Remote Databases


kinvey

Table of Contents

Relational Databases

- Represent and store **data** in tables and rows
- Use **S**tructured **Q**uerying **L**anguage (**SQL**)
- Allows you to **link information** from different tables through the use of **foreign keys** (or indexes)

Non Relational Databases

- No-SQL databases
- More **flexibility** and **adaptability**
- Allow us to **store unstructured data** in a single document
- **Additional processing** effort and **more storage** as the document sizes grow

Relational and Non Relational Pros

Relational

- Work with **structured data**
- They support **ACID** transactional consistency and support "**joins**"
- Built-in **data integrity** and a large eco-system
- Relationships in this system have **constraints**
- Limitless **indexing**

Non Relational

- They **scale** out **horizontally**
- Work with **unstructured** and semi-structured **data**
- Schema-free or Schema-on-read options
- **High availability**
- Many are **open source** and so "free"

Backend As a Service

- Solutions that provide **pre-built**, **cloud hosted** components for developing application backends
- **Reduce** the **time** and **complexity** required
- Allow developers to **focus** on **core features** instead of low level tasks
- Types:
  - Cloud BaaS
  - Opensource BaaS

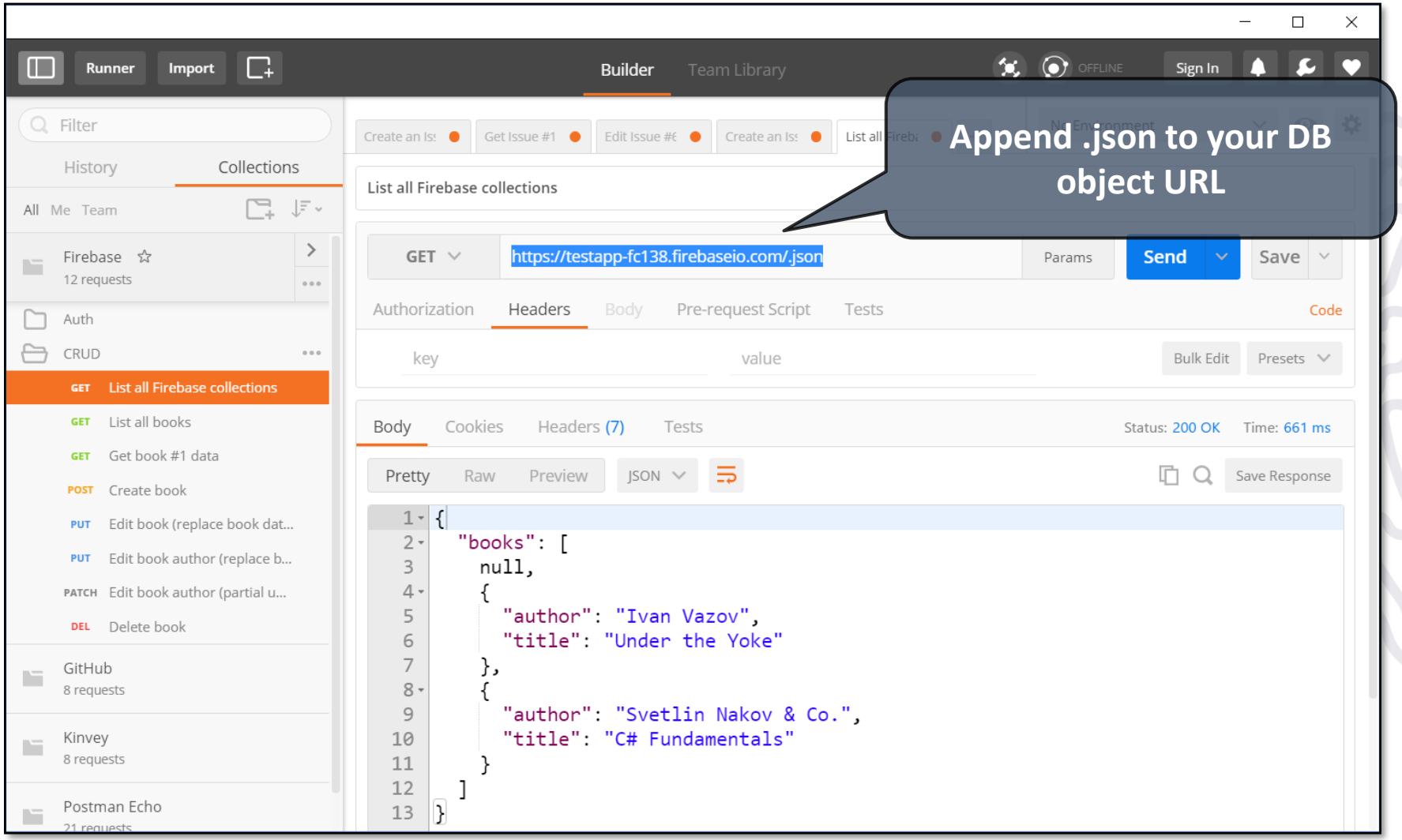
Real-Time Cloud DB and App Platform by Google

Firebase

Firebase

- Mobile and web development platform. It provides:
  - Realtime database
  - Backend as a service
  - JSON-based data structure

Accessing Firebase REST API with Postman



Firebase REST API – CRUD Operations

GET	https://testapp-fc138.firebaseio.com/.json
GET	https://testapp-fc138.firebaseio.com/books.json
GET	https://testapp-fc138.firebaseio.com/books/1.json
GET	https://testapp-fc138.firebaseio.com/books/1/author.json
POST	https://testapp-fc138.firebaseio.com/books.json
Body	{"title":"New title", "author":"New author"}
DELETE	https://testapp-fc138.firebaseio.com/books/6.json

Firebase REST API – CRUD Operations (2)

PUT	https://testapp-fc138.firebaseio.com/books/7.json
Body	{"title":"Edited", "year":1980, "ISBN":"954X"}

PATCH	https://testapp-fc138.firebaseio.com/books/7.json
Body	{"year":1981, "author":"Author Changed"}

PUT	https://testapp-fc138.firebaseio.com/books/7/author.json
Body	"New author was assigned"

DELETE	https://testapp-fc138.firebaseio.com/books/7/author.json

Authentication vs. Authorization

Authentication

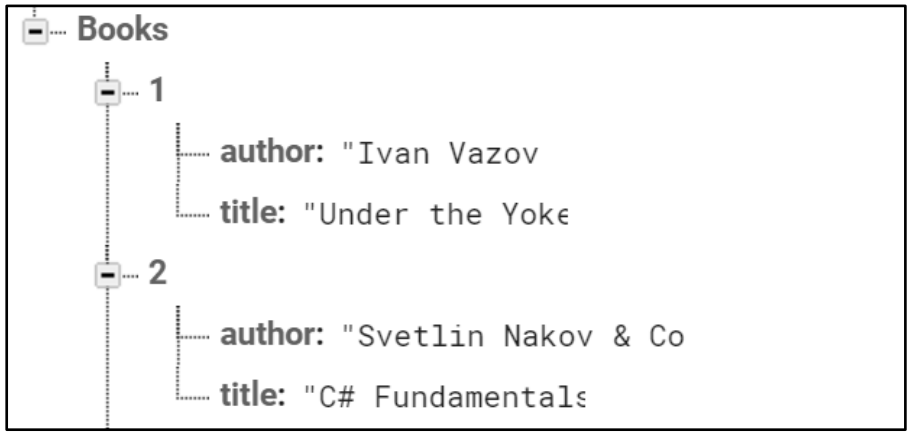
- The process of verifying the identity of a user or computer
- Questions: "**Who are you?**", "**How you prove it? **"
- Credentials can be password, smart card, external token, etc.

Authorization

- The process of determining what a user is permitted to do on a computer or network
- Questions: " **What are you allowed to do?**", "**Can you see this page?**"

Firebase and Postman: All Books

- Open Firebase and create "**TestApp**" project
  - Create the following structure



  - Get all books by AJAX request

Firebase and Postman: Get Book

- Using the recently created project in **Firebase**
  - Get the book with **id: 1**
  - Don't forget the **`.json`** extension at the end

Firebase and Postman: Create Book

- To create a book in our Firebase project
  - Send a "**POST**" request
    - JSON body should be in the following format:

```
{
  "title":"New title",
  "author":"New author"
}
```

Collection-Based Cloud DB (MBaaS)

Kinvey

Kinvey As Back-End

- Mobile Back-End as a Service (mBaaS)
  - Holds your app / mobile **app data** in the cloud
  - Hold **users** and **user data**
  - Users (API for creating an account)
  - Data collections (API for CRUD operations)
  - Files (upload / download / delete)

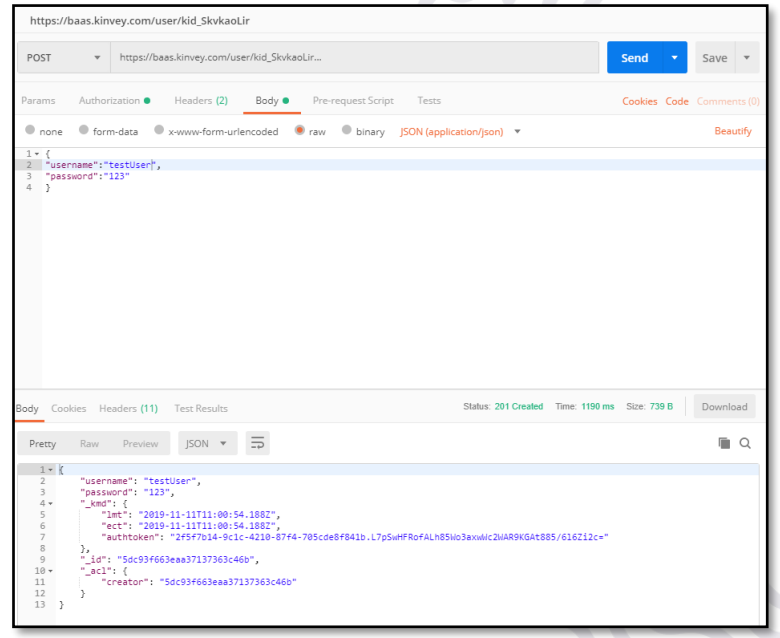
Kinvey and Postman: Register

- URL: https://baas.kinvey.com/user/{app_id}
- Method: **POST**
- Authentication: Kinvey
  - User/pass: **app_key : app_secret**
- Body:

```
{"username":"…", "password":"…"}
```

- Returns: **authtoken**

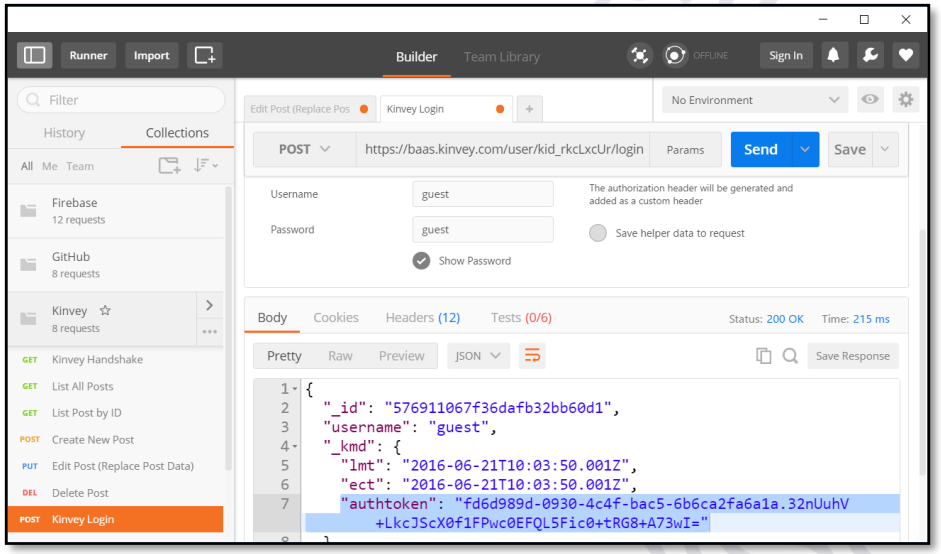
```
"authtoken": "fd6d989d-0930-4c…wI="
```



Kinvey and Postman: Login

- URL: https://baas.kinvey.com/user/{app_id}/login
- Method:
- Authentication: Kinvey
  - User/pass:
- Body:

```
{"username":"…", "password":"…"}
```

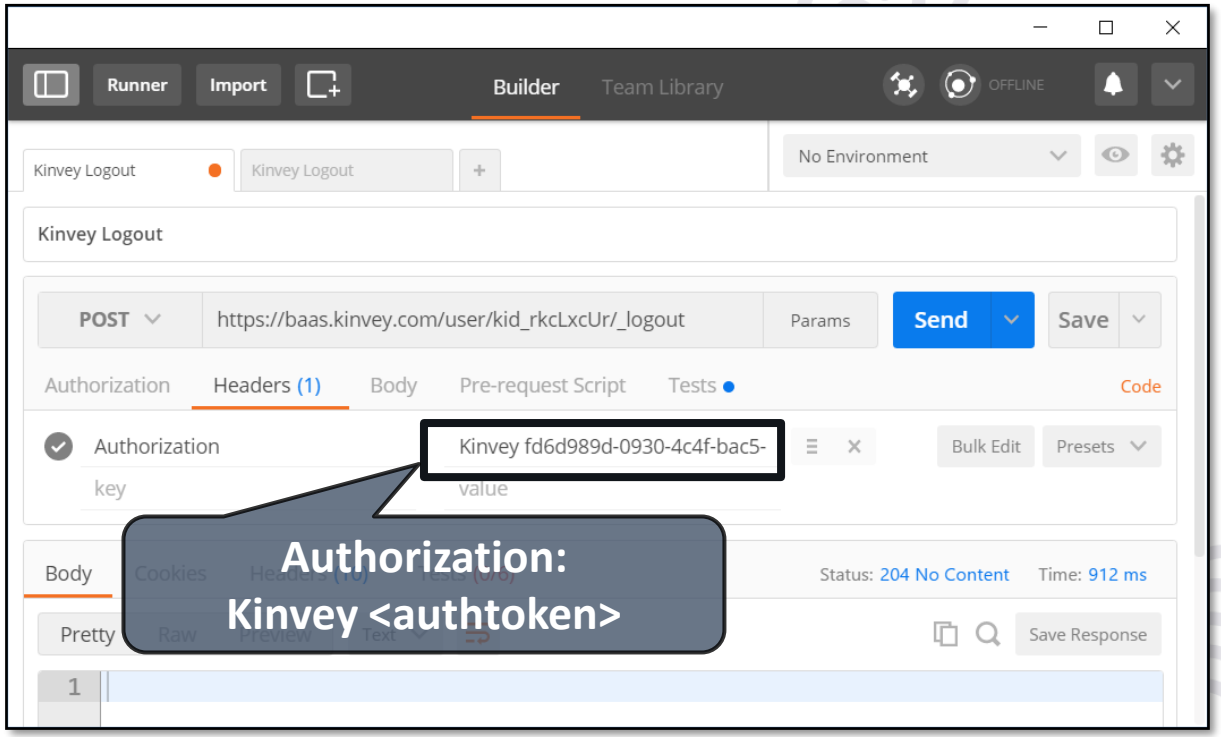


- Returns:

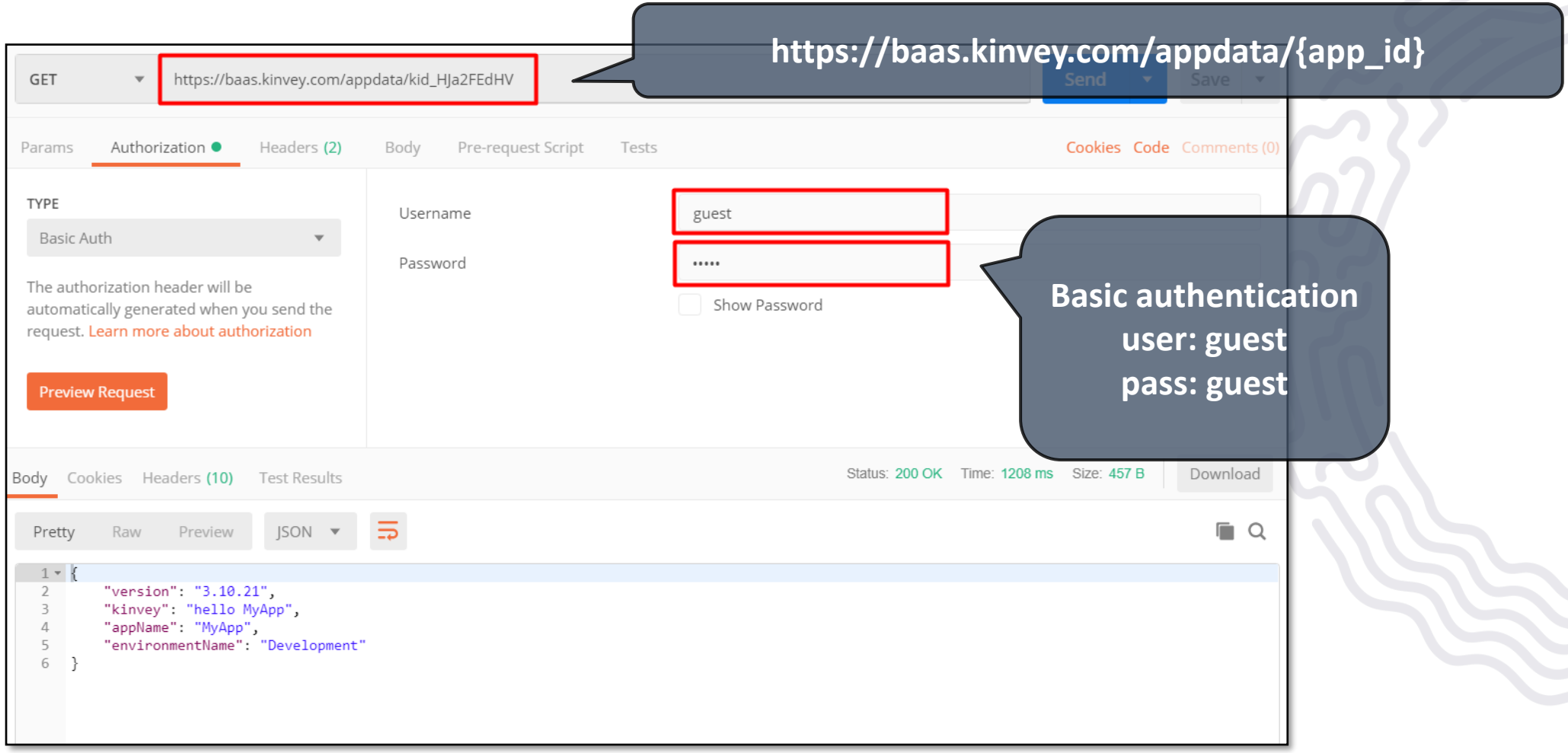
```
"authtoken": "fd6d989d-0930-4c…wI="
```

Kinvey and Postman: Logout

- URL: https://baas.kinvey.com/user/{app_id}/_logout
- Method: **POST**
- Authorization:
  - **Kinvey <authtoken>**
  - Use the token given by the login request
- Returns **204 No Content**

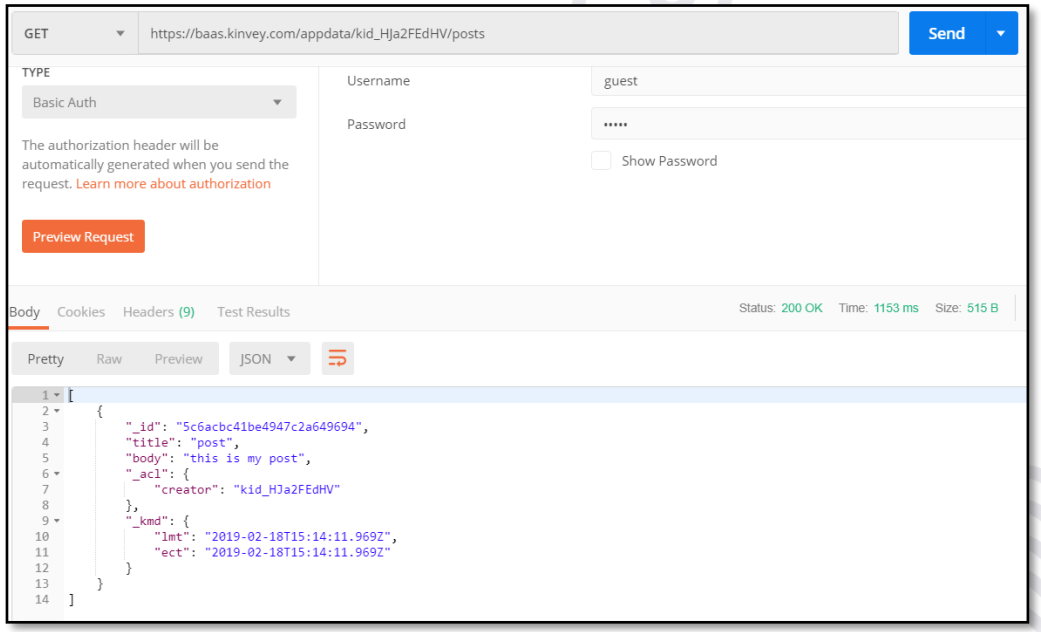


Test Your Backend with Postman



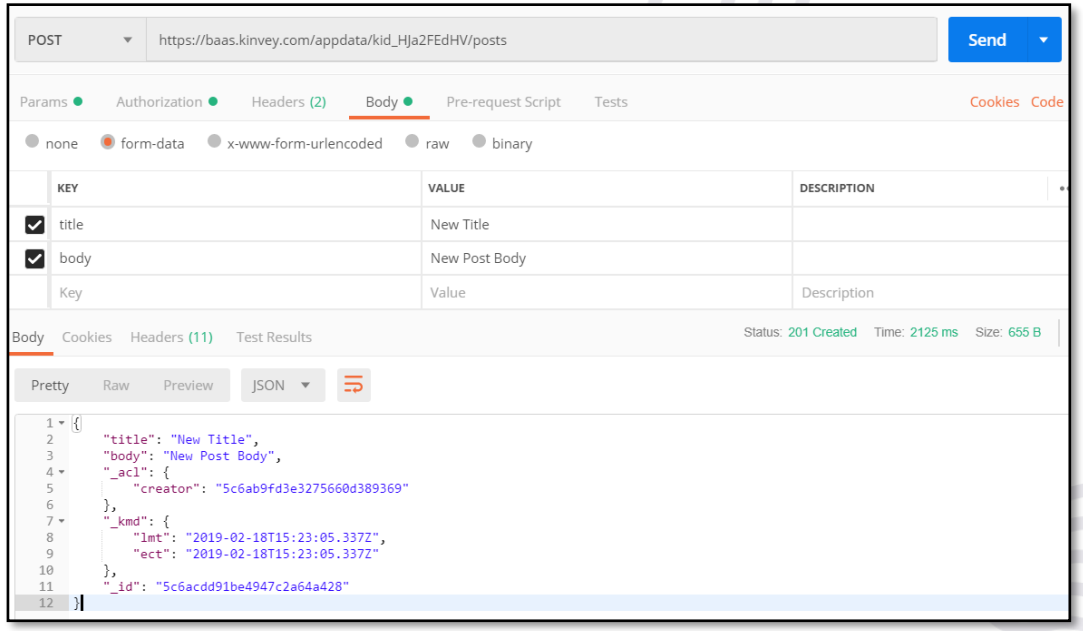
Kinvey and Postman: List All Posts

- URL: https://baas.kinvey.com/appdata/{app_id}/posts
- Method: **GET**
- Authentication: Kinvey + token
  - User: guest
  - Pass: guest



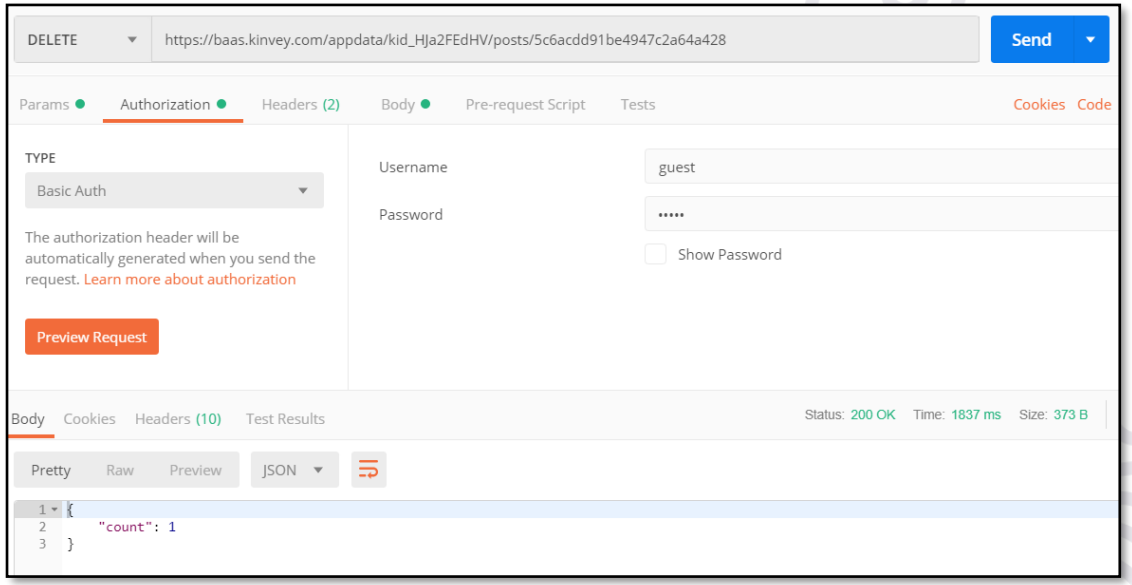
Kinvey and Postman: Create a New Post

- URL: https://baas.kinvey.com/appdata/{app_id}/posts
- Method: **POST**
- Authentication: Kinvey + token
  - User / pass: guest / guest
- Request body
  - title: New Title
  - body: New Post Body



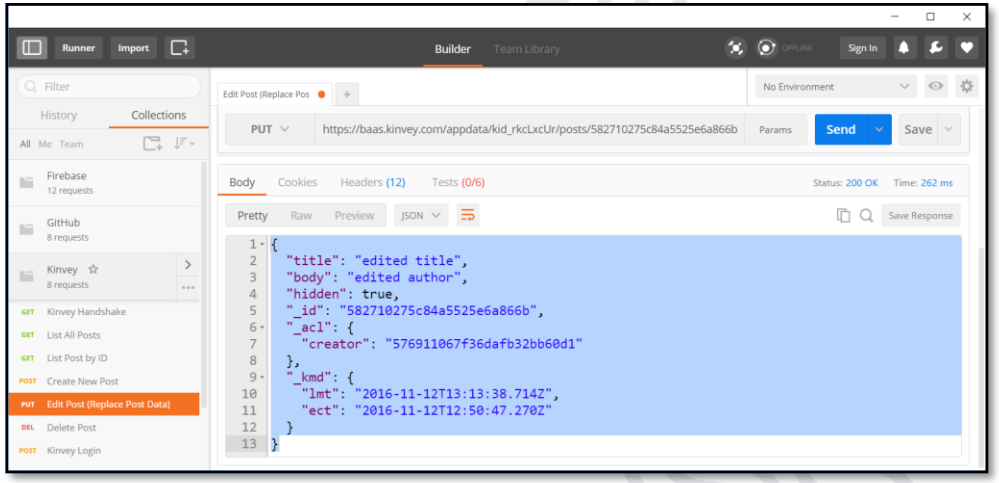
Kinvey and Postman: Delete an Existing Post

- URL: https://baas.kinvey.com/appdata/{app_id}/posts/id
  - Choose an existing post's ID
- Method: **DELETE**
- Authentication: Kinvey + token
  - User / pass: guest / guest
- Body: (empty)



Kinvey and Postman: Edit an Existing Post

- URL: https://baas.kinvey.com/appdata/{app_id}/posts/id
  - Choose an existing post's ID
- Method: **PUT**
- Authentication: Kinvey + token
  - User / pass: guest / guest
- Body (JSON):




```
{"title":"edited title", "body":"edited author", "hidden":true}
```

Live Exercises

Summary

- Relational Databases
- Non Relational Databases
- BaaS
- Firebase is **JSON-based** cloud database
- (mBaaS) with **REST API**
- Kinvey is **collection-based** could database (mBaaS)

Questions?

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © Kingsland University – https://kingslanduniversity.com


KINGSLAND
UNIVERSITY
THANK YOU