** **payload body**



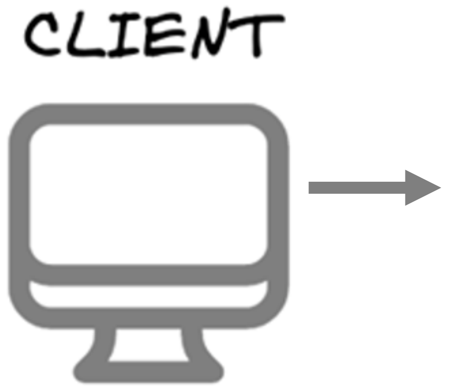
```
POST https://example.com/surveys/123/responses
{
  survey_id: 123,
  nps_score: 9,
  feedback: "love the service",
  respondent_id: 42
}
```

EXPLAIN

HTTP **Verbs** *(or methods)*

MORE INFO

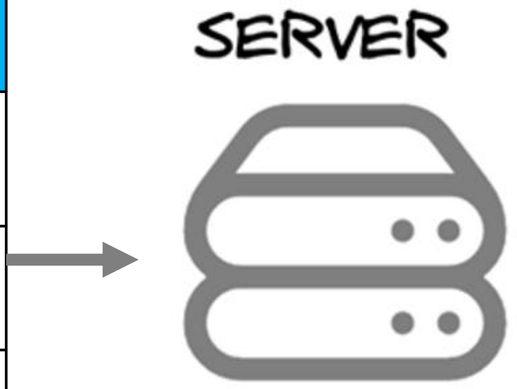
There are 5 basic verb commands when making a HTTP request



VERB	MEANING	EXAMPLE

GET	Retrieve all users
	Retrieve a user
		`/users`
		`/users/{id}`

POST	Create a user	`/users`
PUT / PATH	Update a user	`/users/{id}`
DELETE	Delete a user	`/users/{id}`



EXPLAIN

Realtime Database **REST API**

Database URL

Nodes

GET

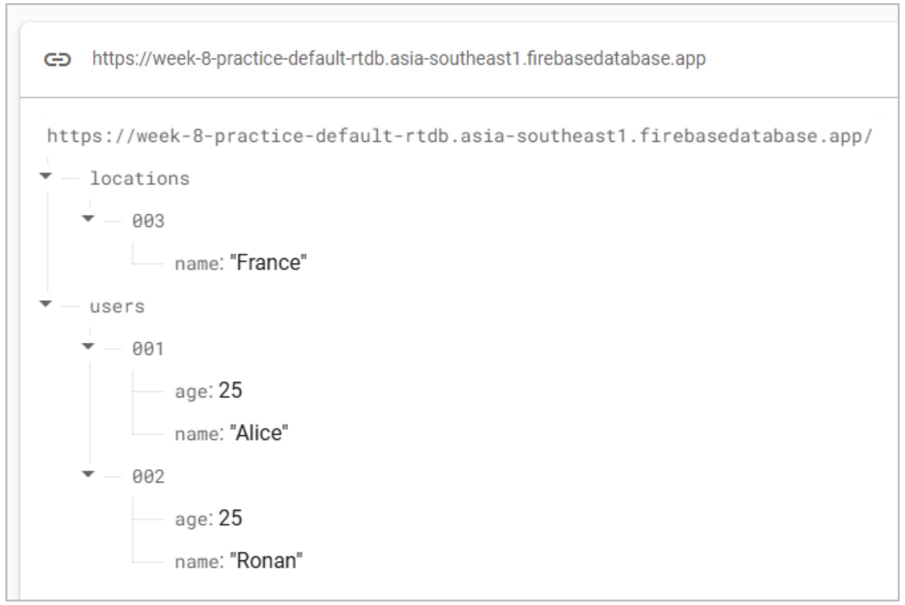
https://test-firebase-http-7ee8c-default-rtdb.asia-southeast1.firebasedatabase.app/users.json

Database name

Firebase server

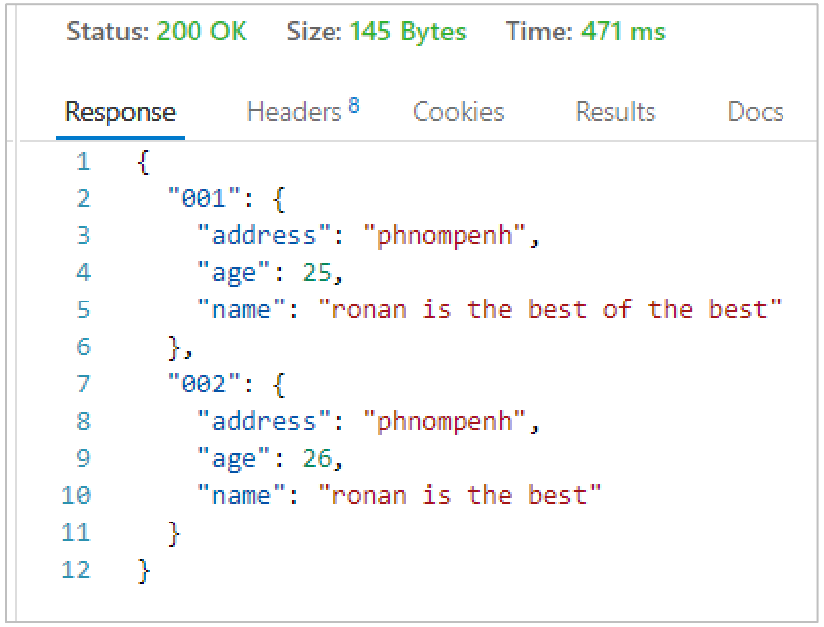
FIREBASE RESP API Support HTTPS protocol only

The REQUEST shall Specify the .JSON extension



https://week-8-practice-default-rtdb.asia-southeast1.firebasedatabase.app

```
https://week-8-practice-default-rtdb.asia-southeast1.firebasedatabase.app/
  locations
    003
      name: "France"
  users
    001
      age: 25
      name: "Alice"
    002
      age: 25
      name: "Ronan"
```



Status: 200 OK   Size: 145 Bytes   Time: 471 ms

```
{
  "001": {
    "address": "phnompenh",
    "age": 25,
    "name": "ronan is the best of the best"
  },
  "002": {
    "address": "phnompenh",
    "age": 26,
    "name": "ronan is the best"
  }
}
```

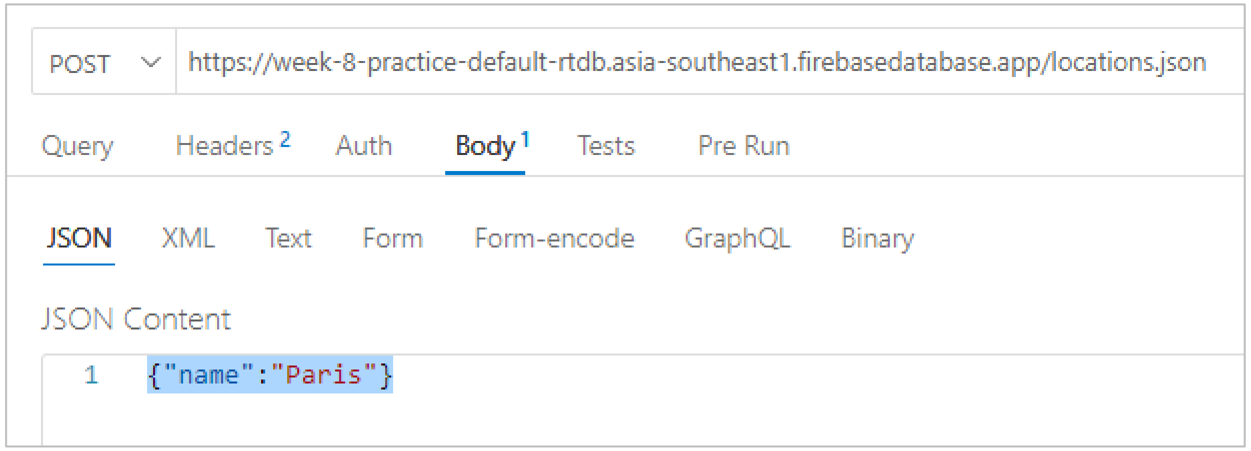
EXPLAIN

POST REQUEST & Generated ID

When you add data using Firebase REST API, Firebase automatically generates unique IDs.

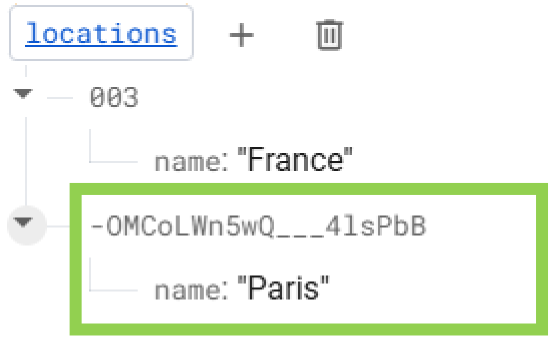
POST `https://week-8-practice-default-rtdb.asia-southeast1.firebasedatabase.app/locations.json`

BODY `{"name":"Paris"}`

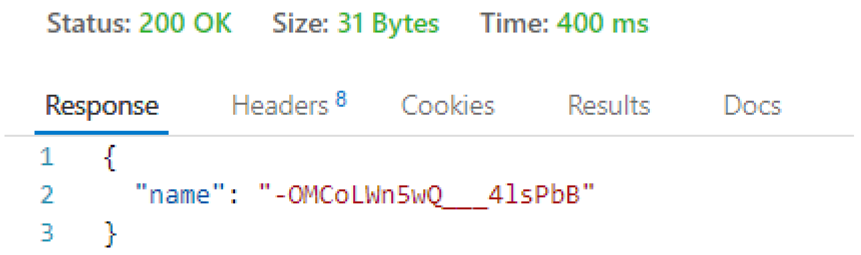


POST https://week-8-practice-default-rtdb.asia-southeast1.firebasedatabase.app/locations.json

```
{"name":"Paris"}
```



```
locations
  003
    name: "France"
  -OMCoLWn5wQ___41sPbB
    name: "Paris"
```



Status: 200 OK   Size: 31 Bytes   Time: 400 ms

```
{
  "name": "-OMCoLWn5wQ___41sPbB"
}
```

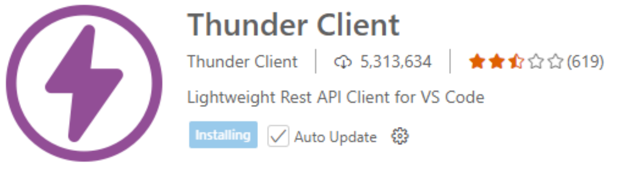
ACTIVITY

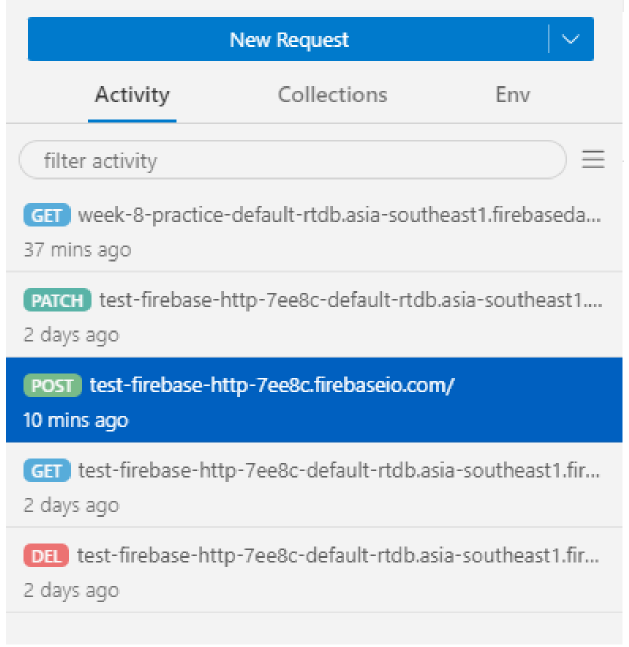
10 MIN

Let’s **CRUD** with a REST API client

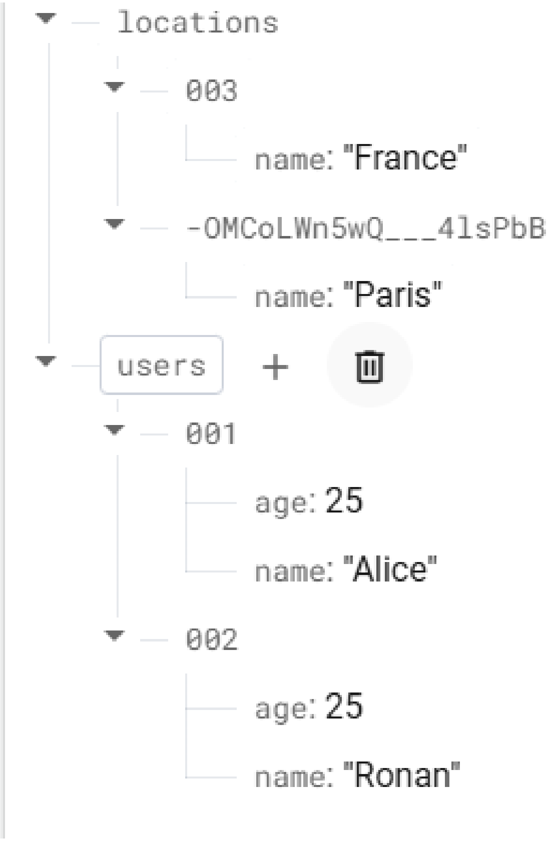
MORE INFO

You need to install a REST API client


Thunder Client
Thunder Client | 5,313,634 | (619)
Lightweight Rest API Client for VS Code
Installing ✓ Auto Update



GET week-8-practice-default-rtdb.asia-southeast1.firebaseda...
37 mins ago
PATCH test-firebase-http-7ee8c-default-rtdb.asia-southeast1....
2 days ago
POST test-firebase-http-7ee8c.firebaseio.com/
10 mins ago
GET test-firebase-http-7ee8c-default-rtdb.asia-southeast1.fir...
2 days ago
DEL test-firebase-http-7ee8c-default-rtdb.asia-southeast1.fir...
2 days ago


```
locations
    003
        name: "France"
    -OMCoLWn5wQ___4lsPbB
        name: "Paris"
users
    001
        age: 25
        name: "Alice"
    002
        age: 25
        name: "Ronan"
```

Perform a POST (add an item)

- On Thunder client, click New Request → Select POST.
- Enter your Firebase Database URL + XXX.json
- XXX is the name of a collection
- In the body, specify the JSON content
- Click SEND

Firebase will respond with an automatically generated unique ID

- Get this ID from RESPONSE *(copy this ID for the next steps)*
- Check also on Firebase console you item has bee added

Perform a PATCH (modify an item)

- On Thunder client, click New Request → Select PATCH.
- Enter your Firebase Database URL with the previous ID + .json
- In the body, specify a specific attribute to change on this item
- Click SEND

- Perform a GET to check the changes
- Check also on Firebase console you item has been updated

Perform a DELETE

- On Thunder client, click New Request → Select DELETE.
- Enter your Firebase Database URL with the previous ID + .json
- Click SEND

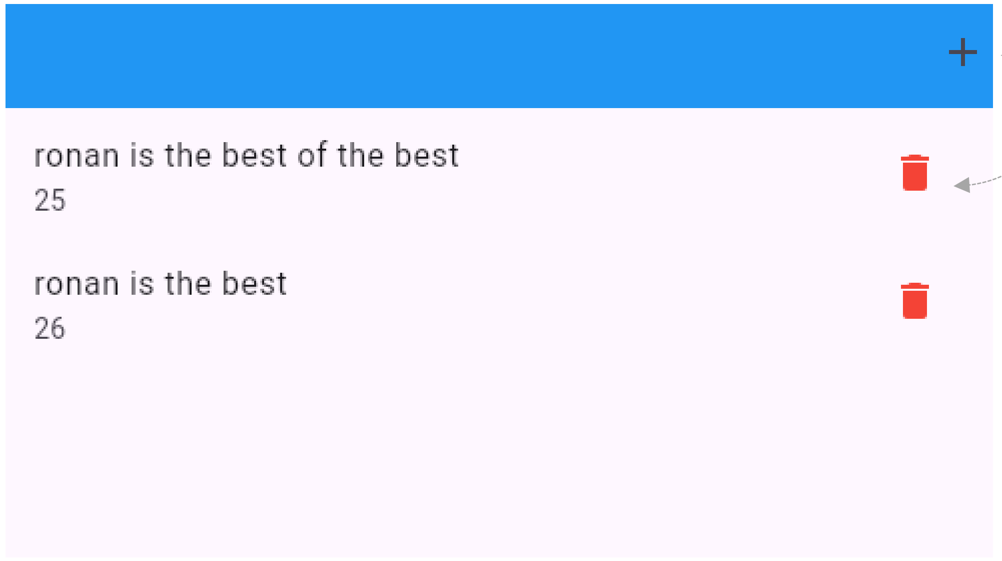
- Perform a GET to check the changes
- Check also on Firebase console you item has been deleted

ACTIVITY

60 MIN

Build a **CRUD App** w a Firebase Repo

You goal is now to bind your Firebase Database with a Flutter APP and test CRUD operations


ronan is the best of the best
25

ronan is the best
26

List all items

Create a new item

Remove the item

ACTIVITY

60 MIN

Build a **CRUD App** w a Firebase Repo

Replace pancake with you entities type !

DATA

DTO

+ pancakes_dto.dart

Finally create the firebase repo and the DTO Json/model

REPOSITORY

+ pancakes _repository.dart
+ pancakes _mock_repository.dart
+ pancakes _firebase_repository.dart

Define the abstract repo with CRUD methods

Test it with a mock repo

MODEL

MODEL

+ pancakes _model.dart

First create the model of your entity

UI

PROVIDER

+ pancakes _provider.dart

Create the provider, handle the entity state using AsyncData

SCREENS

+ pancakes_screen

Create the screen and test it with the mock repo

ACTIVITY

10 MIN

Build a **CRUD App** w a Firebase Repo

Replace pancake with you entities type !

Important points to notice

ACTIVITY

10 MIN

Build a **CRUD App** w a Firebase Repo

Replace pancake with you entities type !

Optimisation of the cache

async/await action + fetch
async/await action + cache update
asynchronous action + cache update + recovery action

ACTIVITY

10 MIN

Build a **CRUD App** w a Firebase Repo

BONUS

FORM TO ADD
REMOVE ACTION

BONUS Secure database access with Firebase Authentication

- Database security rules
- Authentication tokens

https://firebase.google.com/docs/database/rest/auth

7 - BONUS Offline/cache approaches

Now you know

✓ Understand how to interact with **Firebase Realtime Database** using **REST API**

✓ Perform **CRUD** (Create, Read, Update, Delete) operations using **HTTP requests**

✓ Handle **JSON serialization** and **deserialization**

✓ Handle **Async states** and **cache** optimizations

RESOURCES

Here are the tools and resource referenced in this session

FIREBASE REALTIME DATABASE
https://firebase.google.com/docs/database/rest/start
https://retool.com/blog/your-guide-to-crud-in-firebase-realtimedb-with-rest-api
https://www.youtube.com/watch?v=RW_luvxS0Rs

REST API
https://mannhowie.com/rest-api
https://www.youtube.com/watch?v=tkfVQK6UxDI

AUTHENTICATION WITH REST API
https://firebase.google.com/docs/database/rest/auth

GOING FURTHER
https://www.youtube.com/watch?v=cYinms8LurA
https://www.youtube.com/watch?v=joVi3thZOqc&list=PLl-K7zZEsYLmgdxMEHar35Wo26fLWm9BI&index=2