

REST Services

HTTP, RESTful Web Services, HTTP and REST Tools: Postman, Fiddler



POSTMAN



SoftUni Team
Technical Trainers



SoftUni
Foundation



Software University

<http://softuni.bg>

Table of Contents

1. The HTTP Protocol
2. HTTP Developer Tools
3. REST and RESTful Services
4. Accessing the GitHub API
5. Using Firebase BaaS
6. Using Kinvey mBaaS



Have a Question?

sli.do

#JS-CORE

- HTTP (**H**yper **T**ext **T**ransfer **P**rotocol)
 - Text-based client-server protocol for the Internet
 - For transferring Web resources (HTML files, images, styles, etc.)
 - Request-response based



Web Client

HTTP request









Web Server

HTTP response

HTTP Request Methods

HTTP defines **methods** to indicate the desired action to be performed on the identified resource

Method		Description
GET		Retrieve / load a resource
POST		Create / store a resource
PUT		Update a resource
DELETE		Delete (remove) a resource
PATCH		Update resource partially
HEAD		Retrieve the resource's headers
OPTIONS		Returns the HTTP methods that the server supports for the specified URL

HTTP GET Request - Example

GET /users/testnakov/repos **HTTP/1.1**

HTTP request line

Host: api.github.com

Accept: */*

HTTP headers

Accept-Language: en

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/54.0.2840.71 Safari/537.36

Connection: Keep-Alive

Cache-Control: no-cache

<CRLF>

The request **body** is empty

HTTP POST Request - Example

POST /repos/testnakov/test-nakov-repo/issues **HTTP/1.1**

Host: api.github.com

Accept: */*

Accept-Language: en

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0 (compatible;MSIE 6.0; Windows NT 5.0)

Connection: Keep-Alive

Cache-Control: no-cache

<CRLF>

{"title": "Found a bug",

"body": "I'm having a problem with this.",

"labels": ["bug", "minor"]}

<CRLF>

HTTP request line

HTTP headers

The request **body** holds
the submitted data

HTTP Response - Example

HTTP/1.1 200 OK

HTTP response status **line**

Date: Fri, 11 Nov 2016 16:09:18 GMT+2

Server: Apache/2.2.14 (Linux)

Accept-Ranges: bytes

Content-Length: 84

Content-Type: text/html

HTTP response
headers

<CRLF>

<html>

<head><title>Test</title></head>

<body>Test HTML page.</body>

HTTP response
body

</html>

HTTP Response Status Codes

Status Code	Action	Description
200	OK	Successfully retrieved resource
201	Created	A new resource was created
204	No Content	Request has nothing to return
301 / 302	Moved	Moved to another location (redirect)
400	Bad Request	Invalid request / syntax error
401 / 403	Unauthorized	Authentication failed / Access denied
404	Not Found	Invalid resource
409	Conflict	Conflict was detected, e.g. duplicated email
500 / 503	Server Error	Internal server error / Service unavailable

Content-Type and Disposition

The **Content-Type** / **Content-Disposition** headers specify how the HTTP request / response body should be processed

JSON-encoded data

Content-Type: **application/json**

UTF-8 encoded HTML page. Will be shown in the browser

Content-Type: **text/html**; charset=utf-8

Content-Type: **application/pdf**

Content-Disposition: attachment;
filename="Financial-Report-April-2016.pdf"

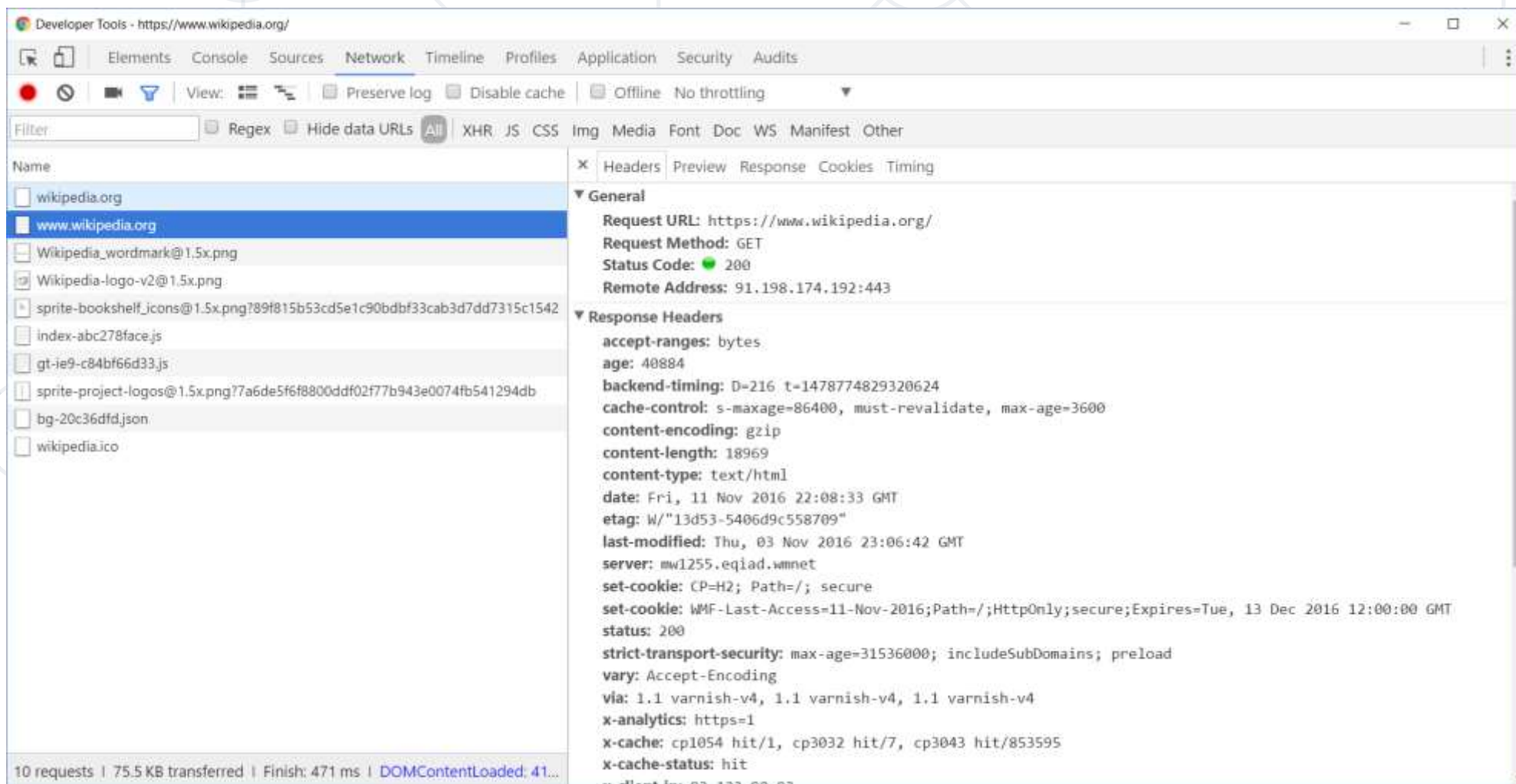
This will download a PDF file named
Financial-Report-April-2016.pdf



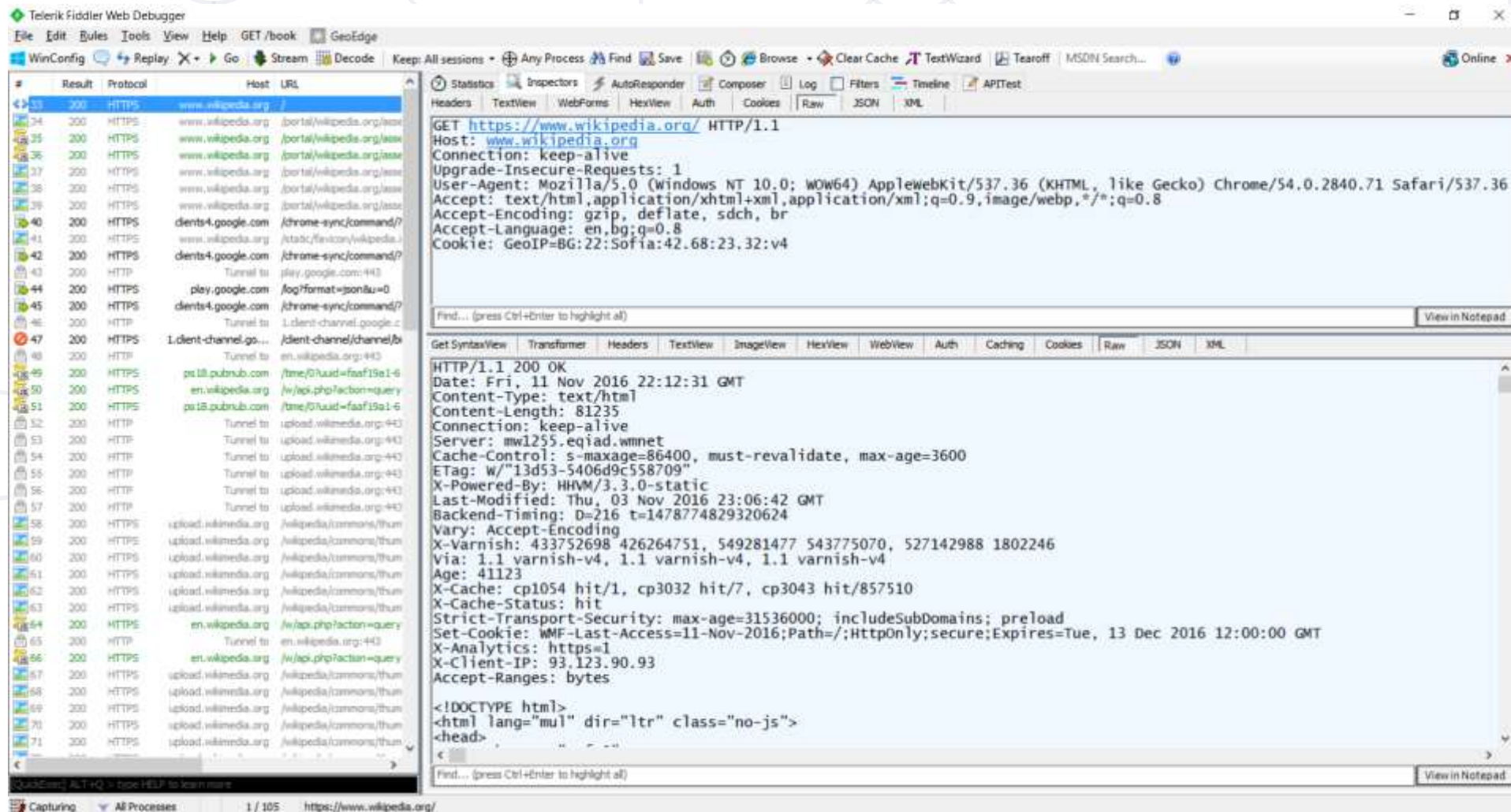
HTTP Developer Tools

Chrome Dev Tools, Fiddler, Postman

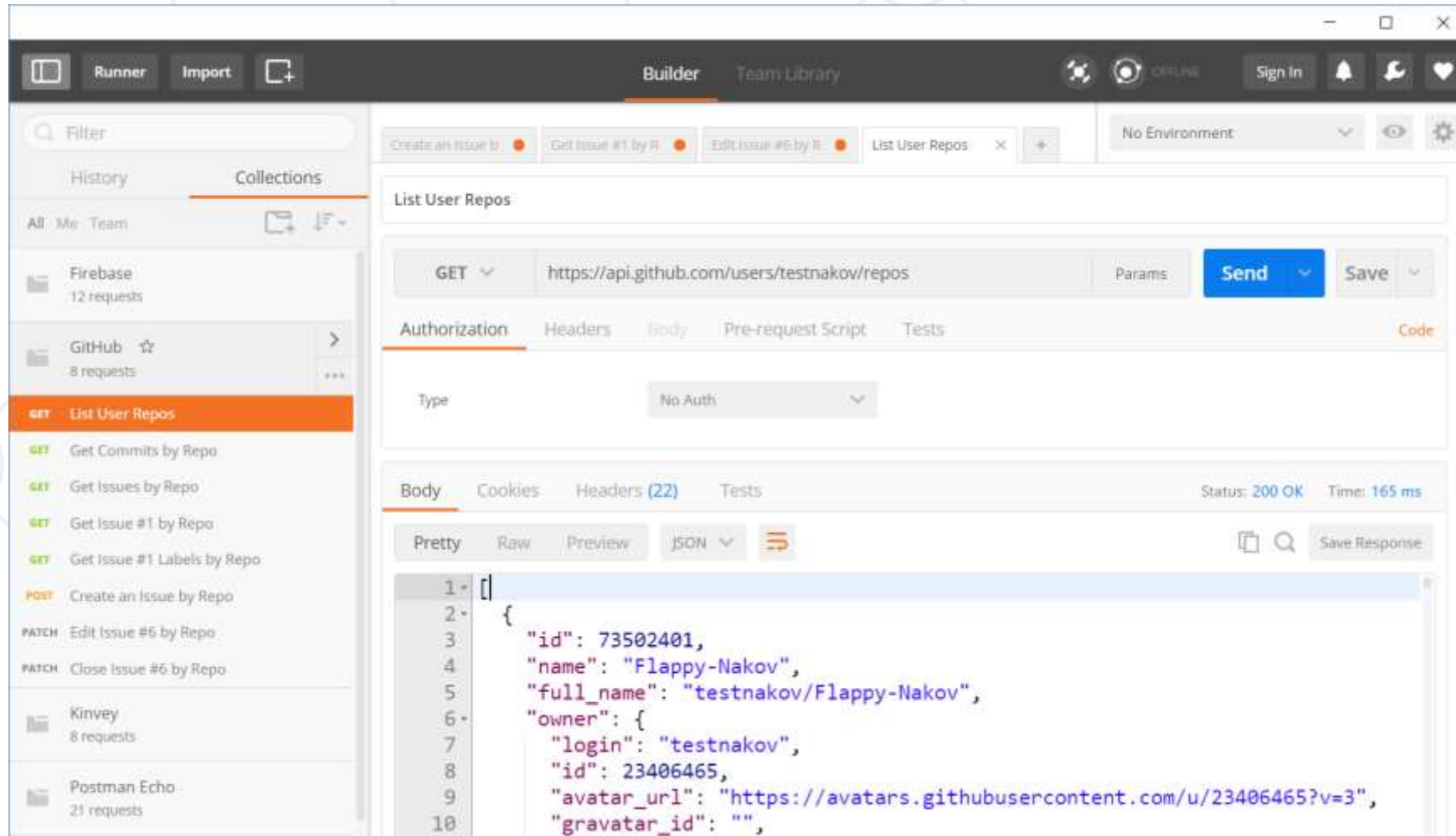
Chrome Developer Tools



[Read more about Chrome Developer Tools](#)

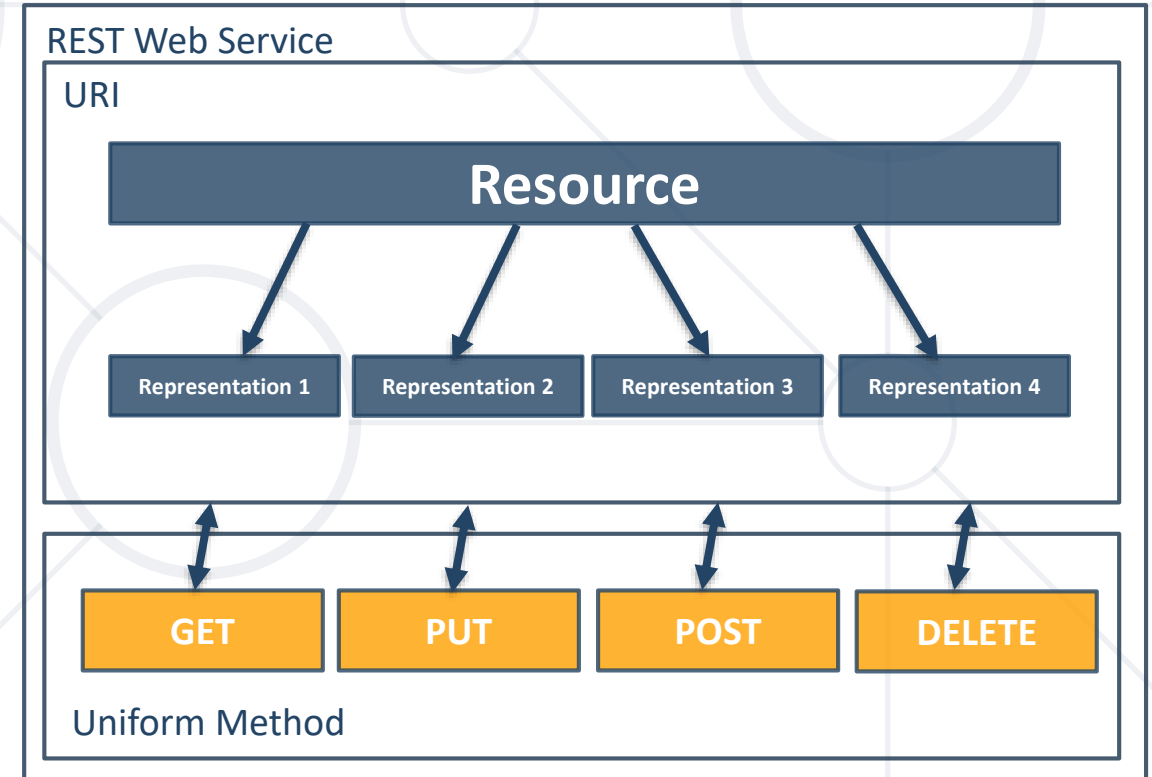


[Read more about Fiddler](#)



[Read more about Postman REST Client](#)

- **Re**presentational **S**tate **T**ransfer (**REST**)
 - Architecture for **client-server communication** over HTTP
 - Resources have **URI** (address)
 - Can be **created/retrieved/modified/deleted**/etc.
- RESTful API/RESTful Service
 - Provides access to **server-side resources** via **HTTP** and **REST**



REST defines **6 architectural constraints** which make any web service - a true RESTful API

- Client-server architecture
- Statelessness
- Cacheable
- Layered system
- Code on demand (optional)
- Uniform interface



[Read more about REST Architectural Constraints](#)

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?**

REST and RESTful Services - Example

- Create a new post

POST	http://some-service.org/api/posts
------	---

- Get all posts / specific post

GET	http://some-service.org/api/posts
-----	---

GET	http://some-service.org/api/posts/17
-----	---

- Delete existing post

DELETE	http://some-service.org/api/posts/17
--------	---

- Replace / modify existing post

PUT/PATCH	http://some-service.org/api/posts/17
-----------	---



GitHub REST API

Accessing GitHub through HTTP

- List user's all public repositories:

GET	https://api.github.com/users/testnakov/repos
-----	---

- Get all commits from a public repository:

GET	https://api.github.com/repos/testnakov/softuniada-2016/commits
-----	---

- Get all issues/issue #1 from a public repository

GET	/repos/testnakov/test-nakov-repo/issues
-----	---

GET	/repos/testnakov/test-nakov-repo/issues/1
-----	---

- Get all labels for certain issue from a public repository:

GET	https://api.github.com/repos/testnakov/test-nakov-repo/issues/1/labels
-----	---

- Create a new issue to certain repository (with authentication)

POST	https://api.github.com/repos/testnakov/test-nakov-repo/issues
Headers	Authorization: Basic base64(user:pass)
Body	<pre>{"title": "Found a bug", "body": "I'm having a problem with this."}</pre>



Firebase

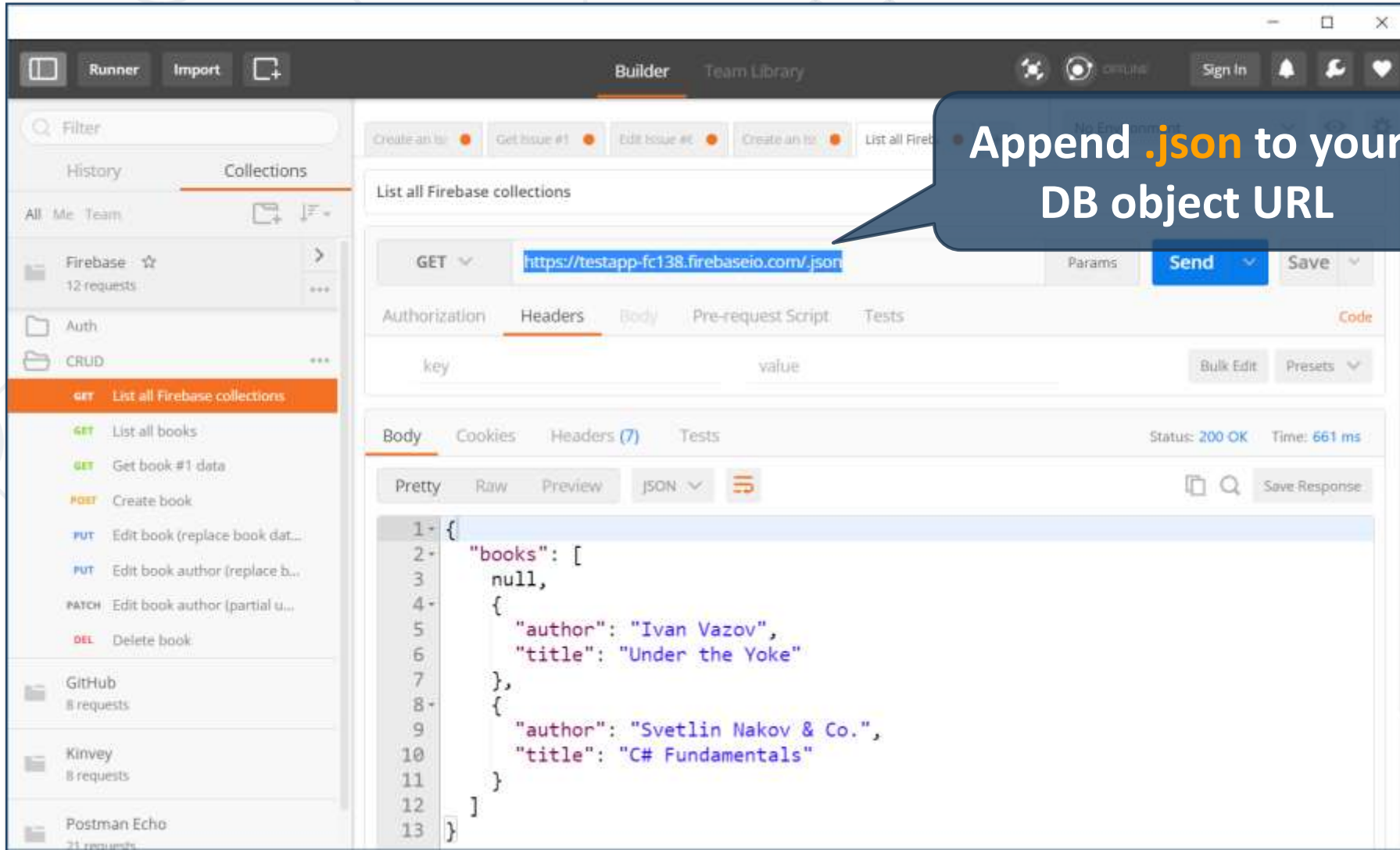
Real-Time Cloud DB and App Platform by Google

Mobile and web development platform. It provides:

- Realtime database
- Backend as a service
- JSON-based data structure



Accessing Firebase REST API with Postman



Append **.json** to your DB object URL

GET `https://testapp-fc138.firebaseio.com/.json`

Authorization Headers Body Pre-request Script Tests

key value Bulk Edit Presets

Body Cookies Headers (7) Tests Status: 200 OK Time: 661 ms

Pretty Raw Preview JSON

```
1 {
2   "books": [
3     null,
4     {
5       "author": "Ivan Vazov",
6       "title": "Under the Yoke"
7     },
8     {
9       "author": "Svetlin Nakov & Co.",
10      "title": "C# Fundamentals"
11    }
12  ]
13 }
```




Firebase

Firebase Demo

Firestore 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	<code>{"title": "New title", "author": "New author"}</code>
DELETE	https://testapp-fc138.firebaseio.com/books/6.json

Firestore REST API - CRUD Operations (2)

PUT	https://testapp-fc138.firebaseio.com/books/7.json
Body	<code>{"title": "Edited", "year": 1980, "ISBN": "954X"}</code>
PATCH	https://testapp-fc138.firebaseio.com/books/7.json
Body	<code>{"year": 1981, "author": "Author Changed"}</code>
PUT	https://testapp-fc138.firebaseio.com/books/7/author.json
Body	<code>"New author was assigned"</code>
DELETE	https://testapp-fc138.firebaseio.com/books/7/author.json

Check your solution here: <https://judge.softuni.bg/Contests/1567/Lab-HTTP-and-REST>



Kinvey
Collection-Based Cloud DB (mBaaS)

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 Demo

Test Your Backend with Postman



The screenshot shows the Postman interface for a GET request. The URL bar is highlighted with a red box and a callout bubble containing the URL: `https://baas.kinvey.com/appdata/{app_id}`. The Authorization tab is selected, showing Basic Auth with the username `guest` and password `.....` (masked). A callout bubble explains: **Basic authentication**
user: guest
pass: guest. The response is shown in the bottom panel, displaying a JSON object with the following fields: `version`, `kinvey`, `appName`, and `environmentName`. The status is 200 OK.

GET `https://baas.kinvey.com/appdata/kid_Hja2FEeHV`

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Comments (0)

TYPE
Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Preview Request

Username: `guest`
Password: `.....`
☐ Show Password

Status: 200 OK Time: 1208 ms Size: 407 B

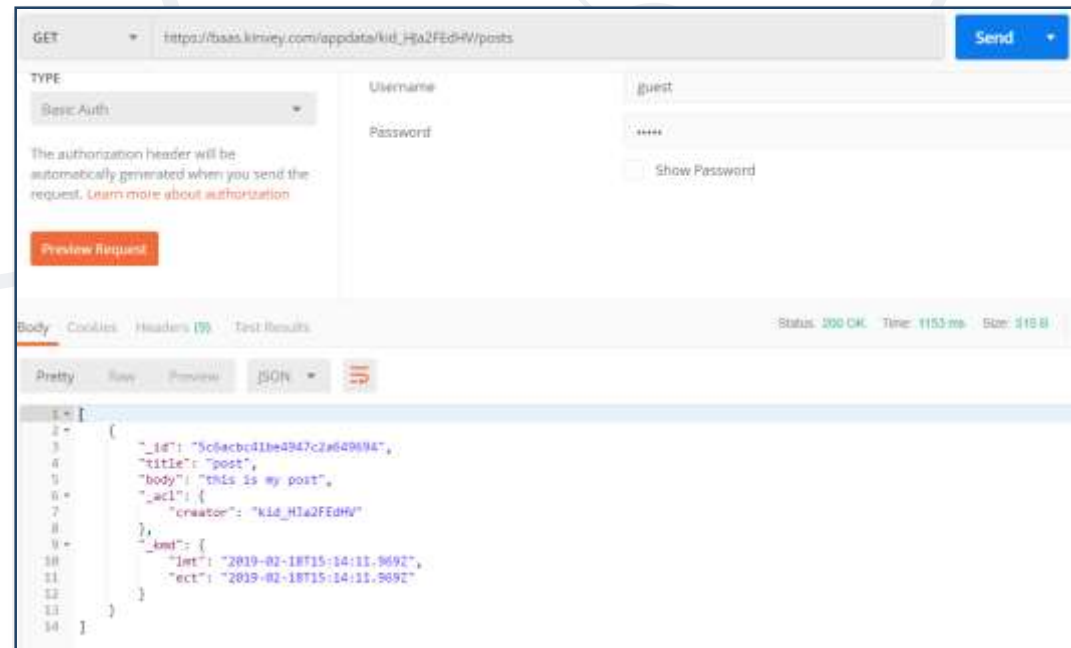
Pretty Raw Preview JSON

```
1 {  
2   "version": "3.10.21",  
3   "kinvey": "hello MyApp",  
4   "appName": "MyApp",  
5   "environmentName": "Development"  
6 }
```

Check your solution here: <https://judge.softuni.bg/Contests/1567/Lab-HTTP-and-REST>

Kinvey and Postman: List All Posts

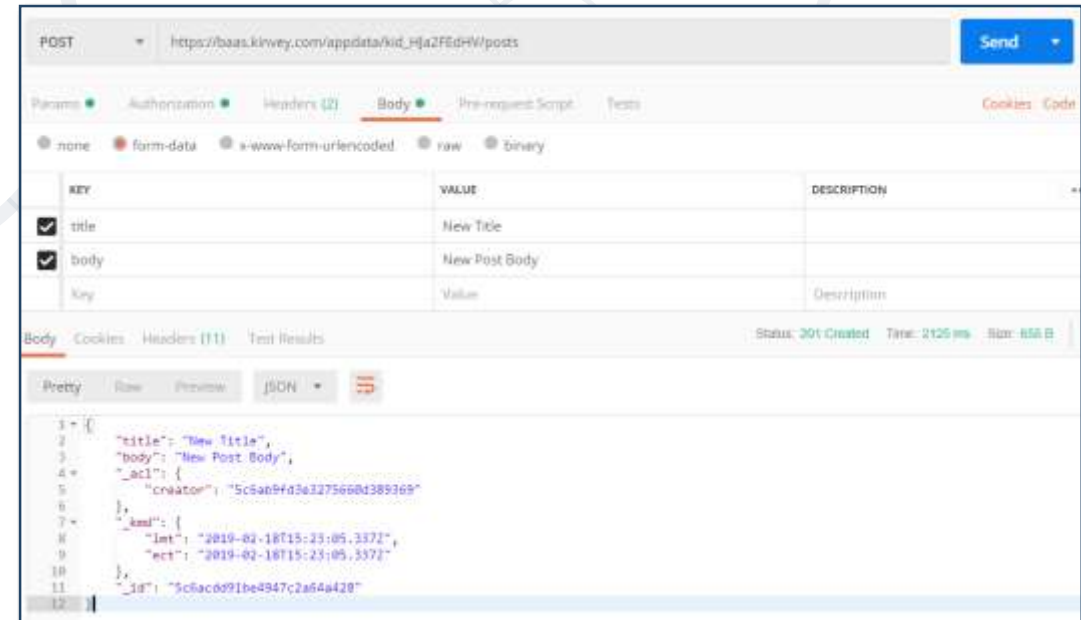
- URL: https://baas.kinvey.com/appdata/{app_id}/posts
- Method: **GET**
- Authentication: Basic
 - User: guest
 - Pass: guest



Check your solution here: <https://judge.softuni.bg/Contests/1567/Lab-HTTP-and-REST>

Kinvey and Postman: Create a New Post

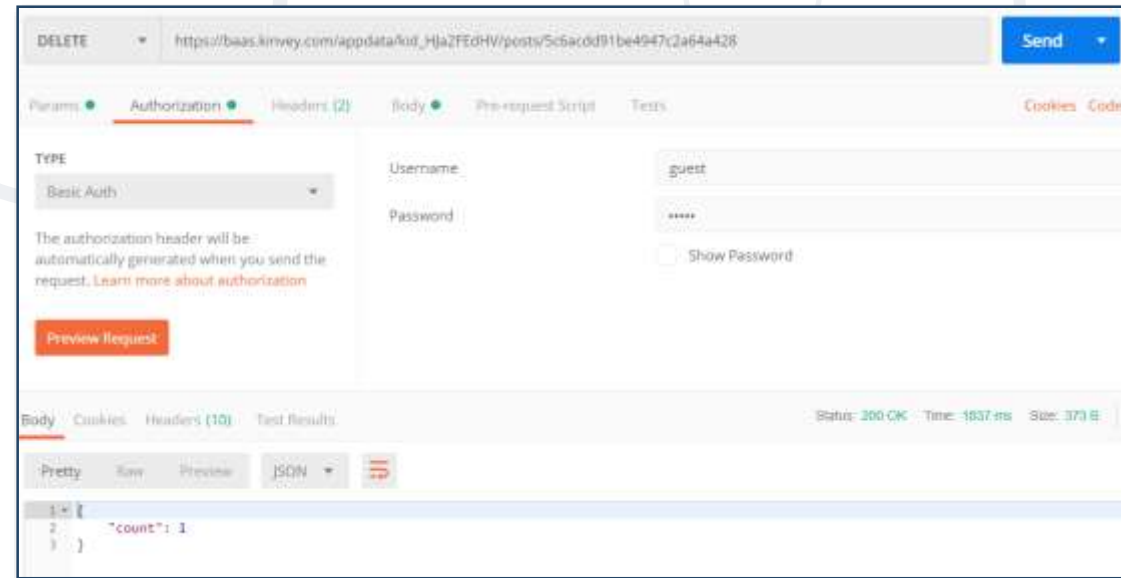
- URL: https://baas.kinvey.com/appdata/{app_id}/posts
- Method: **POST**
- Authentication: Basic
 - User / pass: guest / guest
- Request body
 - title: New Title
 - body: New Post Body



Check your solution here: <https://judge.softuni.bg/Contests/1567/Lab-HTTP-and-REST>

Kinvey and Postman: Delete an Existing Post

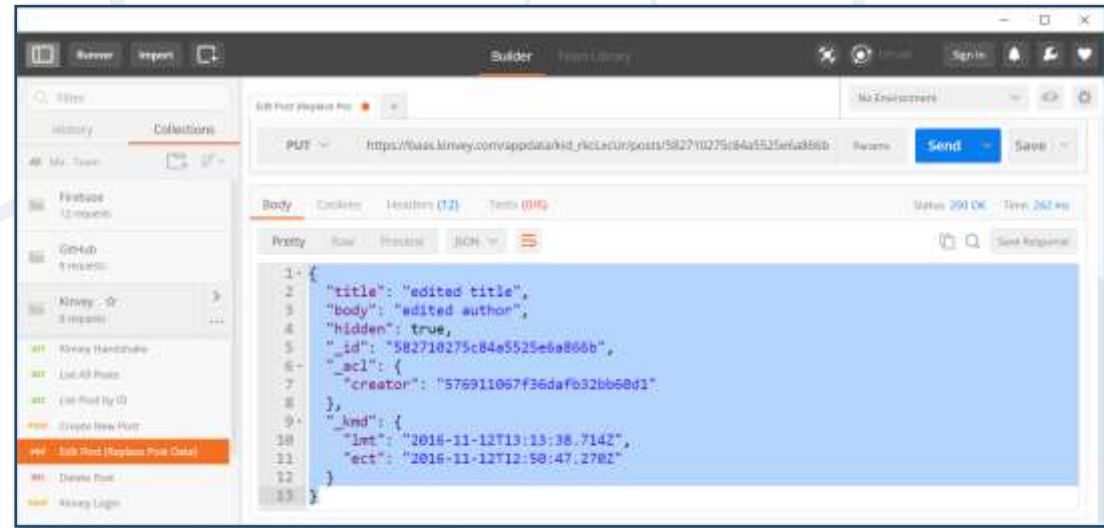
- URL: https://baas.kinvey.com/appdata/{app_id}/posts/id
 - Choose an existing post's ID
- Method: **DELETE**
- Authentication: Basic
 - User / pass:
guest / guest
- Body: (empty)



Check your solution here: <https://judge.softuni.bg/Contests/1567/Lab-HTTP-and-REST>

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: Basic
 - User / pass: guest / guest
- Body (JSON):



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

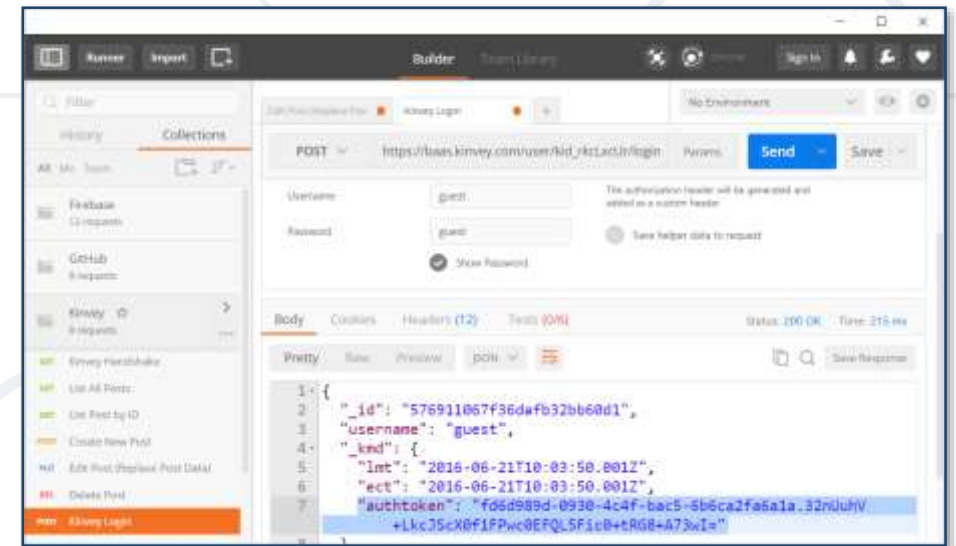
Check your solution here: <https://judge.softuni.bg/Contests/1567/Lab-HTTP-and-REST>

Kinvey and Postman: Login

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

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

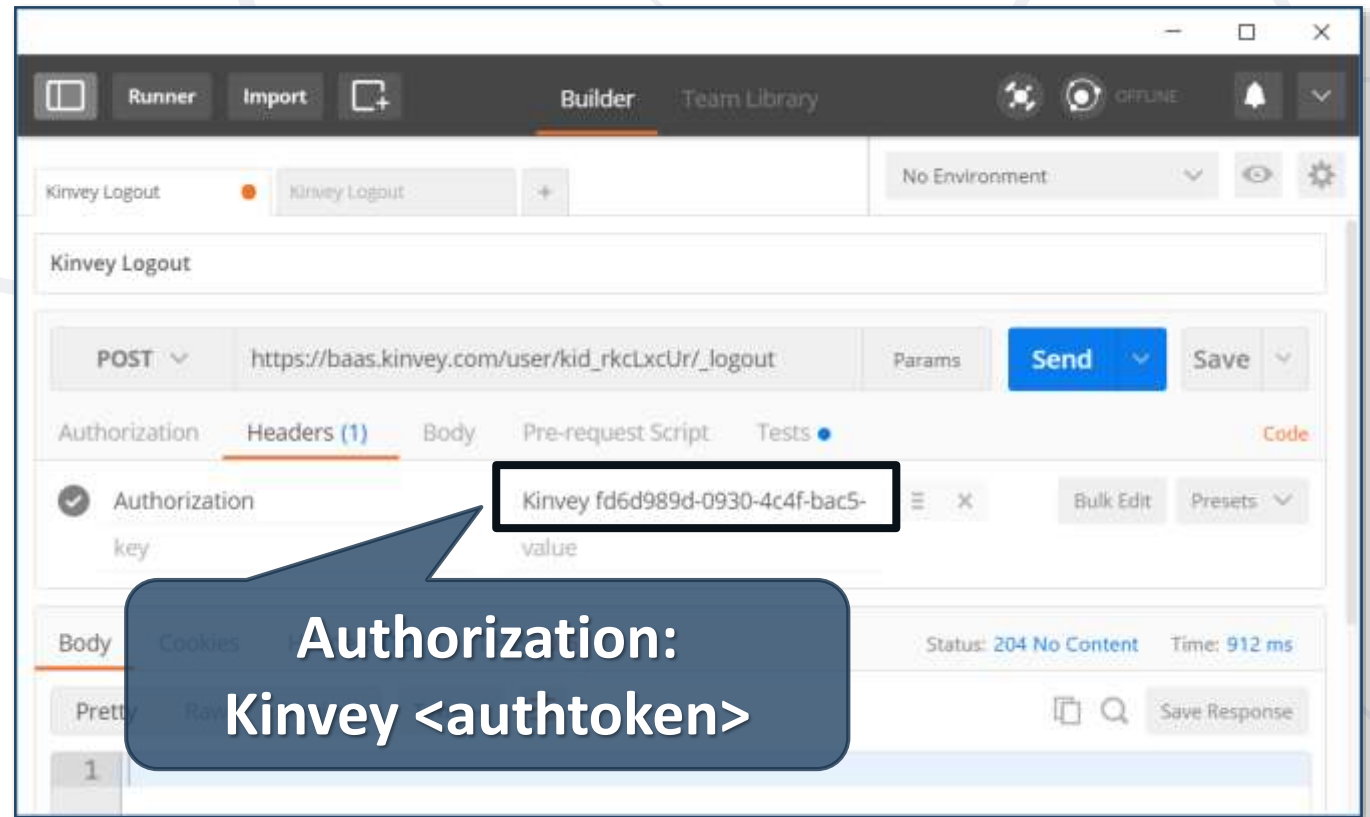
- Returns: **authtoken** `"authtoken": "fd6d989d-0930-4c...wI="`



Check your solution here: <https://judge.softuni.bg/Contests/1567/Lab-HTTP-and-REST>

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**



- **HTTP** is text-based request-response protocol
- **REST** uses **GET, POST, PUT, PATCH, DELETE**
- **RESTful** services address resources by URL
 - Provide **CRUD** operations over HTTP
- Firebase is **JSON-based** cloud database (mBaaS) with **REST API**
- Kinvey is **collection-based** cloud database (mBaaS)



Questions?



SoftUni



**Software
University**



**SoftUni
Svetlina**



**SoftUni
Creative**



**SoftUni
Digital**



**SoftUni
Foundation**



**SoftUni
Kids**

SoftUni Diamond Partners



XSsoftware



SBTech



telenor



SoftwareGroup
doing it right

NETPEAK



SmartIT



Postbank

Решения за твоето утре

**SUPER
HOSTING
.BG**

INDEAVR

Serving the high achievers



INFRAGISTICS®

LIEBHERR



aeternity



SoftUni Organizational Partners



OneBit
SOFTWARE



WORLD
OF
MYTHS

Trainings @ Software University (SoftUni)

- Software University - High-Quality Education and Employment Opportunities
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg



- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

