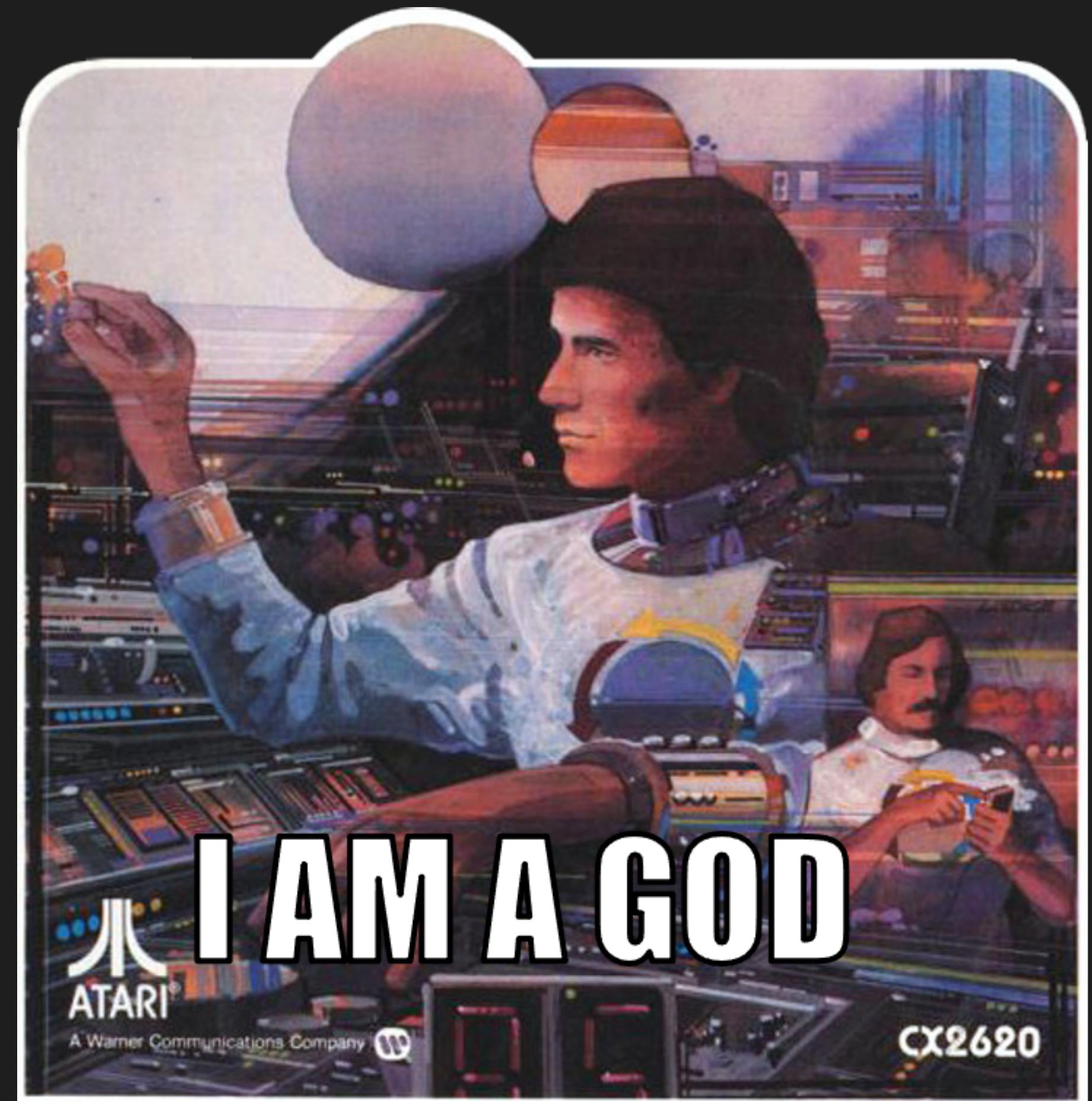
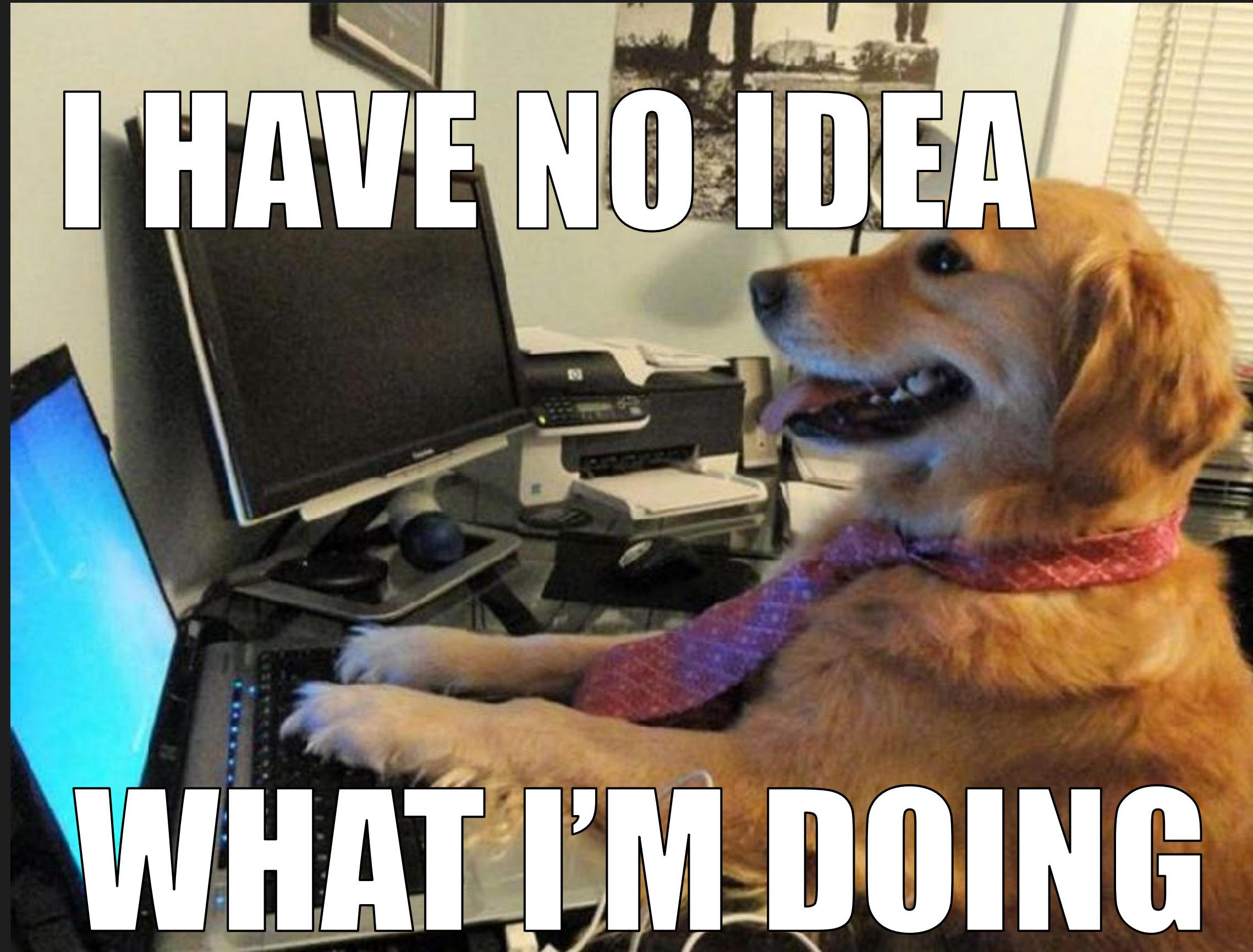


React has `useState`.
A React Native Developer has two states:





Firebase



Firebase

1. Introduction to Firebase
2. Client Server
3. CRUD & REST
4. Realtime Database
REST API
5. Setup Firebase
6. Post App Part 5:
Firebase and CRUD

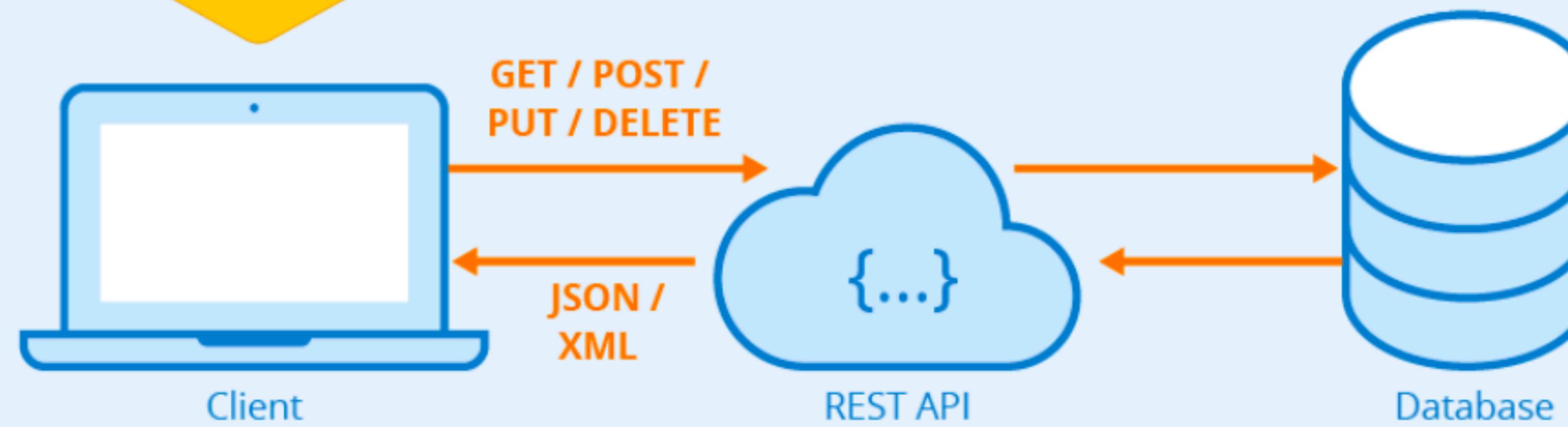
When someone asks you how to get data from a database in a js code



made with mematic



Firebase





What is Firebase?

Platform, a suite of tools & Backend-as-a-Service
for Web & App Development

100 *SECONDS OF*



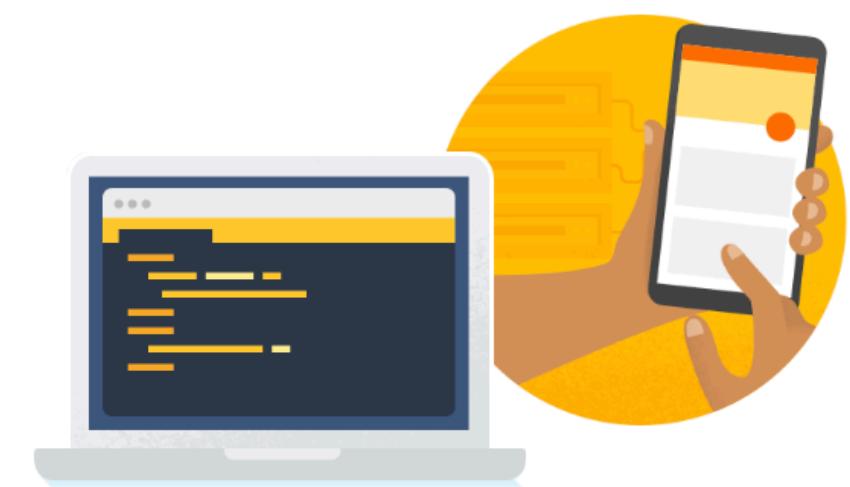
<https://www.youtube.com/watch?v=vAoB4VbhRzM>





Introducing Firebase

<https://www.youtube.com/watch?v=iosNuldQoy8>



Build better apps



Cloud Firestore

Store and sync app data at global scale



ML Kit BETA

Machine learning for mobile developers



Cloud Functions

Run mobile backend code without managing servers



Authentication

Authenticate users simply and securely



Hosting

Deliver web app assets with speed and security



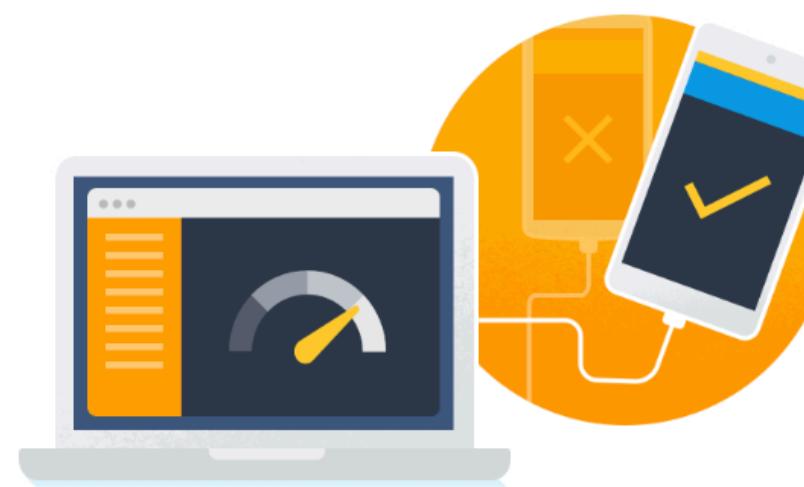
Cloud Storage

Store and serve files at Google scale



Realtime Database

Store and sync app data in milliseconds



Improve app quality



Crashlytics

Prioritize and fix issues with powerful, realtime crash reporting



Performance Monitoring

Gain insight into your app's performance



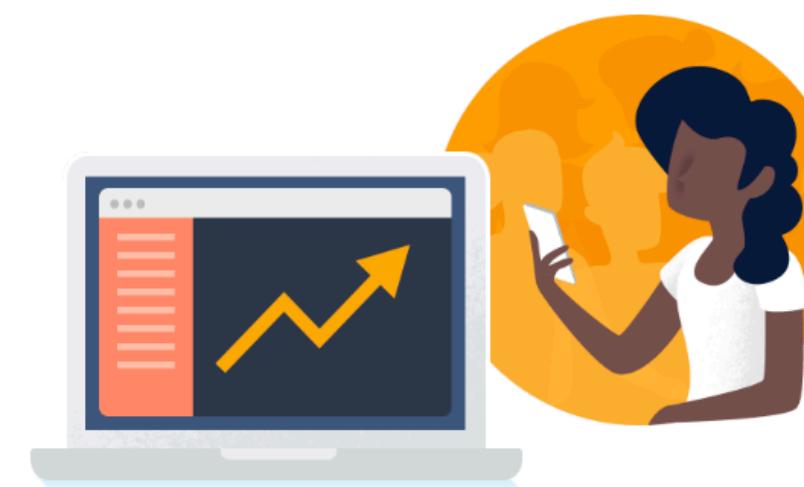
Test Lab

Test your app on devices hosted by Google



App Distribution BETA

Distribute pre-release versions of your app to your trusted testers



Grow your business



In-App Messaging BETA

Engage active app users with contextual messages



Google Analytics

Get free and unlimited app analytics



Predictions

Smart user segmentation based on predicted behavior



A/B Testing BETA

Optimize your app experience through experimentation



Cloud Messaging

Send targeted messages and notifications



Remote Config

Modify your app without deploying a new version



Dynamic Links

Drive growth by using deep links with attribution

MANY DEVICES
ONE PLATFORM





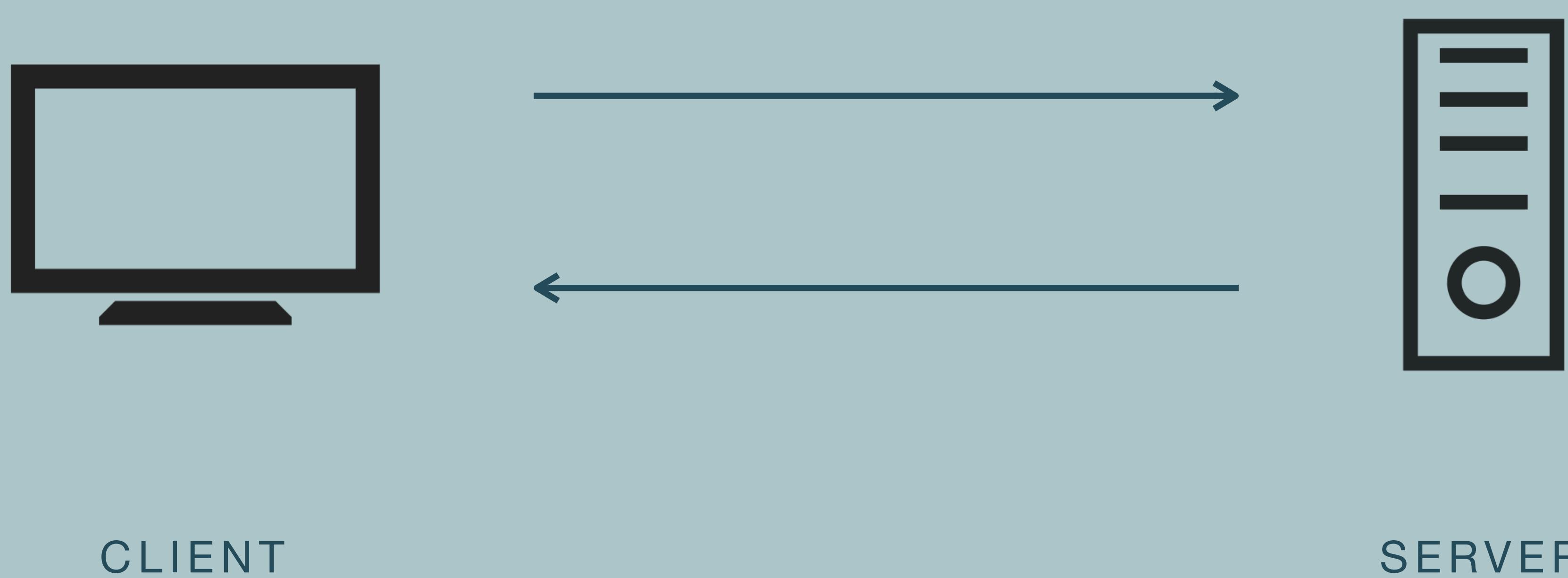
Do we need Backend
Developers?

Client Server

The Basic Architecture of the Web

Client-Server Model

Communication between web **clients** and web **servers**.



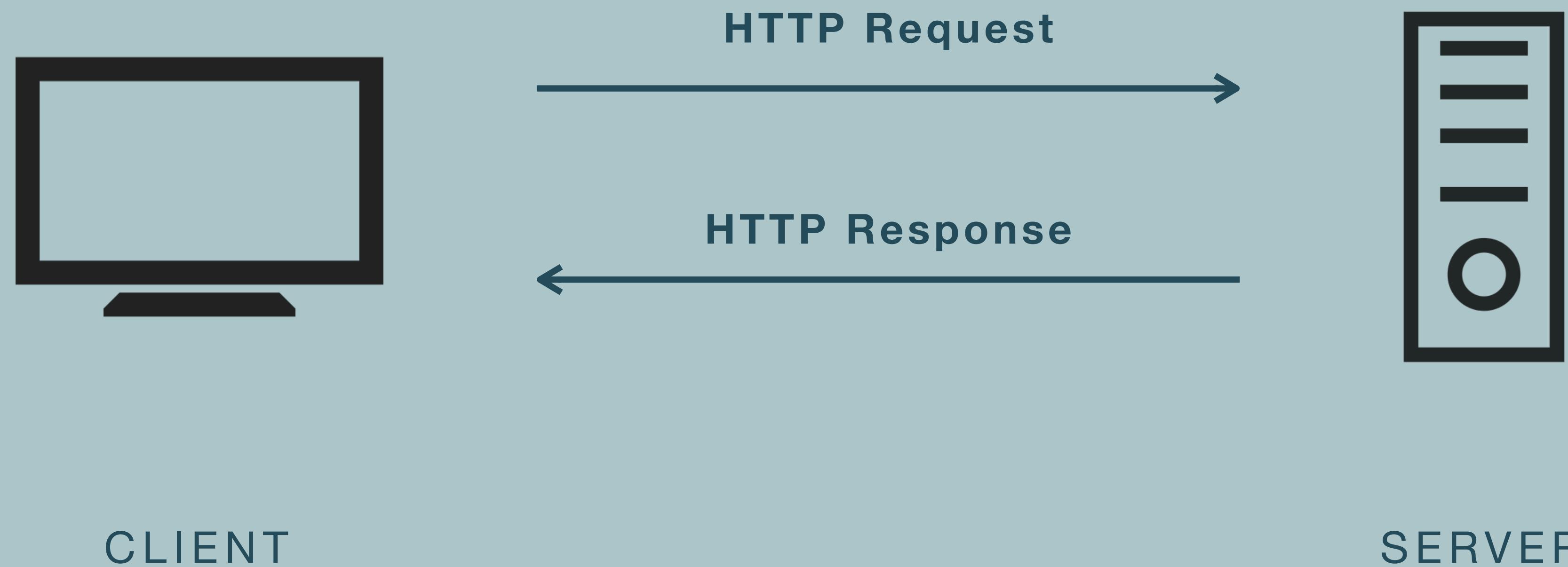
Client-Server Model

Communication between web **clients** and web **servers**.

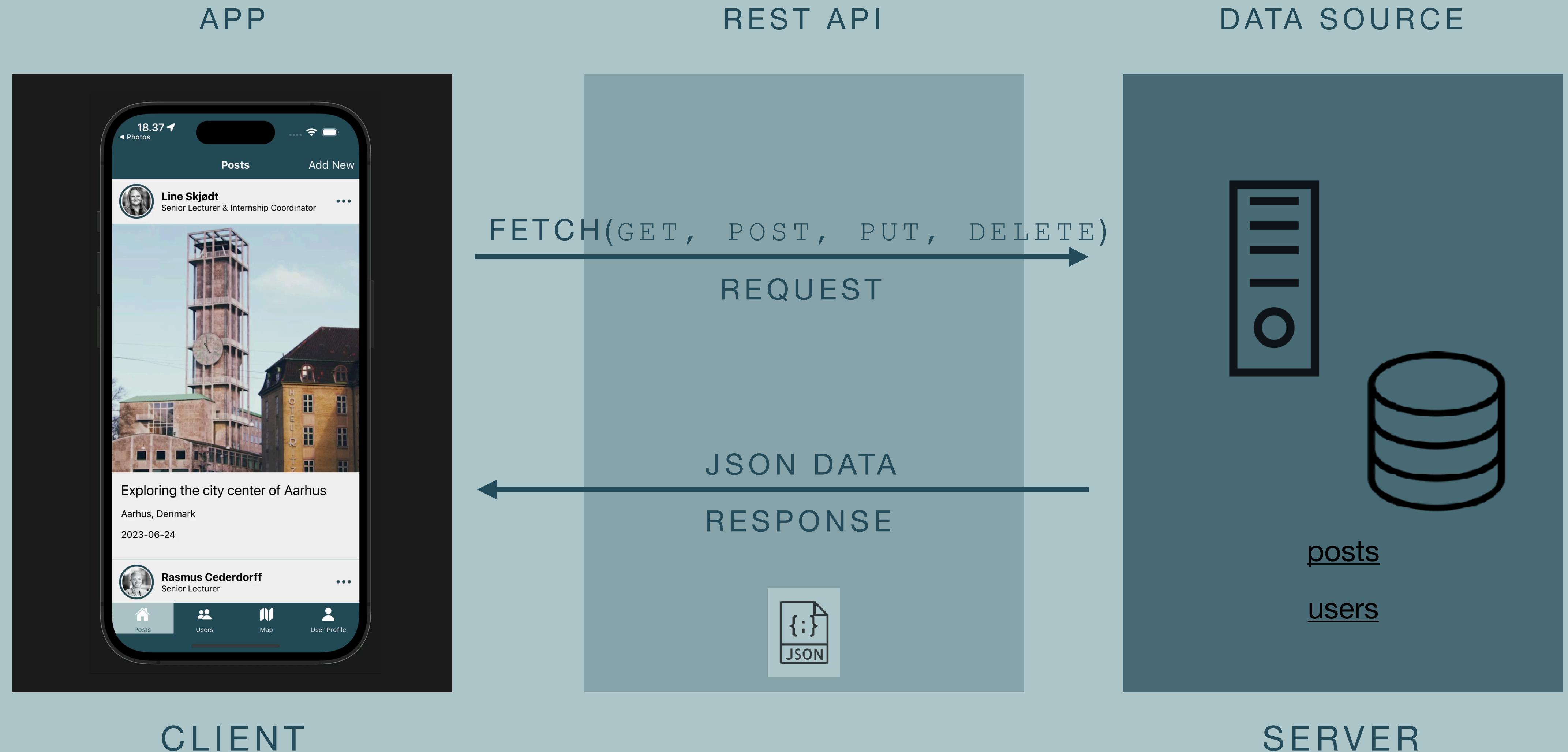


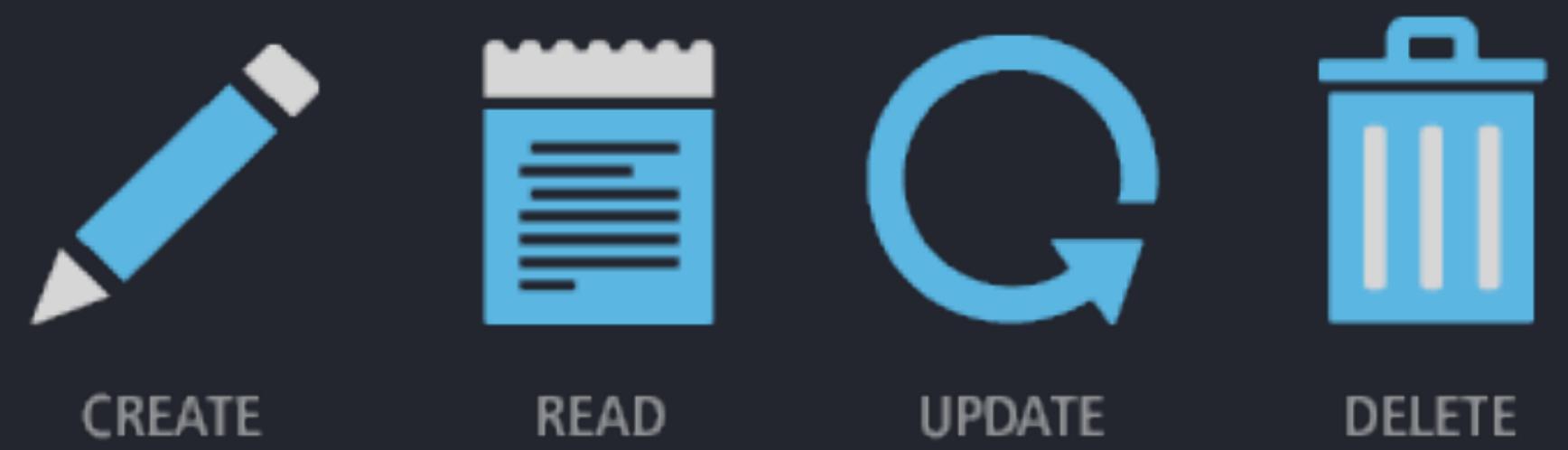
Client-Server Model

Communication between web **clients** and web **servers**.



Client Server Architecture





C R U D

What's CRUD?

- CREATE objects like a post, user, movie, product, etc.
- READ objects like an array (or object) of objects (posts, users, movies, products, etc)
- UPDATE an object, often given by a unique id.
- DELETE an object, often given by a unique id.

What's REST?

GET

POST

PUT

DELETE

- REpresentational State Transfer
- A standard for systems (client & server) to communicate over HTTP to retrieve or modify (data) resources.
- Stateless, meaning the two systems don't need to know anything about the state.
- The client makes the requests using the 4 basic HTTP verbs to define the operation.

REST API Design & CRUD



CRUD operations

Create

/api/products **POST**

Read

/api/products **GET**

Update

/api/products/:id **PUT**

Delete

/api/products/:id **DELETE**

100 *SECONDS OF* node



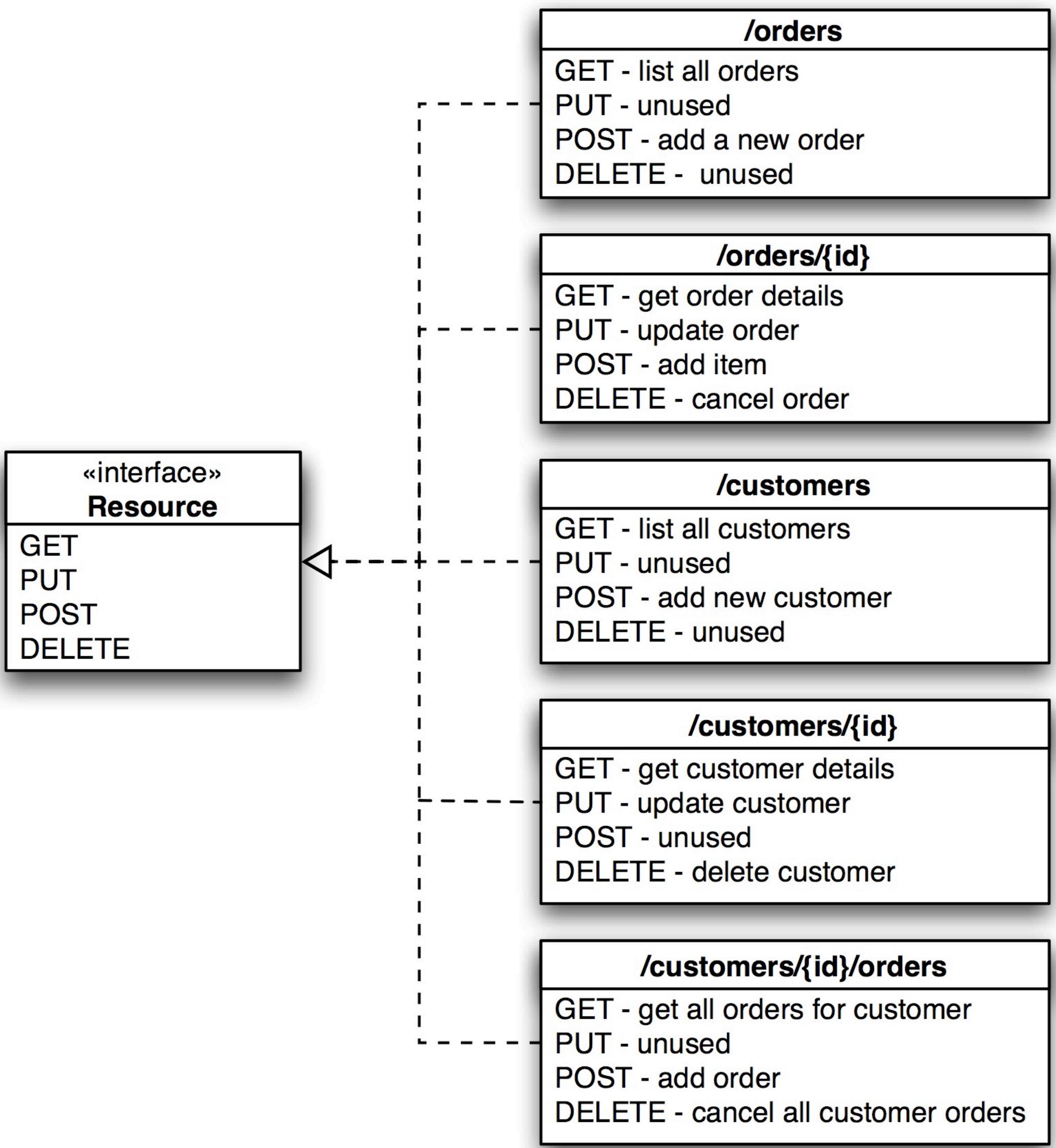
GET ➤➤➤

REST

API

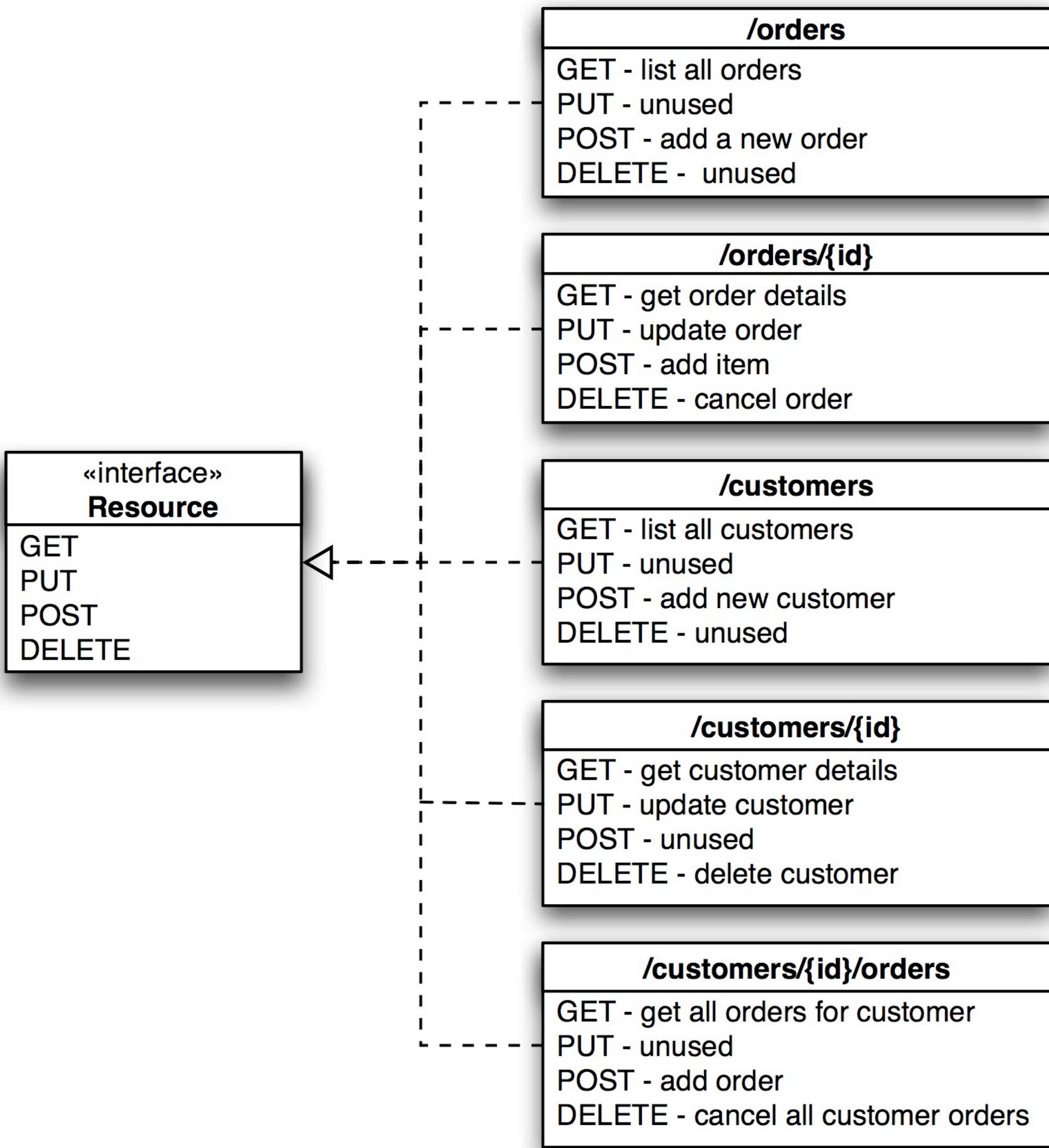
I'M A TEAPOT
418 ←

Principles of REST



- REST defines a set of rules and guidelines on how you can interact with an API:
 - With REST, each piece of data in the database is treated as a resource. It has a unique id and URL. The URL structure represents the hierarchy of the data, allowing you to access specific data nodes.
 - You can use standard HTTP methods like GET, PUT, POST, PATCH, and DELETE to read, write, update, or delete that data.
 - REST promotes stateless communication, meaning that each request contains everything the server needs to process it.
 - The data exchanged with the API is typically in JSON format.

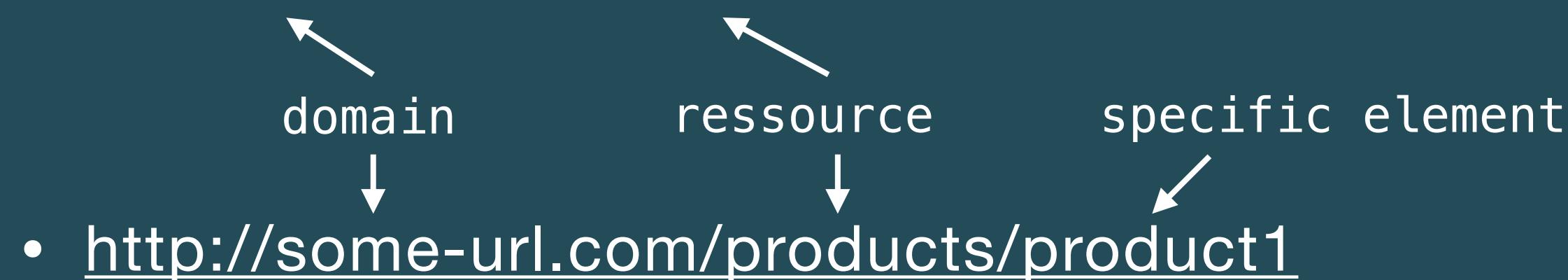
Principles of REST



- Resource identification:** Each piece of data in the Firebase Realtime Database is treated as a resource and is identified by a unique URL (Uniform Resource Locator). The URL structure represents the hierarchy of the data, allowing you to access specific data nodes.
- HTTP methods:** RESTful APIs utilize standard HTTP methods to perform operations on resources. The Firebase Realtime Database REST API supports the following methods:
 1. GET: Retrieves data from the specified endpoint.
 2. PUT: Replaces or updates data at the specified endpoint.
 3. POST: Appends data to a specified endpoint, generating a unique key.
 4. PATCH: Updates specific fields in the data at the specified endpoint.
 5. DELETE: Removes data at the specified endpoint.
- Stateless communication:** Each request sent to the Firebase Realtime Database REST API contains all the necessary information for the server to process it. The API does not maintain any session or state information between requests. This statelessness allows for scalability and simplicity.
- Data format:** The data exchanged with the Firebase Realtime Database REST API is typically in JSON format. JSON provides a lightweight and flexible way to represent structured data.

RESTful API

- Base URL: http://some-url.com/products



- Data type → JSON

Ressource	GET	POST	PUT	DELETE
Collection: <u>http://some-url.com/products</u>	Returns a list with all products	Creates new product, added to the collections	Replaces a collection with a another	Deletes all products
Element: <u>http://some-url.com/products/product1</u>	Returns a specific product	÷	Replaces product with new (updated) data	Deletes product

HTTP REQUEST METHODS (verbs)

GET - POST - PUT - DELETE

HTTP (Hypertext Transfer Protocol) is the standard way to communicate between clients and servers (request-response protocol).

"HTTP defines a set of **request methods** to indicate the desired action to be performed for a given resource."

https://www.w3schools.com/tags/ref_httpmethods.asp

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

CRUD vs REST & HTTP Verbs

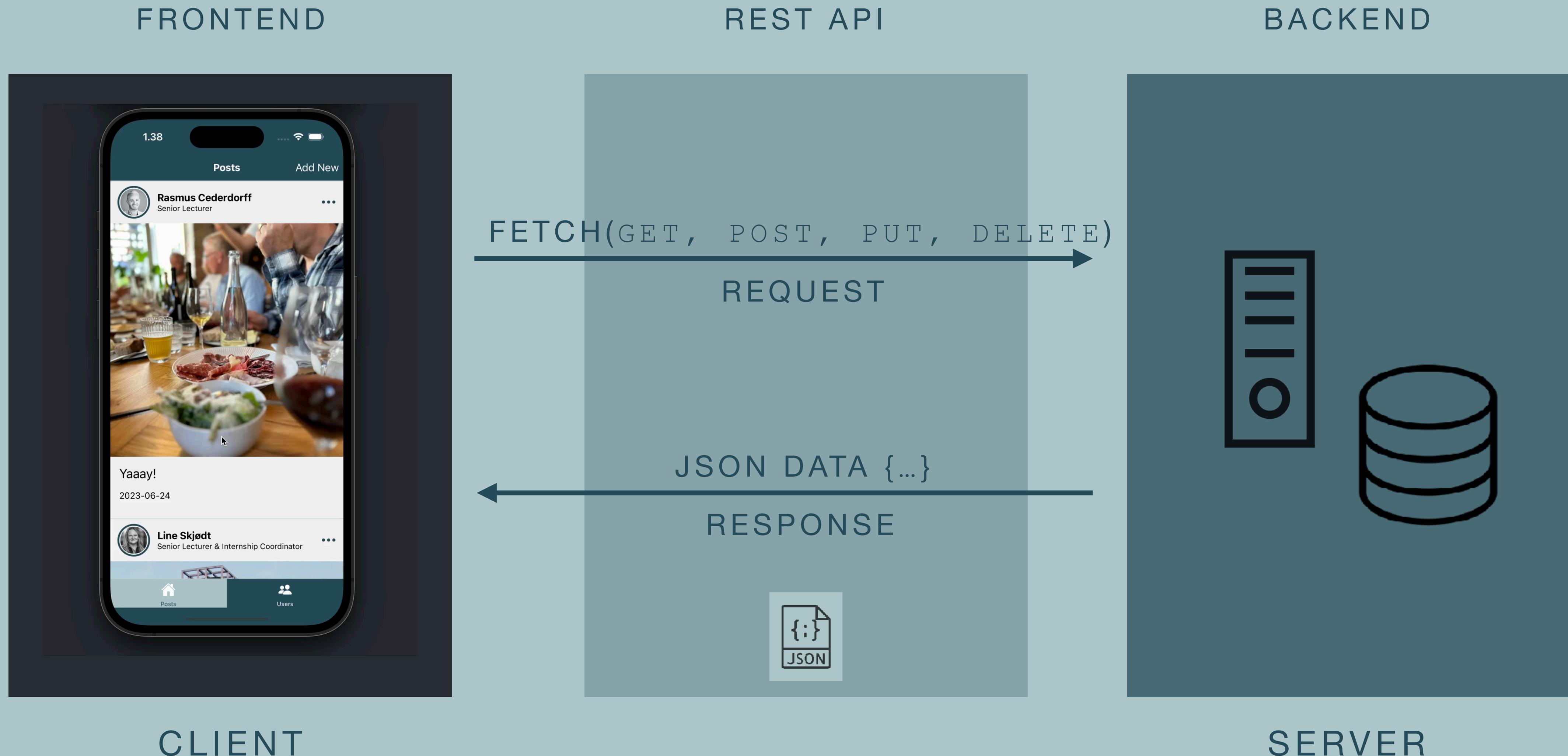
CREATE -> POST: create a new resource (object)

READ -> GET: retrieve a specific resource or a collection

UPDATE -> PUT / PATCH: update a specific resource (by id)

DELETE -> DELETE: remove a specific resource by id

HTTP Request & Response





Realtime Database & REST API

Store and sync data in real time
REST API or SDK

<https://firebase.google.com/products/realtime-database>

Realtime Database REST API

[https://firebase.google.com/docs/
database/rest/start](https://firebase.google.com/docs/database/rest/start)

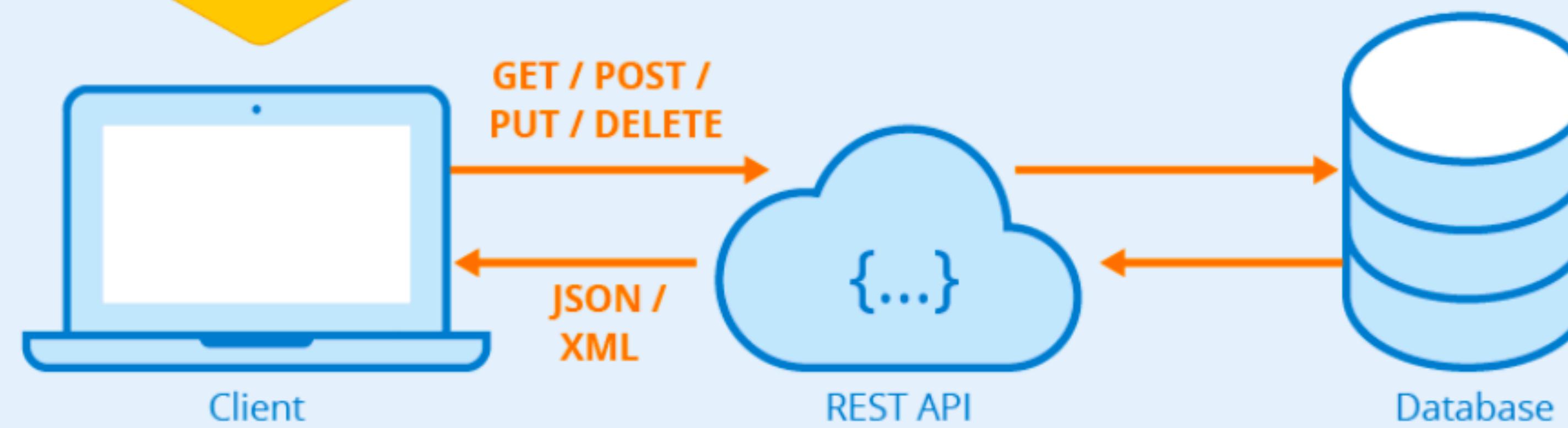
```
export default function PostsPage() {
  const [posts, setPosts] = useState([]);
  const [showLoader, dismissLoader] = useIonLoading();

  async function getPosts() {
    const response = await fetch("https://race-rest-default-rtbd.firebaseio.com/posts.json");
    const data = await response.json();
    // map object into an array with objects
    const postsArray = Object.keys(data).map(key => ({ id: key, ...data[key] }));
    return postsArray;
  }

  async function getUsers() {
    const response = await fetch("https://race-rest-default-rtbd.firebaseio.com/users.json");
    const data = await response.json();
    // map object into an array with objects
    const users = Object.keys(data).map(key => ({ id: key, ...data[key] }));
    return users;
  }
}
```



Firebase



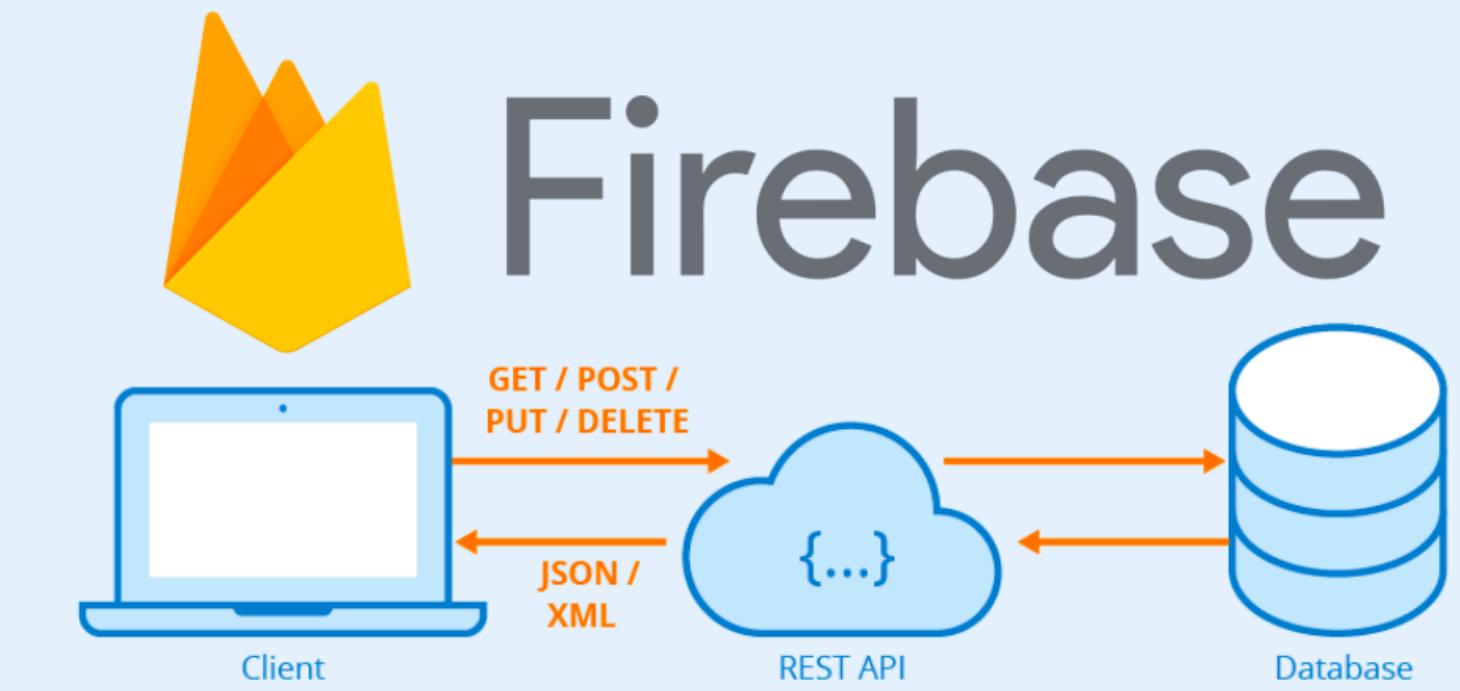
“

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client.

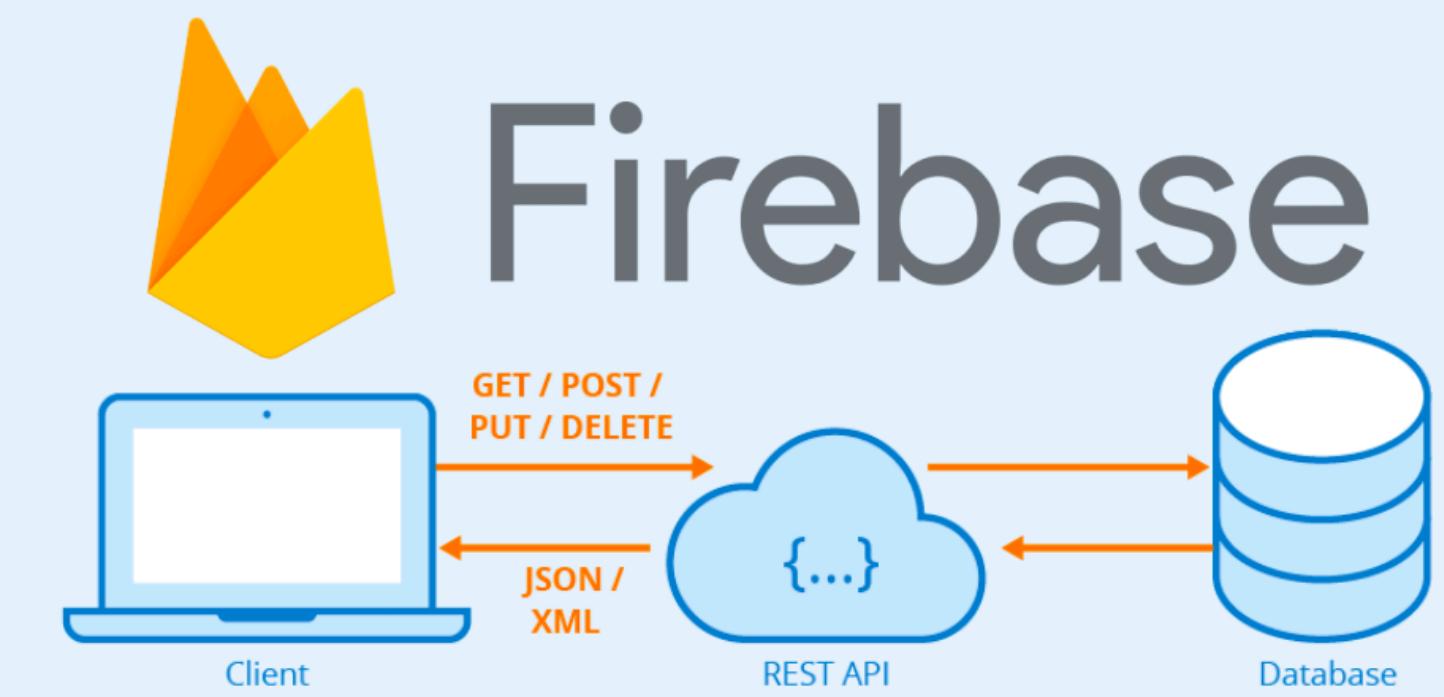
We can use any Firebase Realtime Database URL as a REST endpoint. All we need to do is append .json to the end of the URL and send a request from our favorite HTTPS client.

”

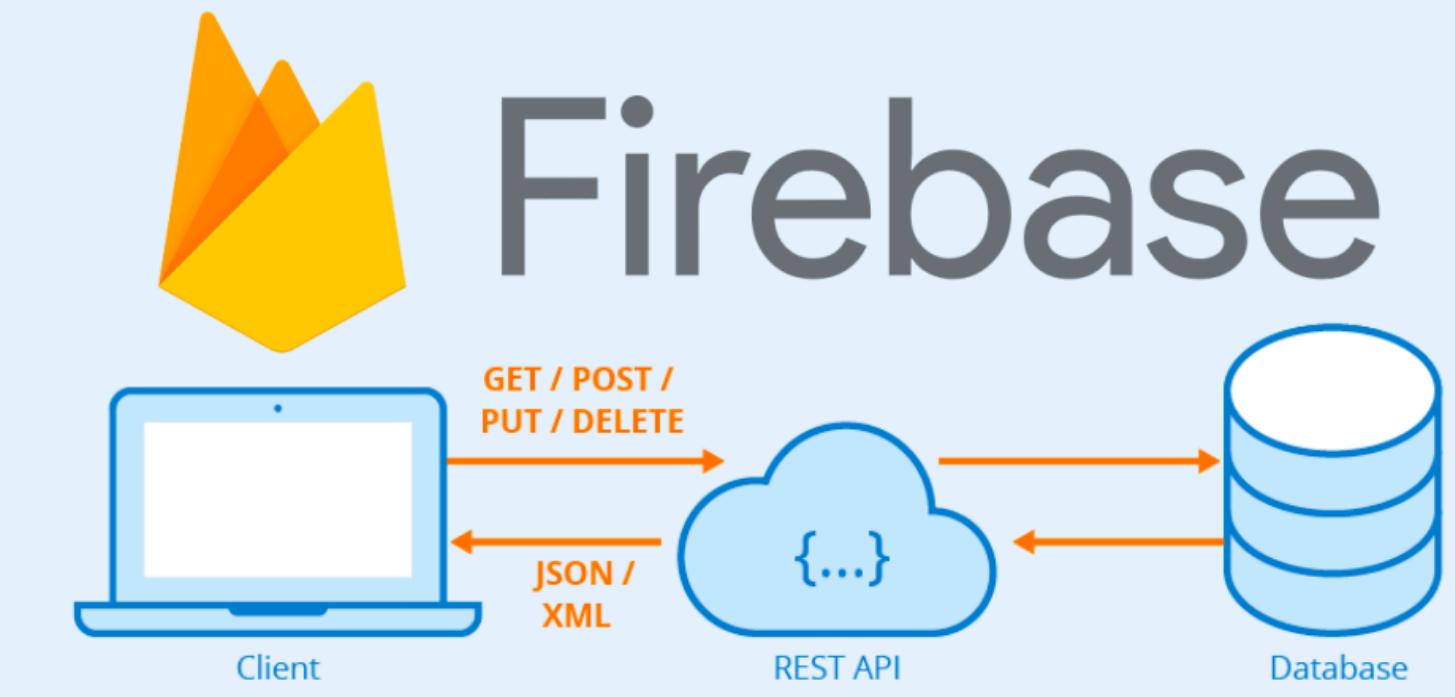
<https://firebase.google.com/docs/database/rest/start>



By following these REST principles, the Firebase Realtime Database REST API provides a simple and consistent way to work with the database. It allows developers to perform common operations using familiar HTTP methods, making it easier to integrate and interact with the database from different programming languages or platforms.



In the context of Firebase Realtime Database REST API, you can perform CRUD operations using HTTP methods like POST, GET, PUT, PATCH, and DELETE to create, read, update, and delete data in the Firebase Realtime Database. This allows you to interact with the database and manipulate data effectively based on your application's needs.





**my firebase doesn't
work. Can you help me?**

**read the
documentation**

But...

**read the
documentation**

How data is structured

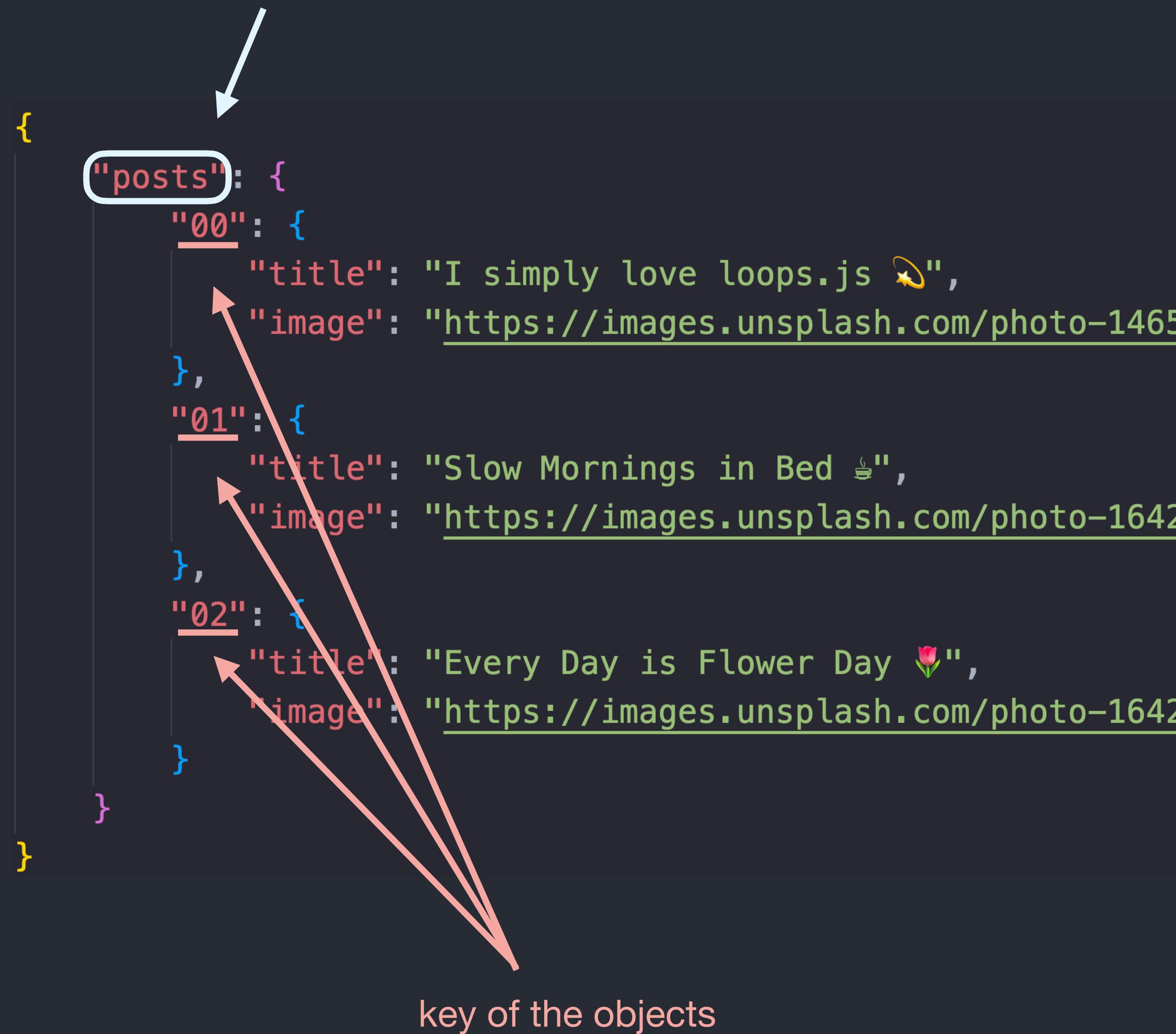
```
{  
  "posts": {  
    "00": {  
      "title": "I simply love loops.js 🌟",  
      "image": "https://images.unsplash.com/photo-1465...  
    },  
    "01": {  
      "title": "Slow Mornings in Bed ☕",  
      "image": "https://images.unsplash.com/photo-1642...  
    },  
    "02": {  
      "title": "Every Day is Flower Day 🌸",  
      "image": "https://images.unsplash.com/photo-1642...  
    }  
  }  
}
```

- One big object with objects.
- All objects have a unique key
- “posts”, “00”, “01” and “02” are all keys.
- “posts” is the key of the collection of post objects (list of posts).
- “00”, “01” and “02” are keys of a post object.
- “title” and “image” are keys inside of every post object. “title” and “image” contains values.
- Remember a property contains a key and a value.

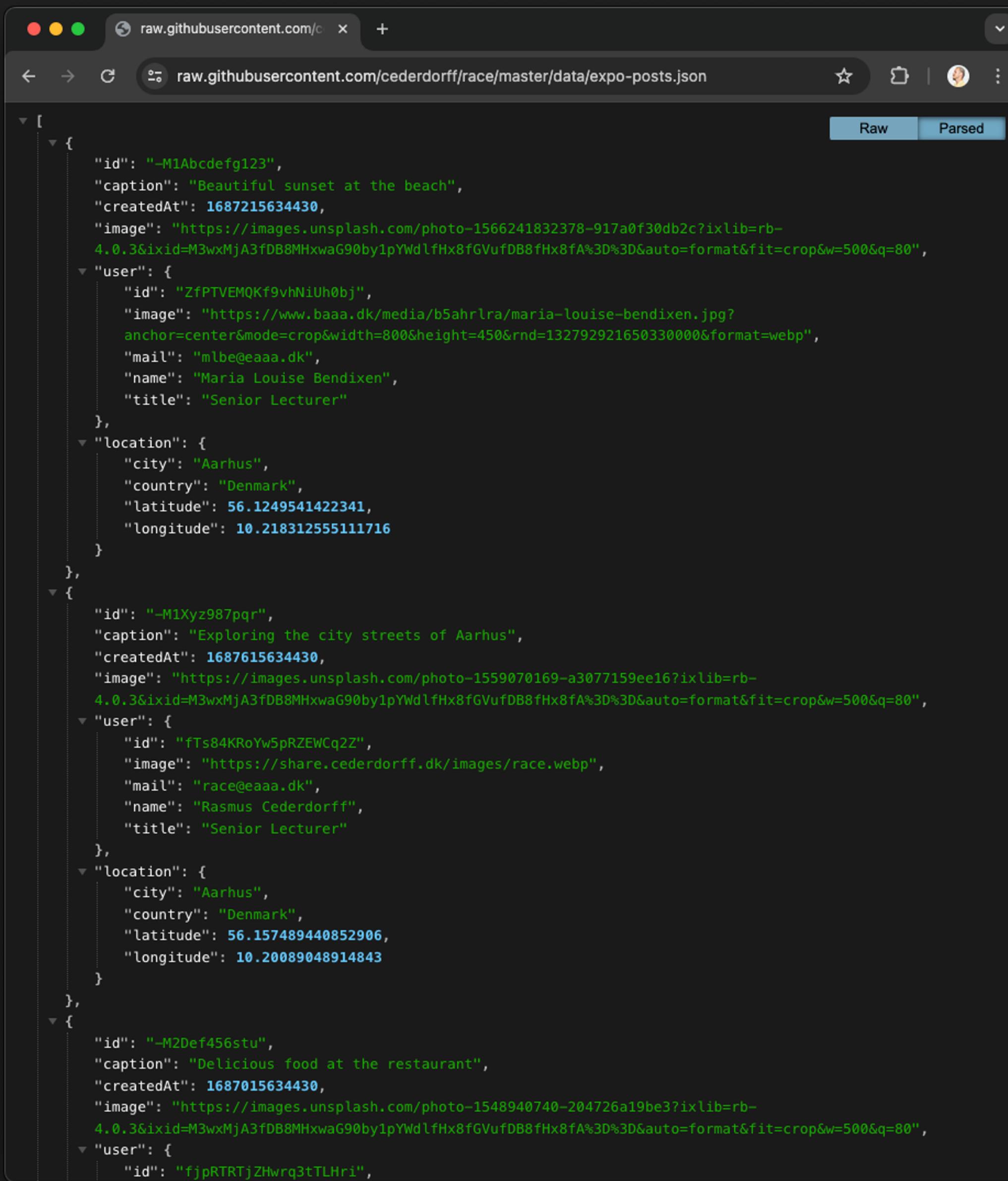
```
{  
  "posts": {  
    "00": {  
      "title": "I simply love loops.js 🌟",  
      "image": "https://images.unsplash.com/photo-1465...  
    },  
    "01": {  
      "title": "Slow Mornings in Bed ☕",  
      "image": "https://images.unsplash.com/photo-1642...  
    },  
    "02": {  
      "title": "Every Day is Flower Day 🌸",  
      "image": "https://images.unsplash.com/photo-1642...  
    }  
  }  
}
```

- Just think of an array of objects (posts array with post objects).
- We are just using an object to hold all the objects instead of an array.
- It's kind of a dictionary. The posts object contains unique prop names (keys) which easily can be "looked up".
- Then every post object (with title and image props) can be returned:
 - `["posts"]["02"]`
 - `["posts"]["00"]`
 - `["posts"]["01"]`
 - `["posts"]["01"]["title"]`

key and name of the collections



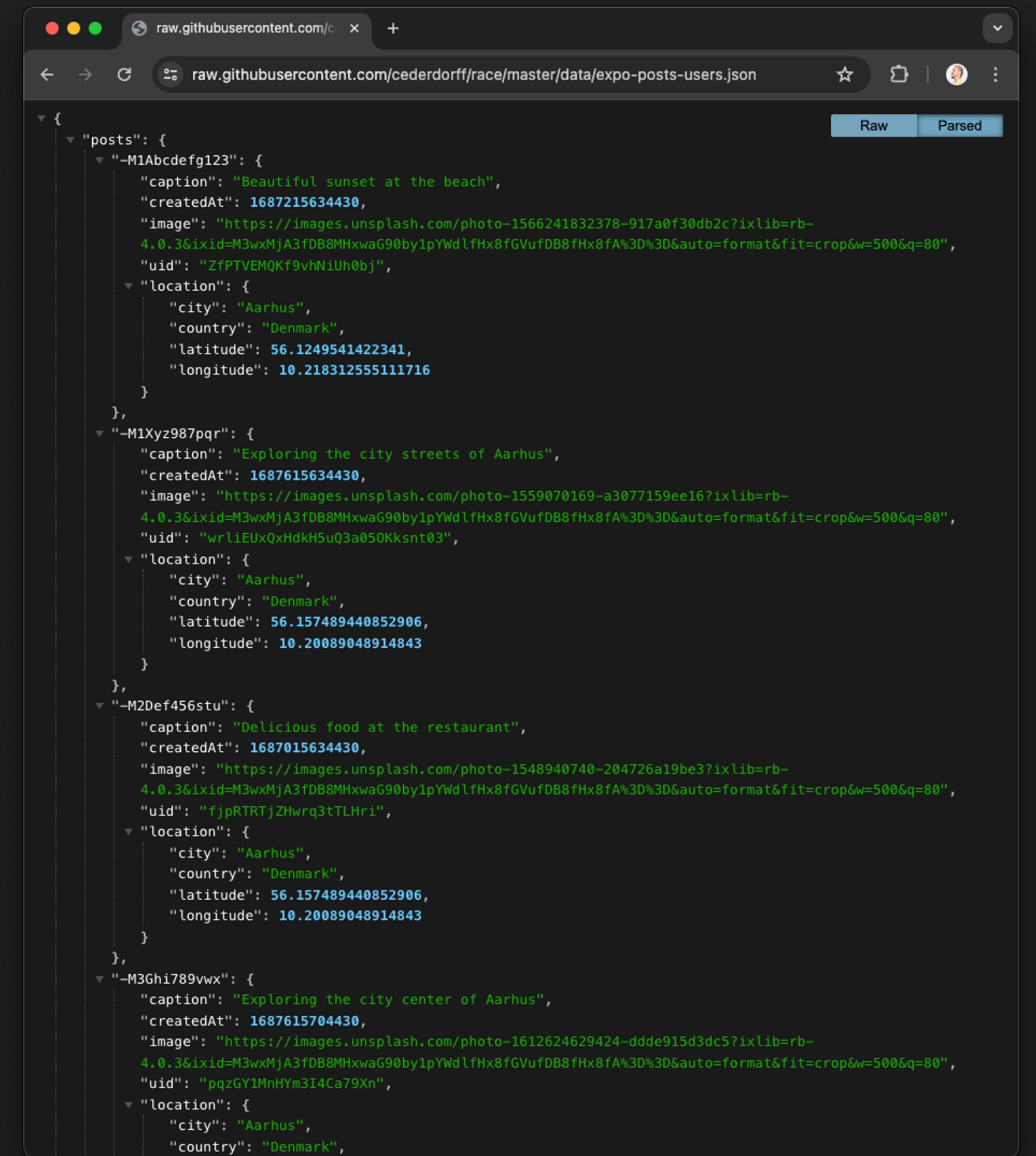
GitHub



A screenshot of a web browser displaying raw JSON data from GitHub at raw.githubusercontent.com/cederdorff/race/master/data/expo-posts.json. The JSON structure represents a collection of posts, each with an ID, caption, creation timestamp, image URL, user information, and location details. The browser interface shows a 'Raw' tab selected.

```
[{"id": "-M1Abcdefg123", "caption": "Beautiful sunset at the beach", "createdAt": 1687215634430, "image": "https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "user": {"id": "ZfPTVEMQKf9vhNiUh0bj", "image": "https://www.baaa.dk/media/b5ahrlra/maria-louise-bendixen.jpg?anchor=center&mode=crop&width=800&height=450&rnd=132792921650330000&format=webp", "mail": "mlbe@aaaa.dk", "name": "Maria Louise Bendixen", "title": "Senior Lecturer"}, "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.1249541422341, "longitude": 10.218312555111716}, {"id": "-M1Xyz987pqr", "caption": "Exploring the city streets of Aarhus", "createdAt": 1687615634430, "image": "https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "user": {"id": "fTs84KR0yW5pRZEWCq2Z", "image": "https://share.cederdorff.dk/images/race.webp", "mail": "race@aaaa.dk", "name": "Rasmus Cederdorff", "title": "Senior Lecturer"}, "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.157489440852906, "longitude": 10.20089048914843}, {"id": "-M2Def456stu", "caption": "Delicious food at the restaurant", "createdAt": 1687015634430, "image": "https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "user": {"id": "fjpRTTjZHwrq3tTLHri", "image": null, "mail": null, "name": null, "title": null}, "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.157489440852906, "longitude": 10.20089048914843}, {"id": "-M3Ghi789vwx", "caption": "Exploring the city center of Aarhus", "createdAt": 1687615704430, "image": "https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "user": {"id": "pqzGY1MnHYM3I4Ca79Xn", "image": null, "mail": null, "name": null, "title": null}, "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.157489440852906, "longitude": 10.20089048914843}], [{"id": "-M1Abcdefg123", "caption": "Beautiful sunset at the beach", "createdAt": 1687215634430, "image": "https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "uid": "ZfPTVEMQKf9vhNiUh0bj", "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.1249541422341, "longitude": 10.218312555111716}, {"id": "-M1Xyz987pqr", "caption": "Exploring the city streets of Aarhus", "createdAt": 1687615634430, "image": "https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "uid": "wrliEuXQxHdkH5uQ3a050Kksnt03", "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.157489440852906, "longitude": 10.20089048914843}, {"id": "-M2Def456stu", "caption": "Delicious food at the restaurant", "createdAt": 1687015634430, "image": "https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "uid": "fjpRTTjZHwrq3tTLHri", "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.157489440852906, "longitude": 10.20089048914843}, {"id": "-M3Ghi789vwx", "caption": "Exploring the city center of Aarhus", "createdAt": 1687615704430, "image": "https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "uid": "pqzGY1MnHYM3I4Ca79Xn", "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.157489440852906, "longitude": 10.20089048914843}]}]
```

Firebase



A screenshot of a web browser displaying raw JSON data from Firebase at raw.githubusercontent.com/cederdorff/race/master/data/expo-posts-users.json. The JSON structure is identical to the GitHub data, representing a collection of posts with their respective users and locations. The browser interface shows a 'Raw' tab selected.

```
[{"posts": {"-M1Abcdefg123": {"caption": "Beautiful sunset at the beach", "createdAt": 1687215634430, "image": "https://images.unsplash.com/photo-1566241832378-917a0f30db2c?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "uid": "ZfPTVEMQKf9vhNiUh0bj", "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.1249541422341, "longitude": 10.218312555111716}}, "-M1Xyz987pqr": {"caption": "Exploring the city streets of Aarhus", "createdAt": 1687615634430, "image": "https://images.unsplash.com/photo-1559070169-a3077159ee16?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "uid": "wrliEuXQxHdkH5uQ3a050Kksnt03", "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.157489440852906, "longitude": 10.20089048914843}}, "-M2Def456stu": {"caption": "Delicious food at the restaurant", "createdAt": 1687015634430, "image": "https://images.unsplash.com/photo-1548940740-204726a19be3?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "uid": "fjpRTTjZHwrq3tTLHri", "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.157489440852906, "longitude": 10.20089048914843}}, "-M3Ghi789vwx": {"caption": "Exploring the city center of Aarhus", "createdAt": 1687615704430, "image": "https://images.unsplash.com/photo-1612624629424-ddde915d3dc5?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1pYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=500&q=80", "uid": "pqzGY1MnHYM3I4Ca79Xn", "location": {"city": "Aarhus", "country": "Denmark", "latitude": 56.157489440852906, "longitude": 10.20089048914843}}]}
```



Code
Every
Day