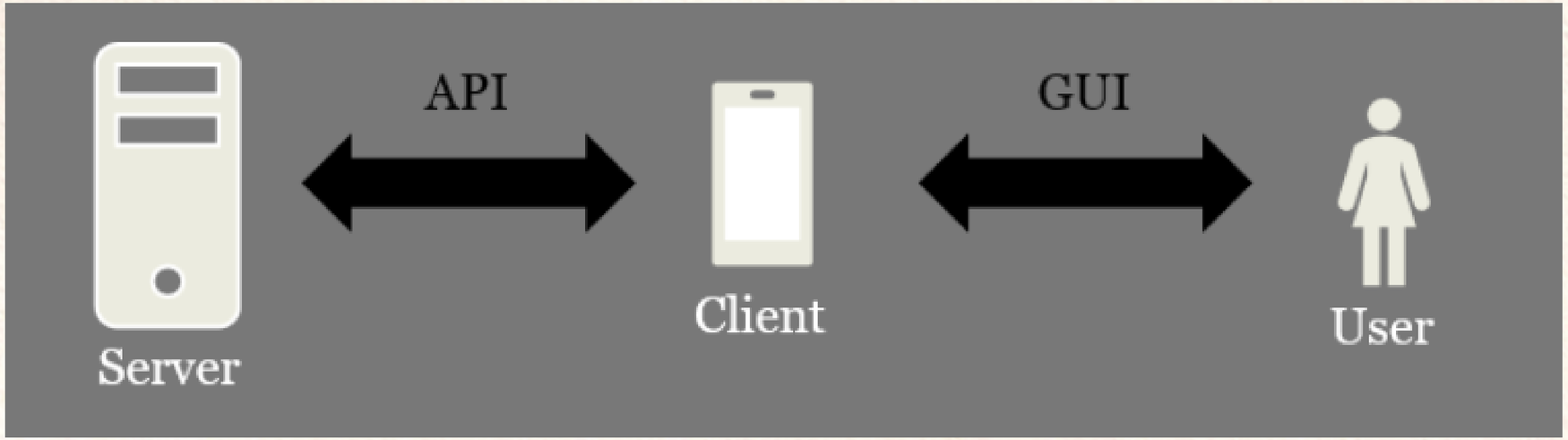
REST API

Vu Nhat Duy-ITITIU17047

TABLE OF CONTENTS

1. APPLICATION PROGRAMMING INTERFACE

A GUI is an interface for human <> machine communication



An API is an interface for machine <> machine communication .

An API making use of HTTP is called Web API

1. DIFFERENT TYPES OF APIS

1. Web APIs
2. Library APIs
3. Operating System APIs
4. Database APIs

2. WHAT IS REST API?

REST (Representational State Transfer):

- Architecture style developed by Roy Fielding in 2000.
- Contains a set of principles used to design and develop web

REST PRINCIPLES

1. Client-Server Architecture
2. Stateless Communication
3. Cacheability
4. Uniform Interface
5. Layered System
6. Code-on-Demand

3. REST API

- Use URLs to identify resources
- Use HTTP methods to specify operations: POST, GET, PUT, DELETE
- Use HTTP headers
- Use HTTP status code to indicate success, failure

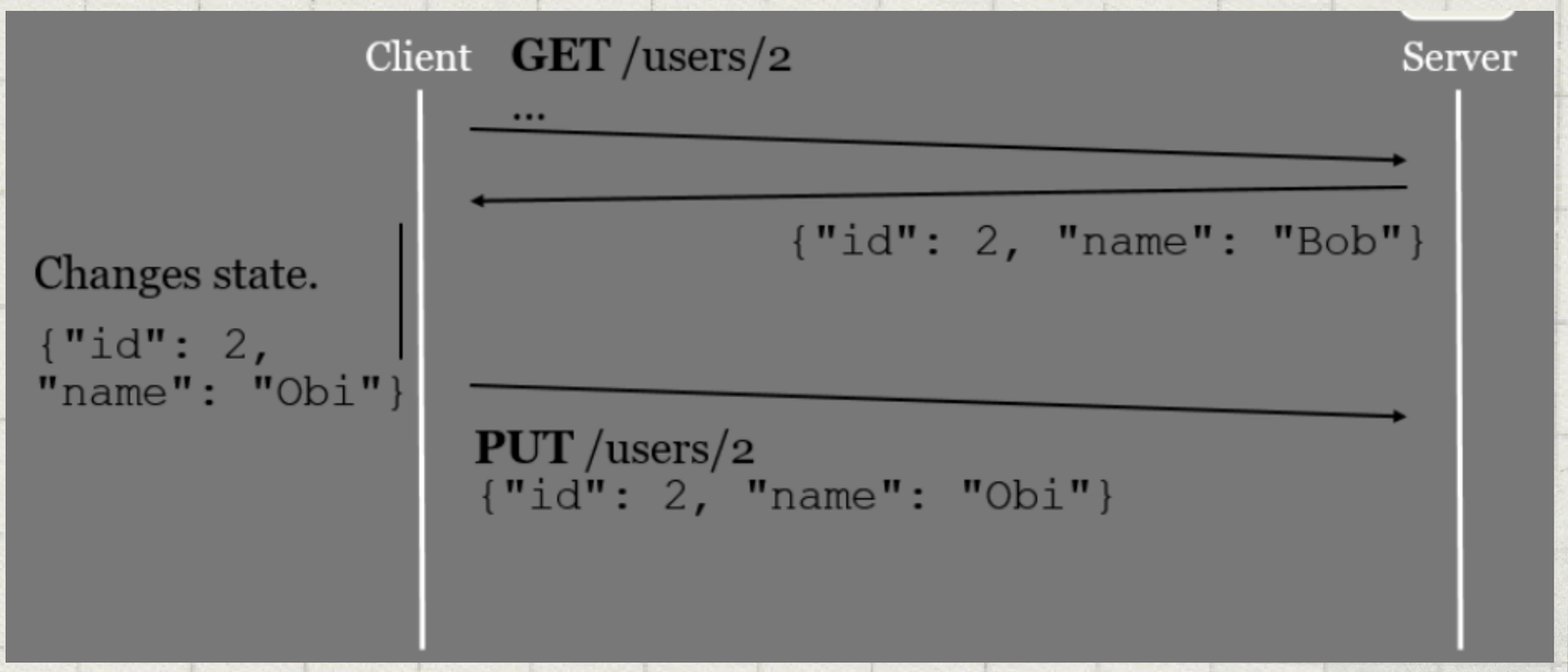
3. THE METHODS

CREATE: POST

RETIREVE: GET

UPDATE: PUT

DELETE: DELETE



REST EXAMPLE

Context: A server with information about users

- The GET method is used to retrieve resources
- GET /users
- GET /users/2
- GET /users/2/gender
- GET /users/2/name
- GET /users?page=1
- GET /users?gender=female

REST EXAMPLE

Context: A server with information about users

- The GET method is used to retrieve resources
- Which data format? Specified by the Accept header!

```
GET /users HTTP/1.1
Host: the-website.com
Accept: application/json
```


```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 66

[
  {"id": 1, "name": "Alice"},
  {"id": 2, "name": "Bob"}
]
```

REST EXAMPLE

Context: A server with information about users

- The POST method is used to create resources
- Which data format? Specified by the Accept and Content-Type header!

```
POST /users HTTP/1.1
Host: the-website.com
Accept: application/json
Content-Type: application/xml
Content-Length: 49

<user>
  <name>Claire</name>
```

```
HTTP/1.1 201 Created
Location: /users/3
Content-Type: application/json
Content-Length: 28

{"id": 3, "name": "Claire"}
```

REST EXAMPLE

Context: A server with information about users

- The PUT method is used to update an entire resource

```
PUT /users/3 HTTP/1.1
Host: the-website.com
Content-Type: application/xml
Content-Length: 52

<user>
  <id>3</id>
  <name>Cecilia</name>
</user>
```

```
HTTP/1.1 204 No Content
```

PUT can also be used to create a resource if you know which URI it should have in advance.

REST EXAMPLE

Context: A server with information about users

- The DELETE method is used to delete a resource

```
DELETE /users/2 HTTP/1.1
Host: the-website.com
```

```
HTTP/1.1 204 No Content
```

REST EXAMPLE

Context: A server with information about users

- What if something goes wrong?
- Use the HTTP status codes to indicate success/failure

```
GET /users/999 HTTP/1.1
Host: the-website.com
Accept: application/json
```

```
HTTP/1.1 404 Not Found
```

- More information about status codes here:
- https://www.restapitutorial.com/httpstatuscodes.html

DEMO

Thank you!

Any questions?