

REST API for Beginners

Davey Herrera





Agenda



REST API

- A brief history
- Basic (and “new”) Concepts
- HTTP response status codes

Examples of the flow

- The Library
- Adding the vocabulary
- HTTP verbs
 - Most common verbs
- Important Notes
- Short demo with POSTMAN



A brief history

- “Invented” in 2000 by Roy Fielding as his PhD dissertation
- Every time we visit a web site, we make a REST call
- Every time we “submit” a web form we make a REST call.

Some definitions first

HTTP & HTTPS

- HyperText
- Transfer
- Protocol
- (Secure)

(A hypertext is text displayed on a computer display or electronic devices)

URL

- Uniform
- Resource
- Locator

<https://anthology.com>

Who is this “Client”



WHAT IS REST?: REpresentational
State
Transfer



WHAT IS REST?: REpresentational State Transfer



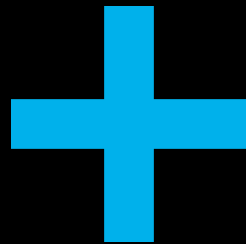
≈

```
{  
  id: 1,  
  objectName: "umbrella",  
  color: "white"  
  isOpen: true,  
  isWet: false  
}
```

REpresentational
State
Transfer

Application
Programming
Interface

REpresentational
State
Transfer
ful

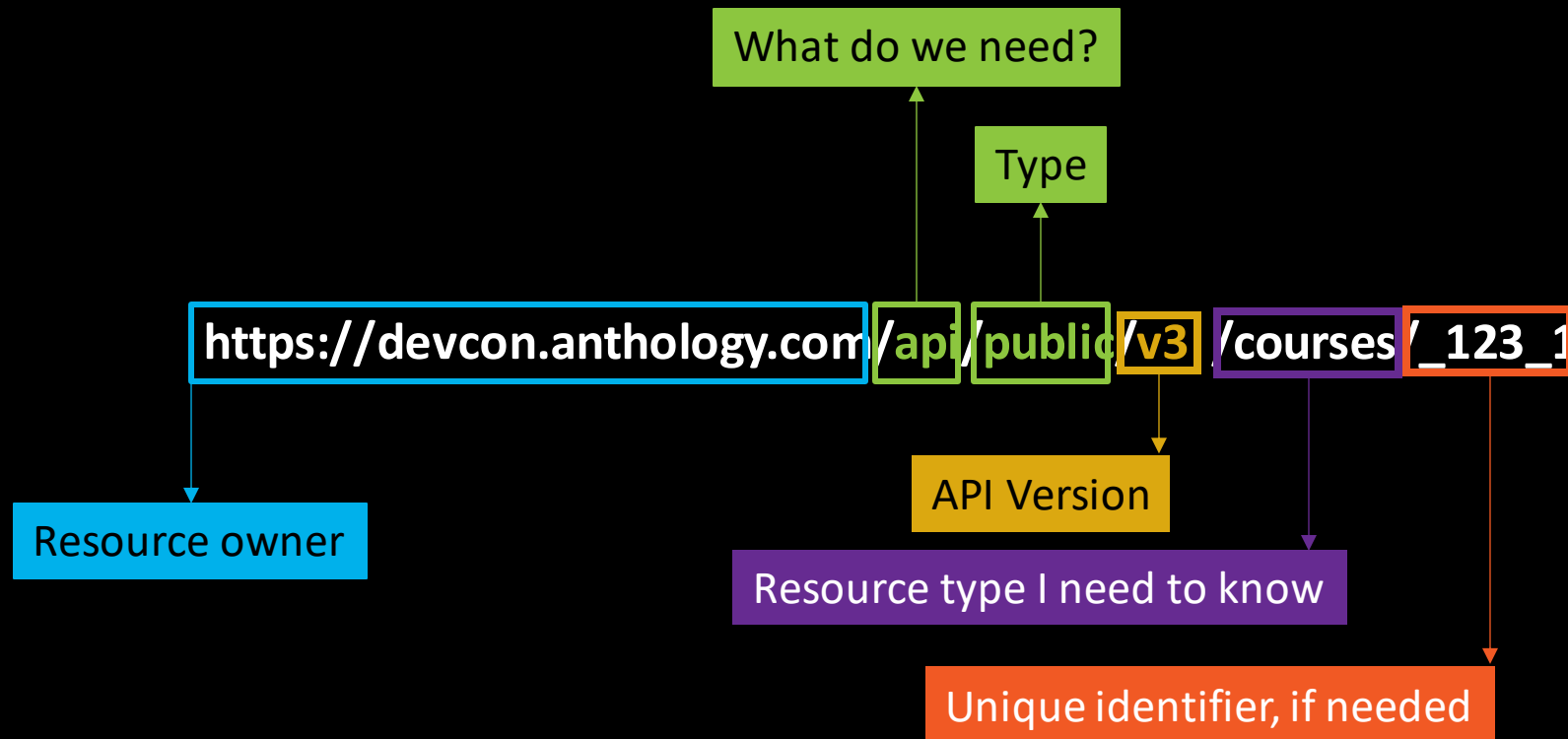


Application
Programming
Interface

Endpoints: This is how we call information from a system

https://devcon.anthology.com/api/public/v3/courses/_123_1

Anatomy of an endpoint

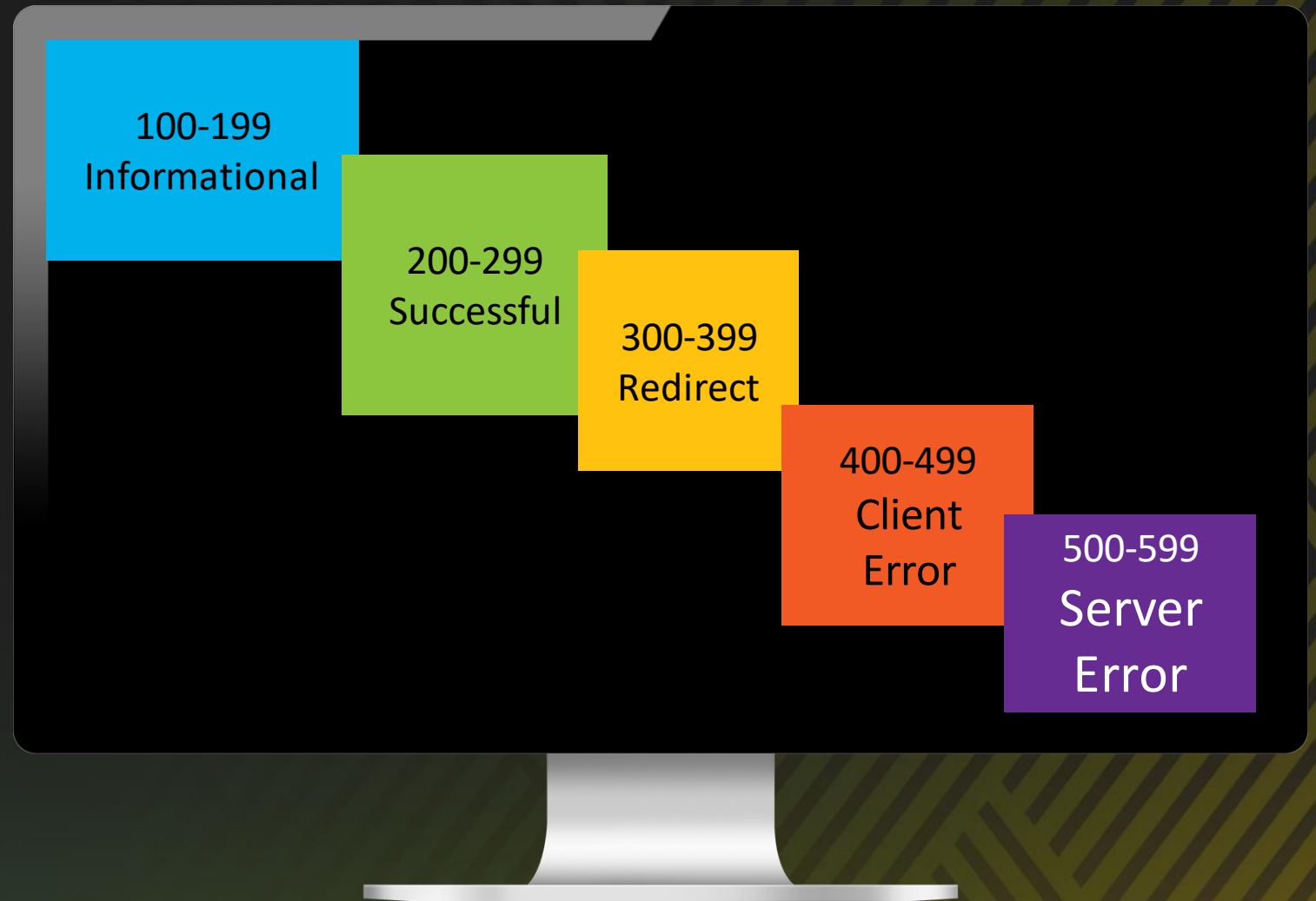


HTTP Response status codes



HTTP Response status codes

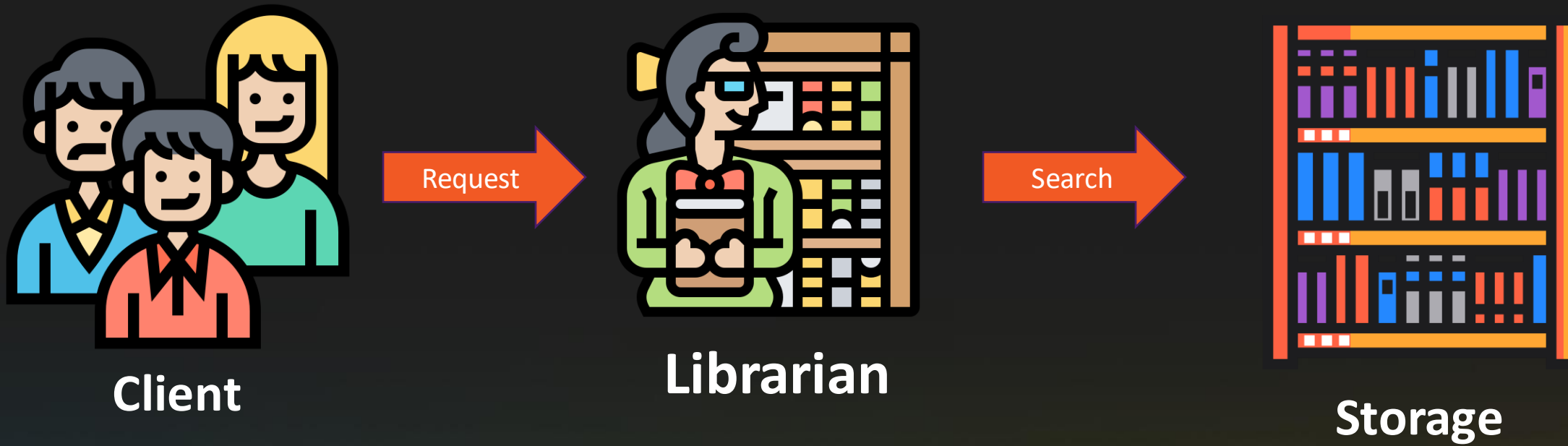
- HTTP response status codes indicate whether a specific HTTP request has been successfully completed, they are grouped in five classes



Examples of the flow

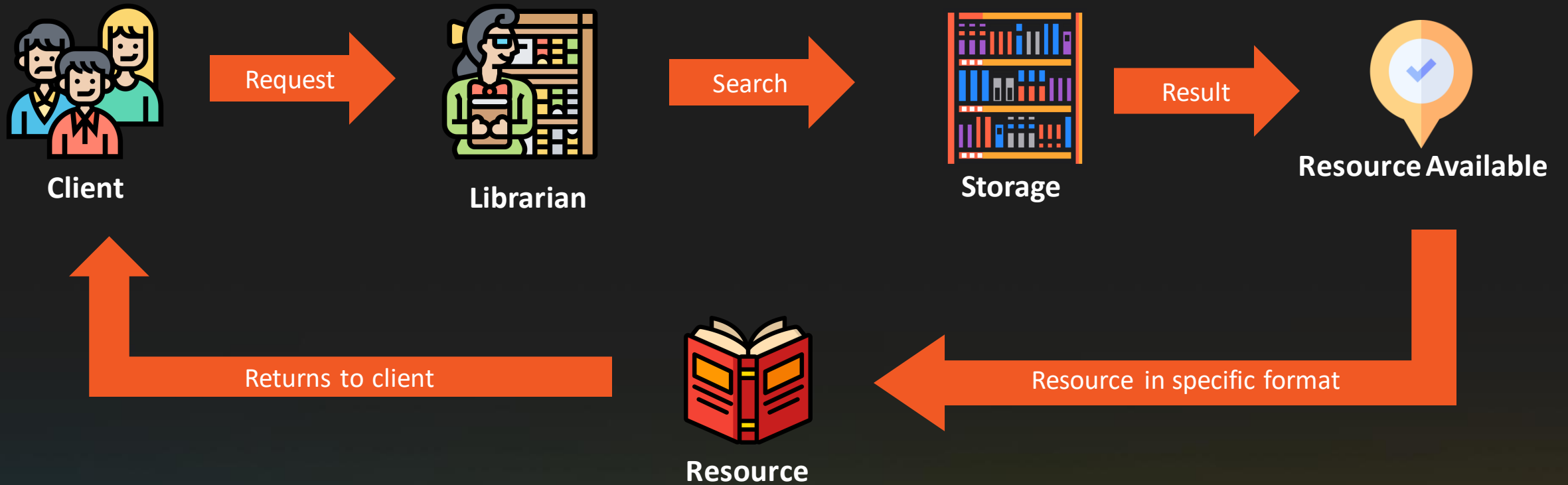


The Library

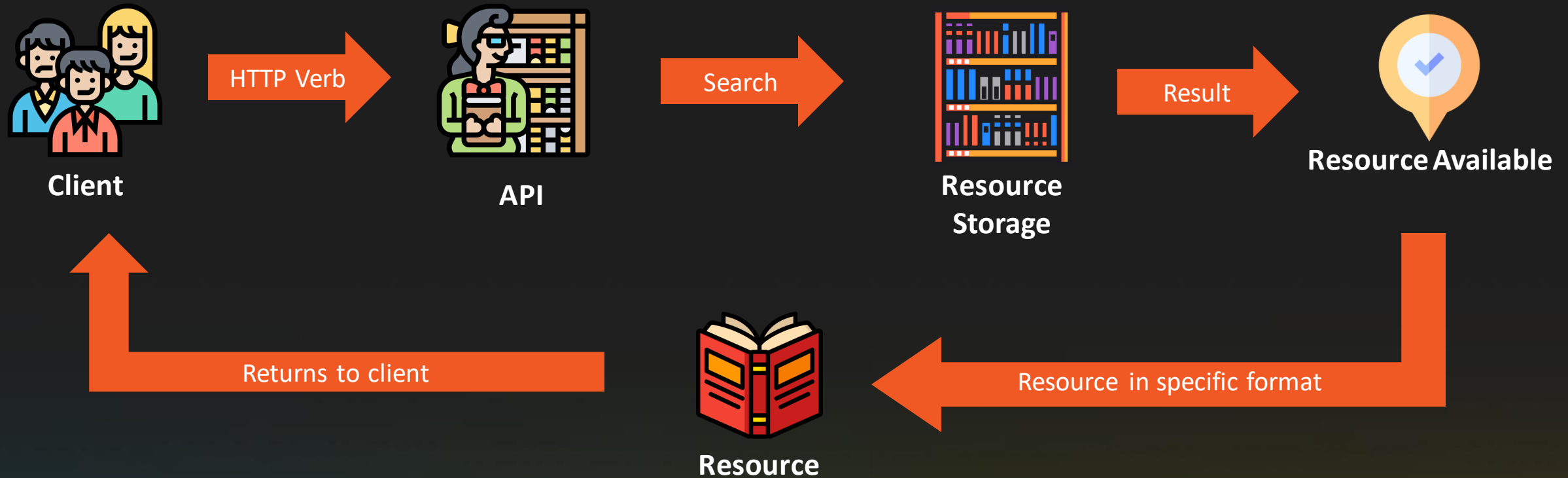


A client requests a librarian a specific resource, the librarian searches for the resource and gives it back to the client

The Library



The Library (But applied)



The Resource



- Course (id)
- Courses
- User (id)
- Users
- Calendar
- Content Item
- Announcement
- Term
- ...

HTTP Verbs



HTTP Verbs

- GET
- HEAD
- POST
- PUT
- DELETE
- CONNECT
- OPTIONS
- TRACE
- PATCH



HTTP Verbs – The usual ones

- **GET** - Give me a list of users
- **POST** – Create a course for Python 101
- **PATCH/PUT**– Update this course name
- **DELETE** – Remove this content



Important Notes

- **Stateless:** “Each request from client to the server must contain all the information necessary to understand and complete the request, the server cannot take advantage of any previously stored context information of the server”
- **Client is the one who does all the work,** the server only listens for requests from the client
- **API documentation is a contract:** If you promise to implement the API “like this”, I promise you will receive data this way.

Resources

- Icons taken from: "Icon made by Pixel perfect from www.flaticon.com"
- Icons taken from: "Icon made by Freepik from www.flaticon.com"

Some definitions

- <https://restfulapi.net/>
- Sources: <https://devmountain.com/blog/what-is-the-difference-between-rest-and-restful-apis/#:~:text=REST%20stands%20for%20representational%20state,adheres%20to%20the%20REST%20architecture.>
- <https://www.ibm.com/cloud/learn/rest-apis>
- <https://aws.amazon.com/what-is/api/#:~:text=API%20stands%20for%20Application%20Programming,other%20using%20requests%20and%20responses.>
- https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional
- <https://www.redhat.com/en/topics/api>
- <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>



DEMO
TIME



anthology
together