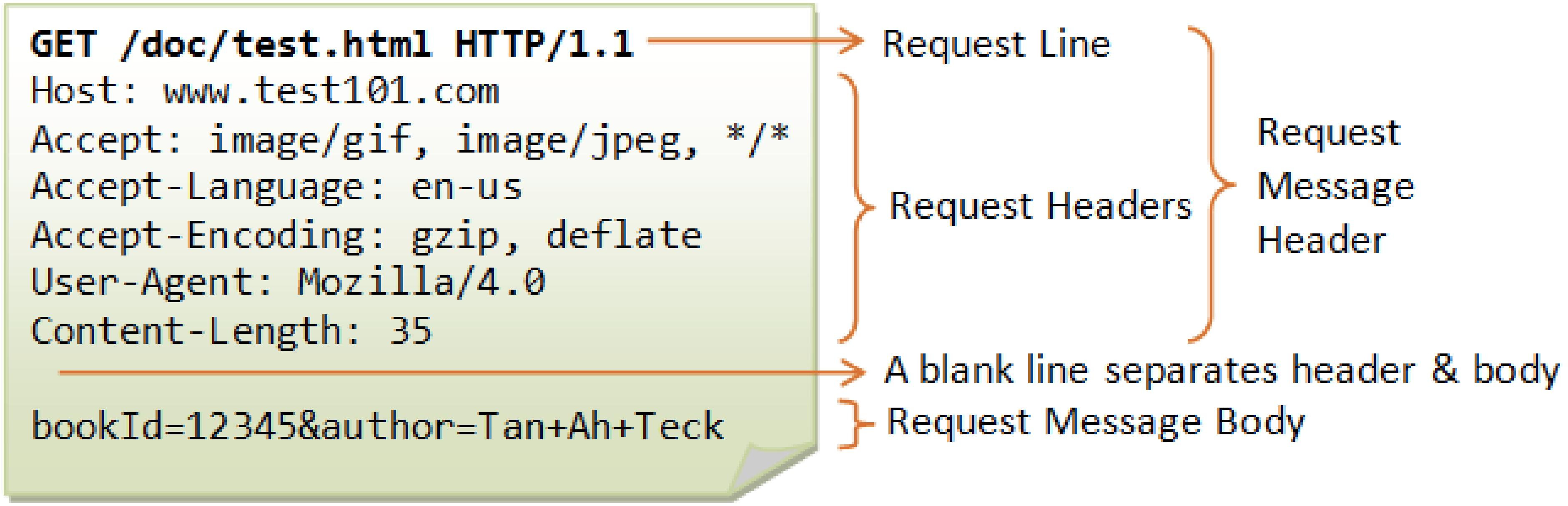
HTTP & RESTful Services

27/05/2024

HTTP request

axess academy

- An HTTP client sends an HTTP request to a server in the form of a request message which includes following format:
  - A Request Line
  - Zero or more headers
  - An empty line - Indicating end of header fields
  - Optionally the message body



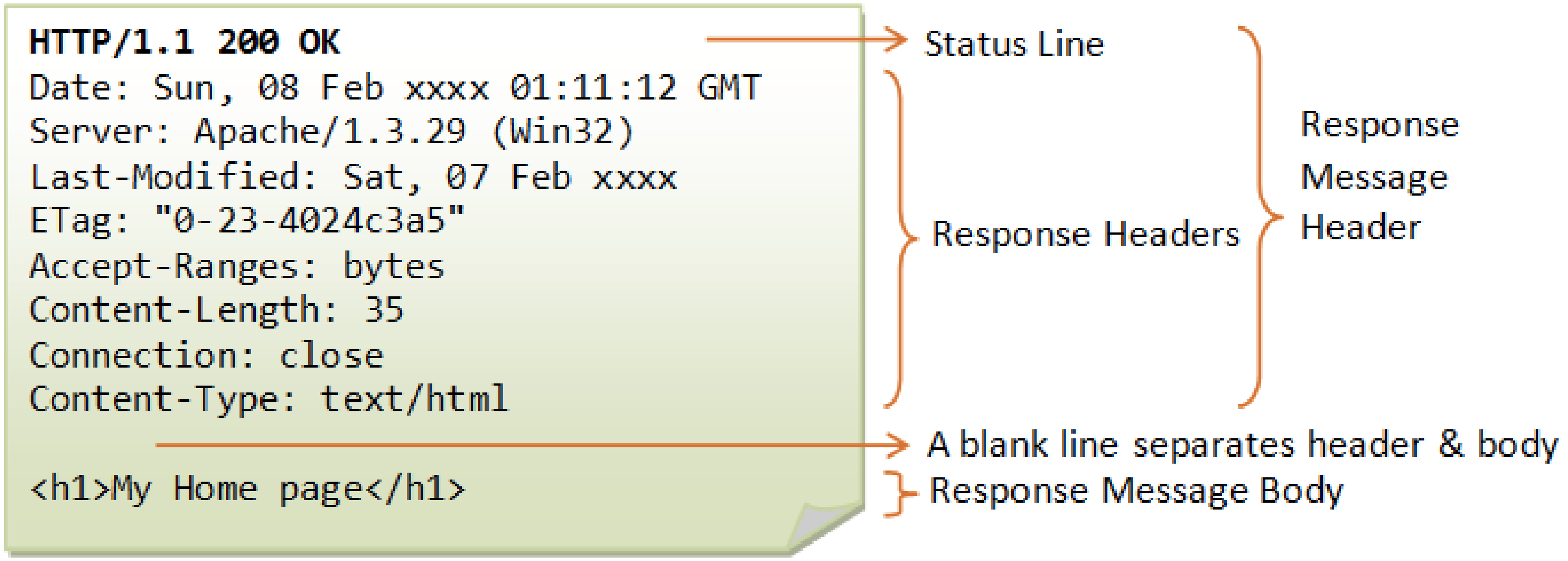
```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

HTTP Response

axess academy

- After receiving and interpreting a request message, a server responds with an HTTP response message:
  - A Status-line
  - Zero or more header (General|Response|Entity) fields followed by CRLF
  - An empty line (i.e., a line with nothing preceding the CRLF) indicating the end of the header fields
  - Optionally a message-body



```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

HTTP Methods

- HTTP GET
- HTTP POST
- HTTP PUT
- HTTP DELETE
- HTTP PATCH
- HTTP OPTIONS
- HTTP HEAD

HTTP Status Codes

axess academy

- 1xx - Informational
- 2xx - Success
- 3xx - Redirection
- 4xx - Client Error
- 5xx - Server Error

HTTP Important Codes

axess academy

- 200 - OK - (GET Request)
- 201 - Created - (POST Request)
- 301 - Moved Permanently
- 302 - Moved Temporarily
- 304 - Not Modified
- 400 - Bad Request
- 401 - Unauthorised
- 403 - Forbidden
- 404 - Not Found
- 405 - Method Not Allowed
- 408 - Request Timeout
- 500 - Internal Server Error
- 503 - Service Unavailable
- 504 - Gateway Timeout

URI, URN, URL

axess academy

URI : Uniform Resource Identifier, is a generic reference to a resource on a network. For example, https://axess.sc.com/apis/account-services/account-information is a URI.

URN : Uniform Resource Name, which is just a unique ID for an object. URI is a generic where's URN is a subset.

URL : In the history of the Internet, the term URL (Uniform Resource Locator) was frequently used to refer to a type of a URI that includes a network protocol.

REST

REST : Representational State Transfer - RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data. It is an architectural style and approach to communications often used in web services development.

- **Representational** : REST resources can be represented in virtually any form including JSON, XML or even HTML - whatever form best suits the consumer of those resources.
-

  State : when working with REST, you're more concerned with State of a resource than with the actions you can take against resources.

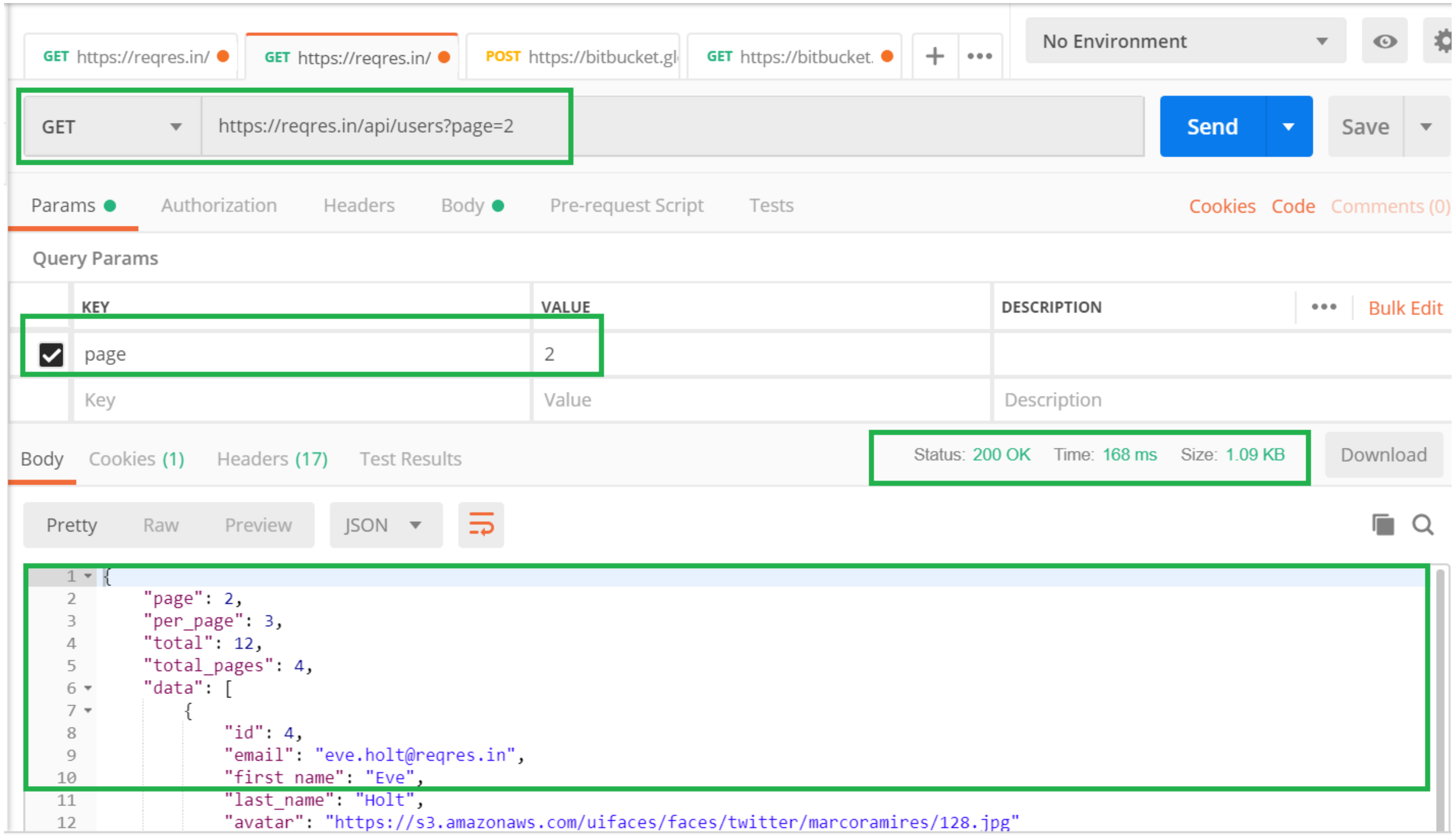
- **Transfer:** REST involves transferring resource data, in some representational form, one application to another.

REST is about transferring the state of resources- in a representational form that is most appropriate for the client or server- from a server to a client (or vice versa)

Lab 1: GET a list of users

axess academy

- GET a single user : https://reqres.in/api/users?page=2



GET https://reqres.in/ | GET https://reqres.in/ | POST https://bitbucket.gl | GET https://bitbucket.

No Environment

GET https://reqres.in/api/users?page=2

Send | Save

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

Query Params

	KEY	VALUE	DESCRIPTION
✓	page	2	
	Key	Value	Description

Body | Cookies (1) | Headers (17) | Test Results

Status: 200 OK Time: 168 ms Size: 1.09 KB

Download

Pretty | Raw | Preview | JSON

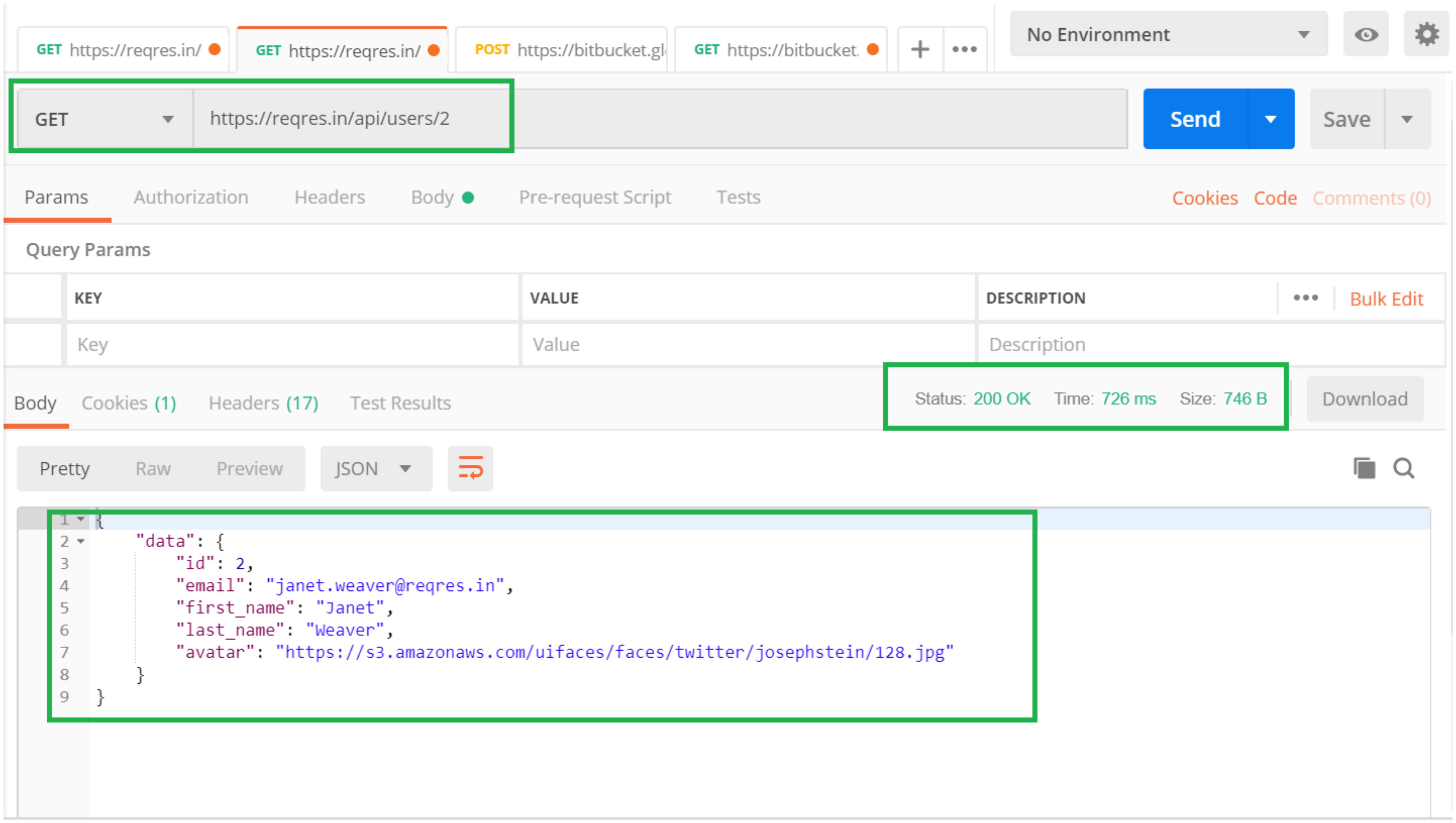
```
{
    "page": 2,
    "per_page": 3,
    "total": 12,
    "total_pages": 4,
    "data": [
        {
            "id": 4,
            "email": "eve.holt@reqres.in",
            "first_name": "Eve",
            "last_name": "Holt",
            "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/marcoramires/128.jpg"
```

INTERNAL

Lab 2: GET a single user

axess academy

- GET a single user : https://reqres.in/api/users/2



GET https://reqres.in/api/users/2

Status: 200 OK Time: 726 ms Size: 746 B

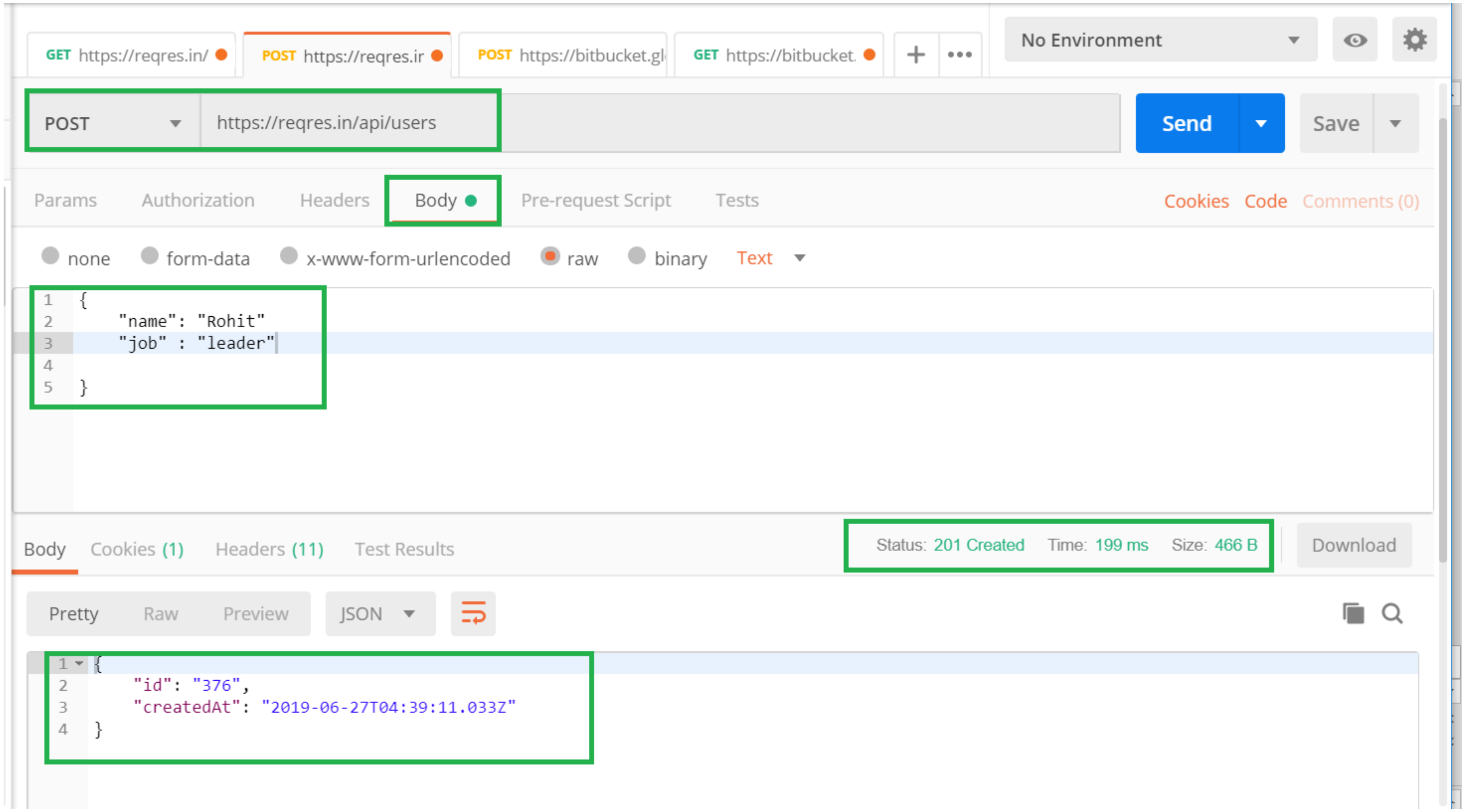
```
{
    "data": {
        "id": 2,
        "email": "janet.weaver@reqres.in",
        "first_name": "Janet",
        "last_name": "Weaver",
        "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/josephstein/128.jpg"
    }
}
```

INTERNAL

Lab 3: Create a new user

axess academy

- POST a single user : https://reqres.in/api/users



POST https://reqres.in/api/users

Body: raw, Text

```
{
    "name": "Rohit"
    "job" : "leader"

}
```

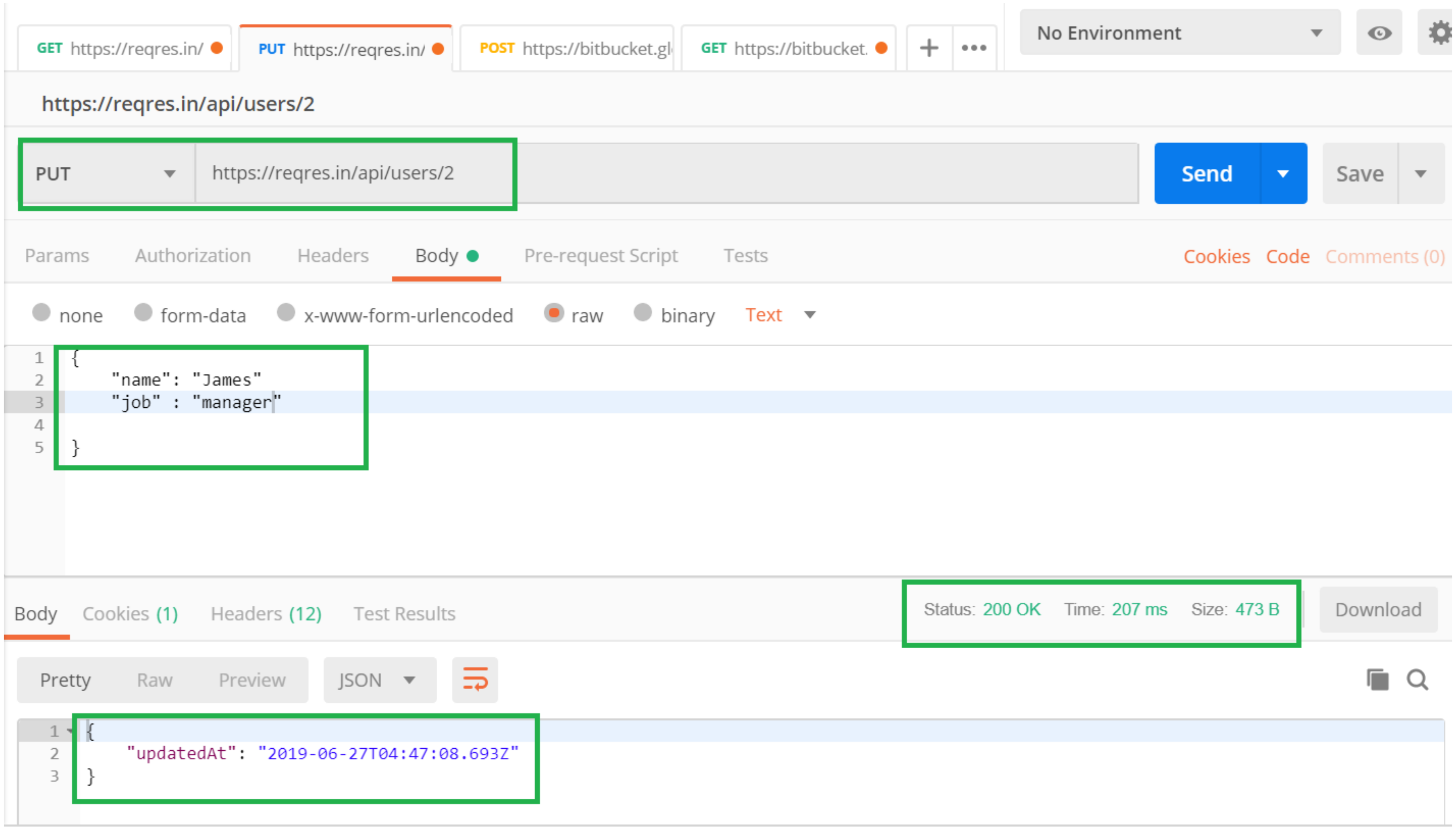
Status: 201 Created Time: 199 ms Size: 466 B

```
{
    "id": "376",
    "createdAt": "2019-06-27T04:39:11.033Z"
}
```

Lab 4: Update an existing user

axess academy

- PUT : https://reqres.in/api/users/2



```
{
    "name": "James"
    "job" : "manager"

}
```

Status: 200 OK Time: 207 ms Size: 473 B

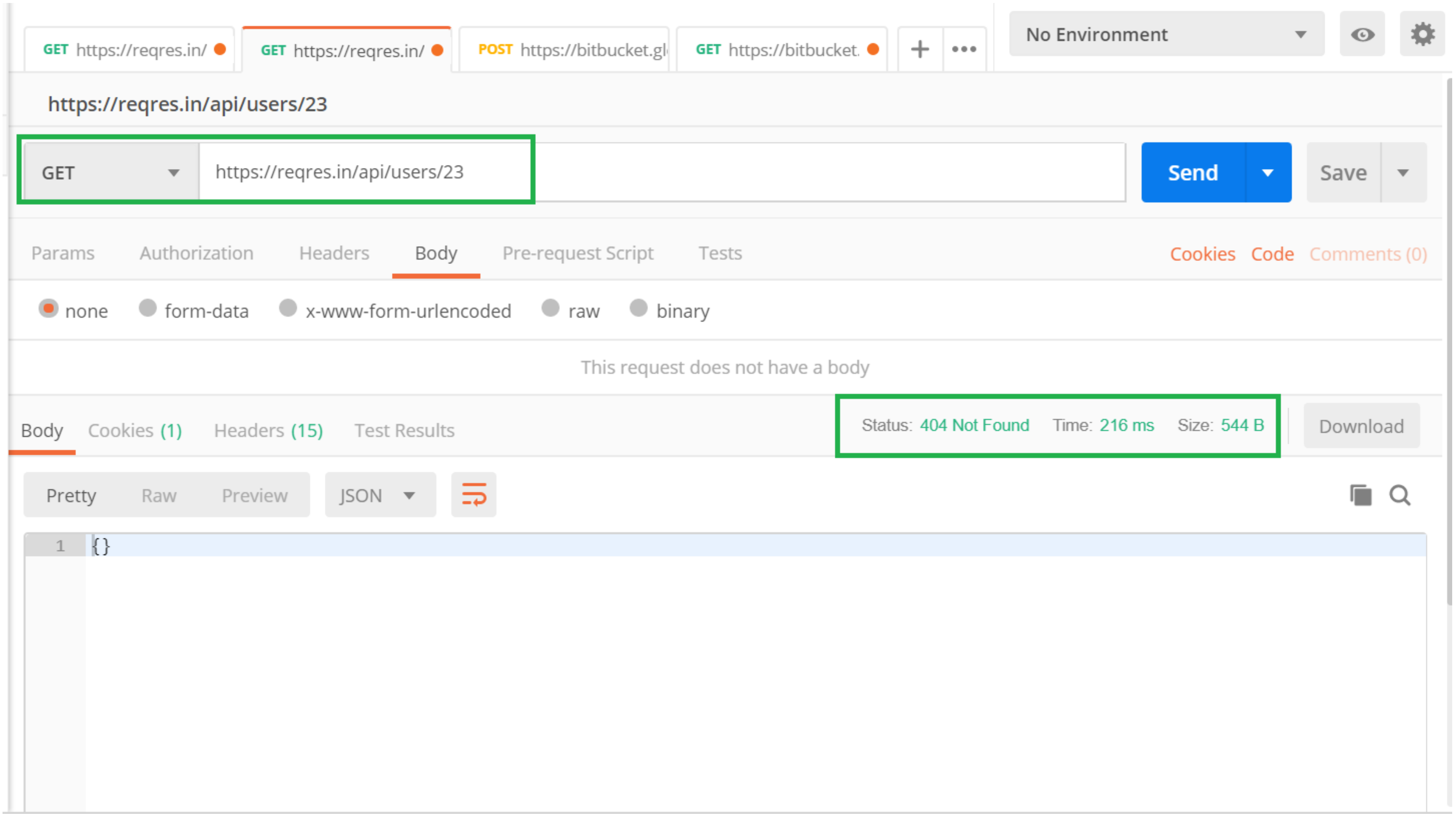
```
{
    "updatedAt": "2019-06-27T04:47:08.693Z"
}
```

INTERNAL

Lab 5: User Not found

axess academy

- GET : https://reqres.in/api/users/23



GET https://reqres.in/api/users/23

This request does not have a body

Status: 404 Not Found Time: 216 ms Size: 544 B

```
{}
```

Guiding Principles of REST

- Client-Server
- Stateless
- Cacheable
- Uniform Resource
- Layered System
- Concept of Resource and HTTP method Verbs (Resource Methods)
- Idempotent REST APIs
- REST != HTTP

Statelessness

- No State stored on the server
- Every HTTP request executes in isolation on the server
- Simple to design and evolve
- Easier to scale
- Avoid HTTP Sessions and Cookies
- No Side effects

Idempotent REST APIs

- In the context of REST APIs, when making multiple identical requests has the same effect as making a single request – then that REST API is called **idempotent**.

- If you follow REST principles in designing API, you will have automatically **idempotent REST APIs** for GET, PUT, DELETE, HEAD, OPTIONS and TRACE HTTP methods. Only POST APIs will not be idempotent.

- POST APIs are used to create a new resource on server. So when you invoke the same POST request N times, you will have N new resources on the server. So, **POST is not idempotent**.

Content Negotiation

axess academy

- Generally, resources can have multiple presentations, mostly because there may be multiple different clients expecting different representations. Asking for a suitable presentation by a client, is referred as content negotiation.

- Server-driven Vs Agent-driven Content Negotiation

    - 1. Content negotiation using HTTP headers - Content-Type: application/json, Accept: application/json

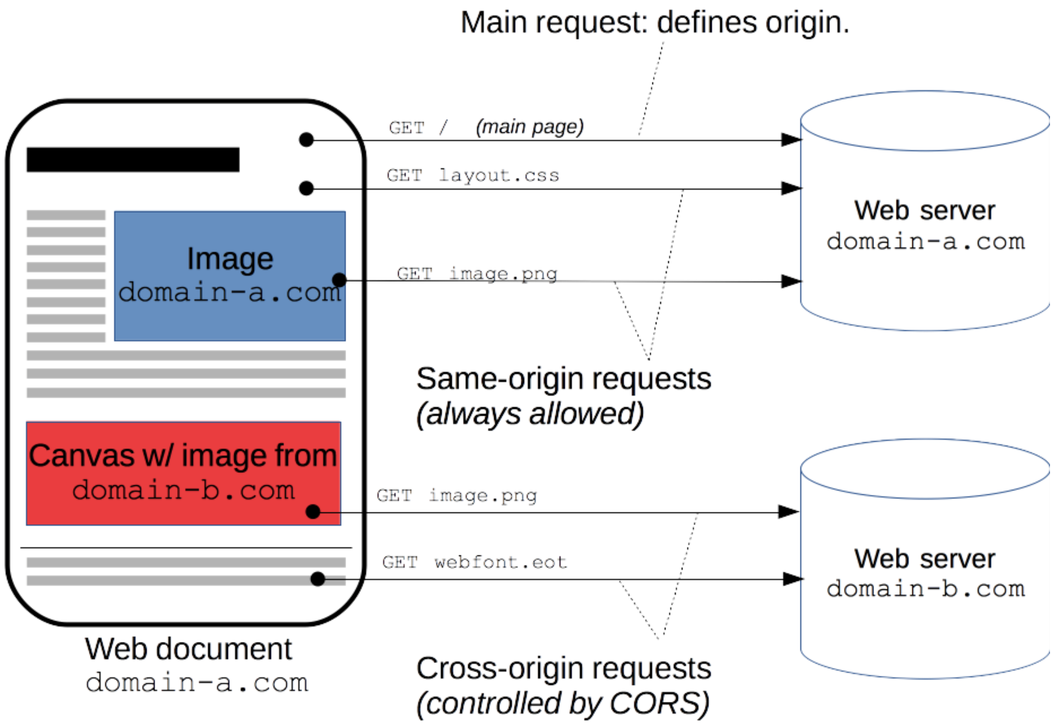
    - 2. Content negotiation using URL patterns - http://rest.api.com/v1/employees/20423.xml, http://rest.api.com/v1/employees/20423.json

CORS - Cross Origin Resource Sharing

axess academy

- Cross-Origin Resource Sharing (CORS) is a mechanism that uses additional HTTP headers to tell a browser to let a web application running at one origin (domain) have permission to access selected resources from a server at a different origin.

- An example of a cross-origin request: The frontend JavaScript code for a web application served from http://domain-a.com uses XMLHttpRequest to make a request for http://api.domain-b.com/data.json.



Tools and Techniques

axess academy

1 cURL (pronounced 'curl') is a computer software project providing a library (libcurl) and command-line tool (curl) for transferring data using various protocols. It was first released in 1997. The name stands for "Client URL".This tool is used in development and testing of API.

2 Postman is a tool used to send requests and receive responses through REST API.You can use this dedicated app interface to organize and save your tests independently

3 Swagger : Design is the foundation of your API development. Swagger makes API design a breeze, with easy-to-use tools for developers, architects, and product owners.

axess academy

Thank You