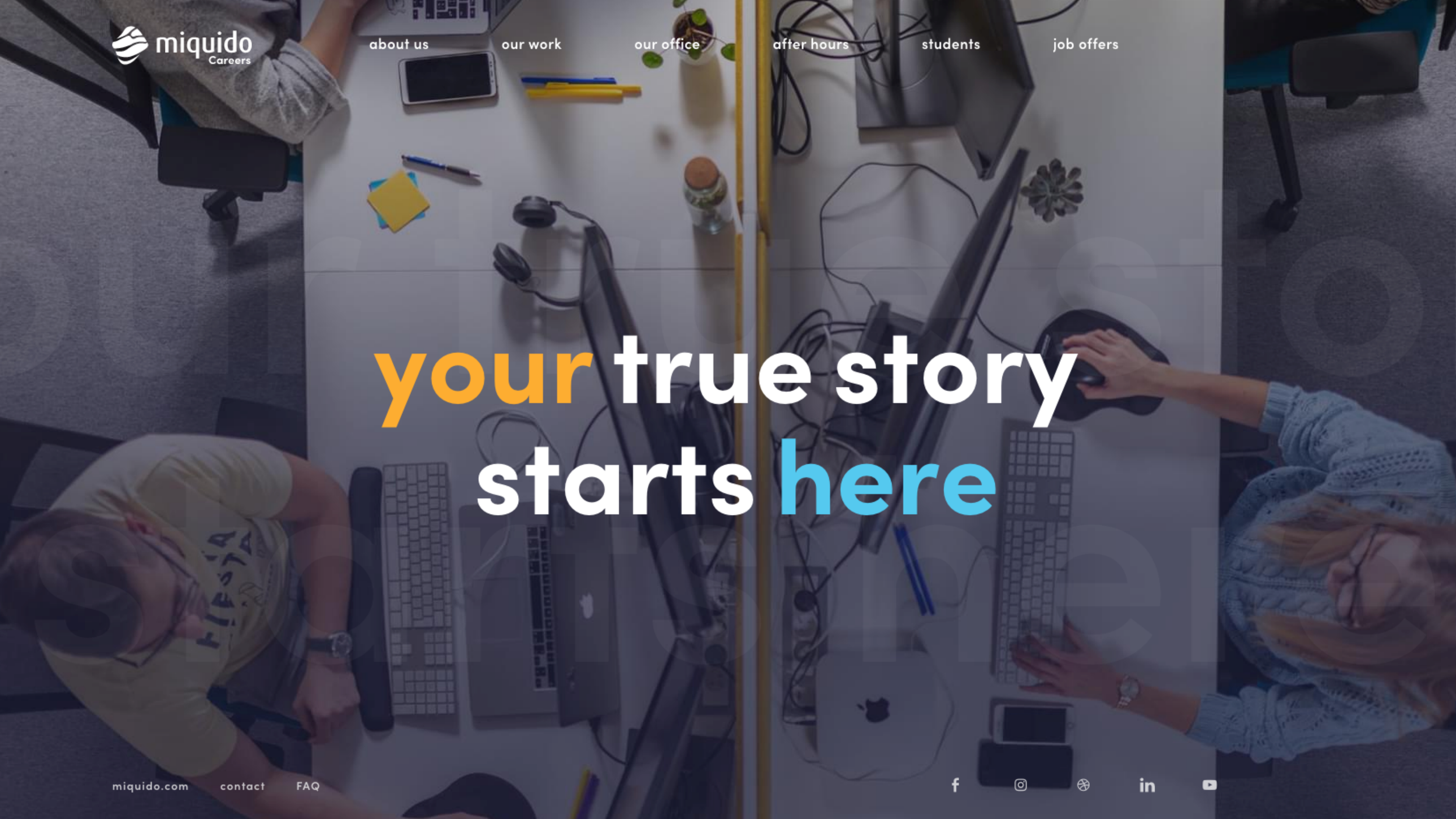
A little bit of REST API testing

#BUTFIRSTOFALL

#WHOAMI


miquido
Careers
about us
our work
our office
after hours
students
job offers
your true story
starts here
miquido.com
contact
FAQ

#whoami

Dmytro Kostiukevych

- perfectionist
- minimalist
- senior

facebook.com/dmytro.kostiukevych

linkedin.com/in/dmytro-kostiukevych-697943103/

I distance myself from the bullshit.

#DIMA

#DISCLAIMER

#SMALLSURVEY

#AGENDA

- INTRODUCTION.
- DEEP DIVE.
- LESSONS LEARNED.

#AGENDA

- WHAT IS AN API?
- WHAT YOU SHOULD KNOW TO START API TESTING?
- TESTING OF THE REST-API DOCUMENTATION.
- CORRECT APPROACH TO REST-API TESTING

(ONE OF...)

CHAPTER #1 -
WHAT IS AN API?

APPLICATION **P**ROGRAMMING

INTERFACE

THE ONLY THING THAT CAN SAVE YOU, THIS IS…

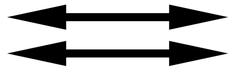
UNDERSTANDING
OF THE SUBJECT AREA

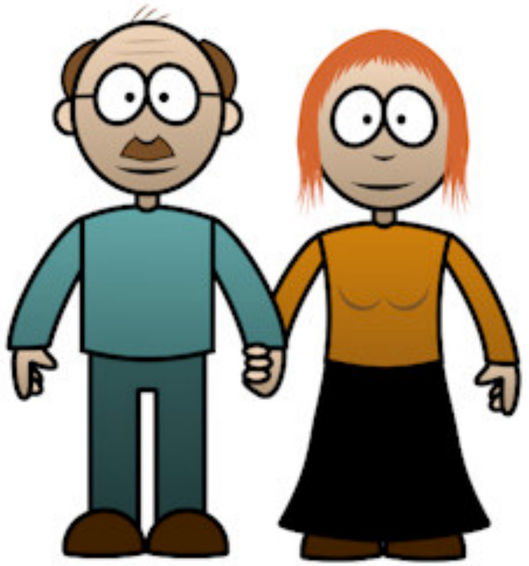
LET'S START BY TRYING TO DESCRIBE IT

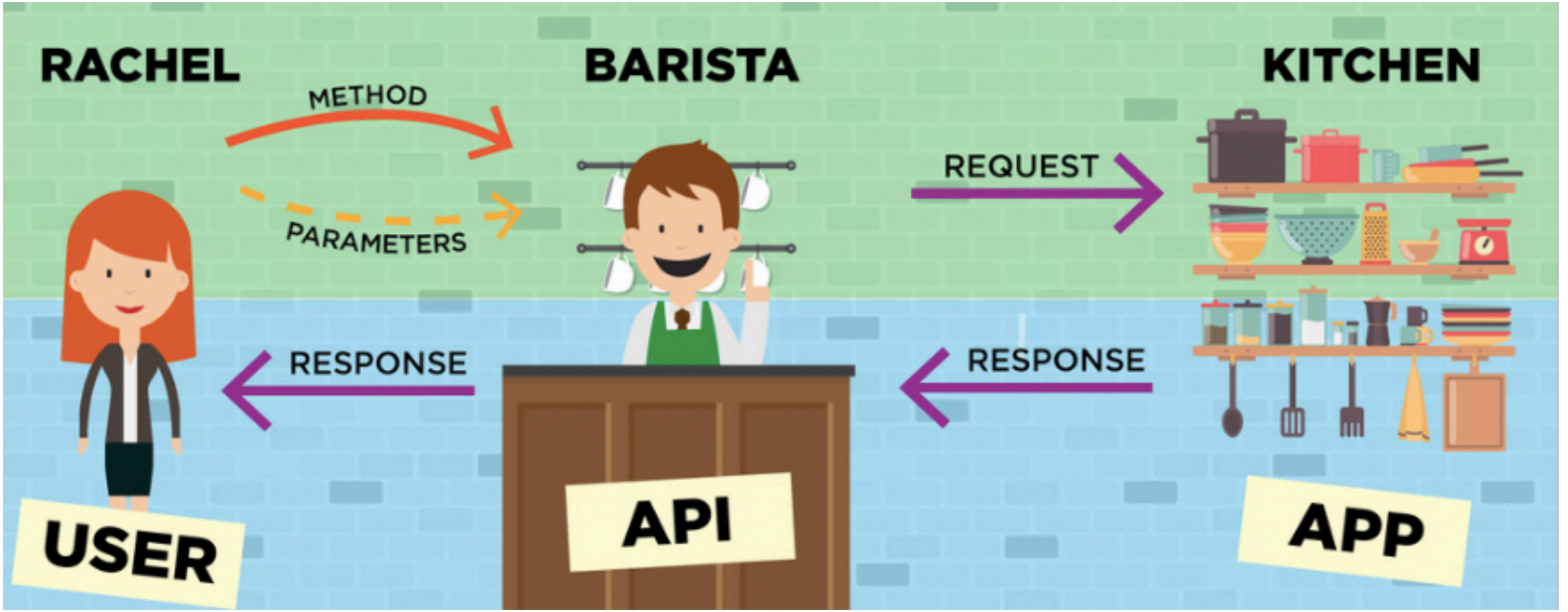
CLIENT

API

SERVER






RACHEL
BARISTA
KITCHEN
METHOD
PARAMETERS
REQUEST
RESPONSE
RESPONSE
USER
API
APP

CHAPTER #2 -

WHAT YOU SHOULD KNOW TO START REST-API TESTING?

CHAPTER #2.1

- HTTP methods / CRUD
- HTTP headers
- HTTP status codes

HTTP Methods

CRUD: create, read, update, delete

- POST
- GET
- PUT, PATCH
- DELETE

HTTP HEADERS

Headers can be grouped according to their contexts:

- **General header:** Headers applying to both requests and responses but with no relation to the data eventually transmitted in the body.

- **Request header:** Headers containing more information about the resource to be fetched or about the client itself.

- **Response header:** Headers with additional information about the response, like its location or about the server itself (name and version etc.).

- **Entity header:** Headers containing more information about the body of the entity, like its content length or its MIME-type.

QUICK DEMO

API TESTING STARTS WITH KNOWING THE STATUSES OF THE CODES…


#rest-api testing
#status_codes

QUIZ

STATUS CODES

200 - ???
201 - ???
204 - ???

304 - ???

400 - ???
401 - ???
403 - ???
404 - ???
405 - ???

500 - ???
502 - ???

STATUS CODES

??? - Accepted
??? - Partial Content

??? - Conflict
??? - I’m a teapot
??? - Unprocessable Entity (WebDAV)

??? - Service Unavailable
??? - Gateway Timeout

1xx Informational

100 Continue
101 Switching Protocols
102 Processing (WebDAV)

2xx Success

★ 200 OK
★ 201 Created
202 Accepted
203 Non-Authoritative Information
★ 204 No Content
205 Reset Content
206 Partial Content
207 Multi-Status (WebDAV)
208 Already Reported (WebDAV)
226 IM Used

3xx Redirection

300 Multiple Choices
301 Moved Permanently
302 Found
303 See Other
★ 304 Not Modified
305 Use Proxy
306 (Unused)
307 Temporary Redirect
308 Permanent Redirect (experimental)

4xx Client Error

★ 400 Bad Request
★ 401 Unauthorized
402 Payment Required
★ 403 Forbidden
★ 404 Not Found
405 Method Not Allowed
406 Not Acceptable
407 Proxy Authentication Required
408 Request Timeout
★ 409 Conflict
410 Gone
411 Length Required
412 Precondition Failed
413 Request Entity Too Large
414 Request-URI Too Long
415 Unsupported Media Type
416 Requested Range Not Satisfiable
417 Expectation Failed
418 I'm a teapot (RFC 2324)
420 Enhance Your Calm (Twitter)
422 Unprocessable Entity (WebDAV)
423 Locked (WebDAV)
424 Failed Dependency (WebDAV)
425 Reserved for WebDAV
426 Upgrade Required
428 Precondition Required
429 Too Many Requests
431 Request Header Fields Too Large
444 No Response (Nginx)
449 Retry With (Microsoft)
450 Blocked by Windows Parental Controls (Microsoft)
451 Unavailable For Legal Reasons
499 Client Closed Request (Nginx)

5xx Server Error

★ 500 Internal Server Error
501 Not Implemented
502 Bad Gateway
503 Service Unavailable
504 Gateway Timeout
505 HTTP Version Not Supported
506 Variant Also Negotiates (Experimental)
507 Insufficient Storage (WebDAV)
508 Loop Detected (WebDAV)
509 Bandwidth Limit Exceeded (Apache)
510 Not Extended
511 Network Authentication Required
598 Network read timeout error
599 Network connect timeout error

CRUD - Create, read, update and delete

HTTP verb	CRUD correspondence	Collection: / orders	Instance: / orders / {id}
GET	READ	Read a list orders. 200 OK.	Read the detail of a single order. 200 OK.
POST	CREATE	Create a new order. 201 Created.	-
PUT	UPDATE / CREATE	-	Full Update. 200 OK. Create a specific order. 201 Created.
PATCH	UPDATE	-	Partial Update. 200 OK.
DELETE	DELETE	-	Delete order. 200 OK or 204.

CHAPTER #2.2
WHAT IS REST?

Representational State Transfer

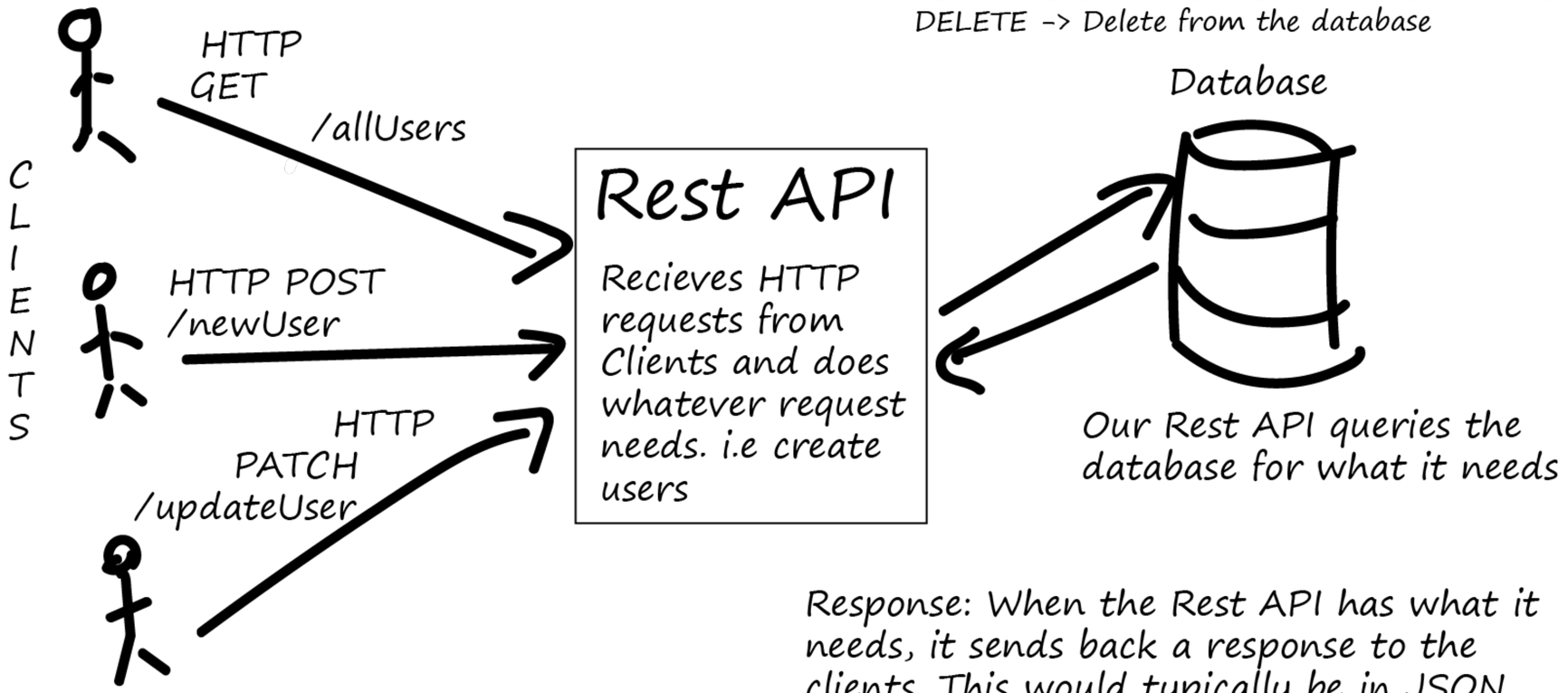
It means when a RESTful API is called, the server will *transfer* to the client
a *representation* of the *state* of the requested resource.

WHAT IS REST?

- Architecture style for designing networked applications.
- Uniform interface.
- Client - server.
- Stateless.
- Cacheable.

Rest API Basics

Typical HTTP Verbs:
GET -> Read from Database
PUT -> Update/Replace row in Database
PATCH -> Update/Modify row in Database
POST -> Create a new record in the database
DELETE -> Delete from the database



Our Rest API queries the database for what it needs

Our Clients, send HTTP Requests and wait for responses

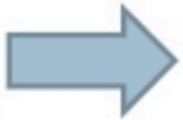
Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.

OK…. BUT, WHAT'S UNDER….?

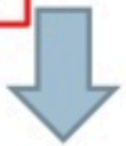
CHAPTER #2.2
SERIALIZATION / DESERIALIZATION

SQL results as JSON

ID	Name	Price	Tags	Data
15	Bike	100	[...]	{...}
16	Car	29000	[...]	{...}
17	BB Ba...	29,99		{...}
18	Blade	18.50	[...]	{...}
19	Helmet	41.99	[...]	{...}



```
SELECT ProductID, Name, Price,
       Tags = JSON_QUERY(Tags),
       Data = JSON_QUERY(Data)
FROM Product
FOR JSON PATH
```

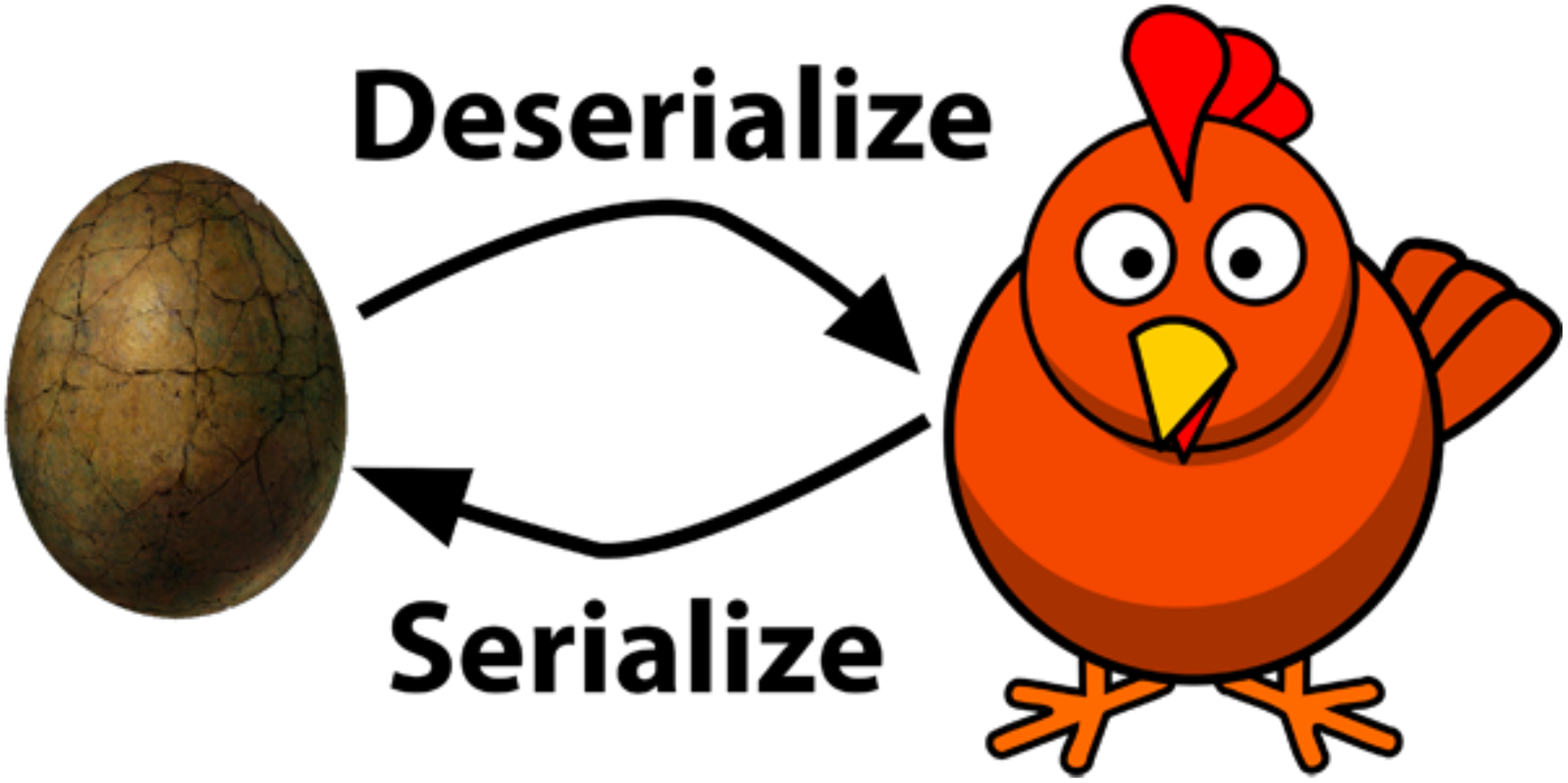



```
[
 {"ProductID":15,"Name":"Bike","Price":100,"Data":{"Type":"Part","MadeIn":"SRB"}},
 {"ProductID":16,"Name":"Car","Price":29000,"Tags":["promo"],"Data":{"Cost":11.67,"Type":"Part"}},
 {"ProductID":17,"Name":"BB Ball Bearing","Price":28.99,"Data":{"Cost":21.162700,"Type":"Part"}},
 {"ProductID":18,"Name":"Blade","Price":18.50,"Tags":["new"],"Data":{}},
 {"ProductID":19,"Name":"Helmet","Price":41.99,"Tags":["promo"],"Data":{"Cost":30.65}}
]
```

WHAT DOES IT MEAN?

TWO WORDS:

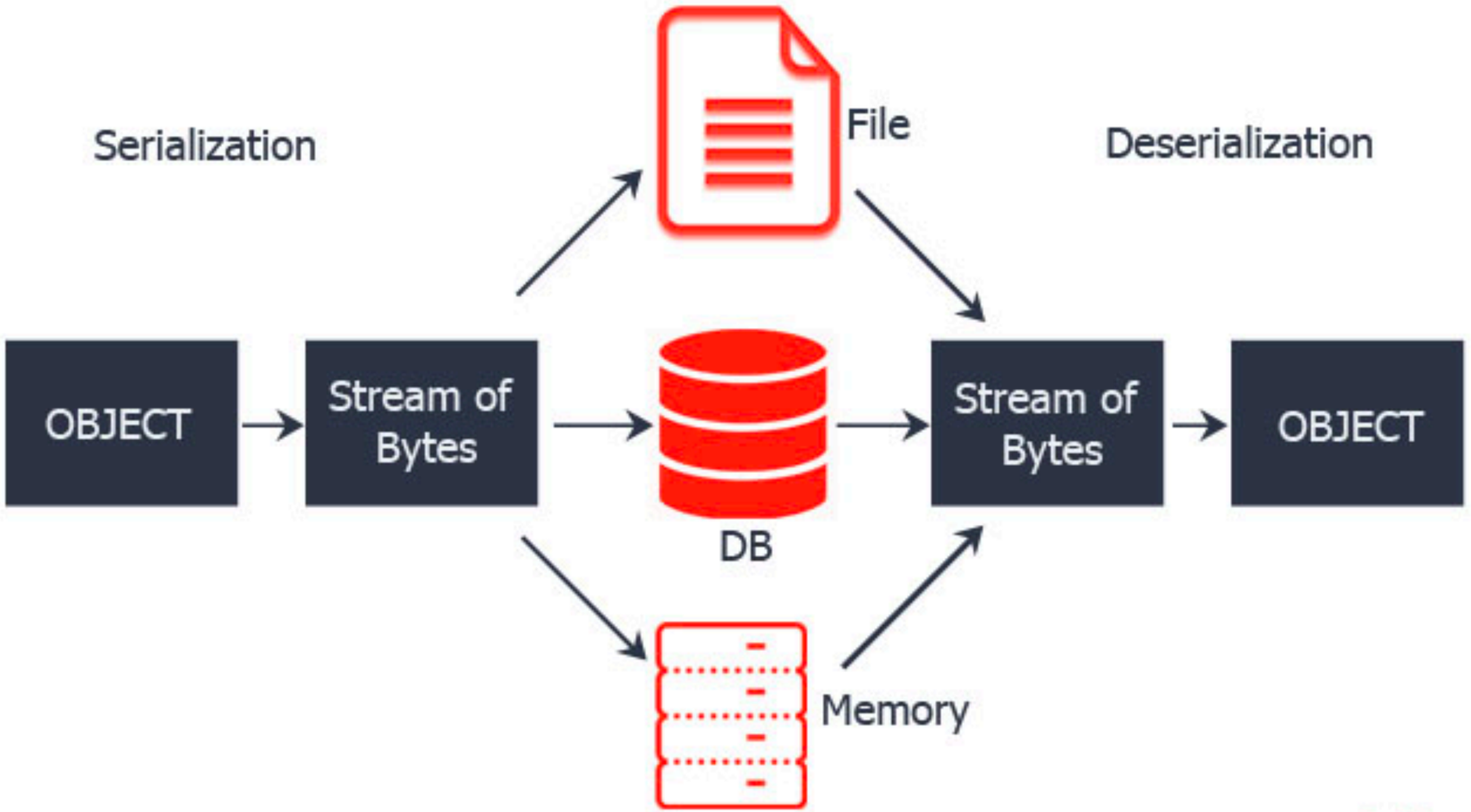
- Serialization.
- Deserialization.


Deserialize
Serialize

SERIALIZATION AND DESERIALIZATION

Serialization is a process of converting an object into a sequence of bytes which can be persisted to a disk or database or can be sent through streams.

The reverse process of creating object from sequence of bytes is called **deserialization**.


Serialization
File
Deserialization
OBJECT
Stream of Bytes
DB
Stream of Bytes
OBJECT
Memory

QUICK DEMO

LET’S TRACK OUR PROCESS:

#1. We are understand “What is an API”.

#2. We know that API TESTING starts with knowing the statuses of the codes.

#3. We know what is “Serialization” and “Deserialization”.

Let’s move on…

CHAPTER #3 -
HOW TO TEST API WITHOUT DOC.?

CHAPTER #4 -
TESTING OF THE REST-API DOCUMENTATION

WHICH TOOLS WE CAN USE TO DOCUMENT OUR API???

- SWAGGER
- APIARY
- API DOCS
- REDOC
- etc.

TOOLS QUICK DEMO

WHAT IS OpenAPI?

OpenAPI Specification (formerly Swagger Specification) is an API description format for REST APIs.

An OpenAPI file allows you to describe your entire API, including:

- Available endpoints (/users) and operations on each endpoint (GET /users, POST /users)
- Operation parameters Input and output for each operation.
- Authentication methods.
- Contact information, license, terms of use and other information.

API specifications can be written in YAML or JSON. The format is easy to learn and readable to both humans and machines.

OpenAPI QUICK DEMO

POST `/store/order` Place an order for a pet

Parameters Try it out

Name	Description
body * required *(body)*	order placed for purchasing the pet **Example Value** | Model `{ "id": 0, "petId": 0, "quantity": 0, "shipDate": "2019-04-28T08:24:51.028Z", "status": "placed", "complete": false }` **Parameter content type** application/json

Responses Response content type application/json

Code	Description
200	*successful operation* **Example Value** | Model `{ "id": 0, "petId": 0, "quantity": 0, "shipDate": "2019-04-28T08:24:58.023Z", "status": "placed", "complete": false }`
400	*Invalid Order*

pet Everything about your Pets

Find out more: http://swagger.io

POST `/pet` Add a new pet to the store

Parameters

Try it out

Name	Description
body * required *(body)*	Pet object that needs to be added to the store **Example Value** | Model

```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "doggie",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Parameter content type: application/json

Responses

Response content type: application/json

Code	Description
405	*Invalid input*

DELETE `/pet/{petId}` Deletes a pet

Parameters Try it out

Name	Description
api_key **string** *(header)*	
petId * required **integer**($int64) *(path)*	Pet id to delete

Responses Response content type application/xml

Code	Description
400	*Invalid ID supplied*
404	*Pet not found*

GET `/store/order/{orderId}` Find purchase order by ID

For valid response try integer IDs with value >= 1 and <= 10. Other values will generated exceptions

Parameters Try it out

Name	Description
orderId * required `integer($int64)` *(path)*	ID of pet that needs to be fetched

Responses Response content type application/json

Code	Description
200	*successful operation* **Example Value** | Model `{ "id": 0, "petId": 0, "quantity": 0, "shipDate": "2019-04-28T08:26:28.634Z", "status": "placed", "complete": false }`
400	*Invalid ID supplied*
404	*Order not found*

#PRACTICE

REST-API TESTING

LET’S TRACK OUR PROCESS:

#1. We are understand “What is an API”.

#2. We know that API TESTING starts with knowing the statuses of the codes.

#3. We know what is “Serialization” and “Deserialization”.

#4. We know that’s all of bugs creates via developers…

#5. You need turn ON your brain during the testing process.

Let’s move on…

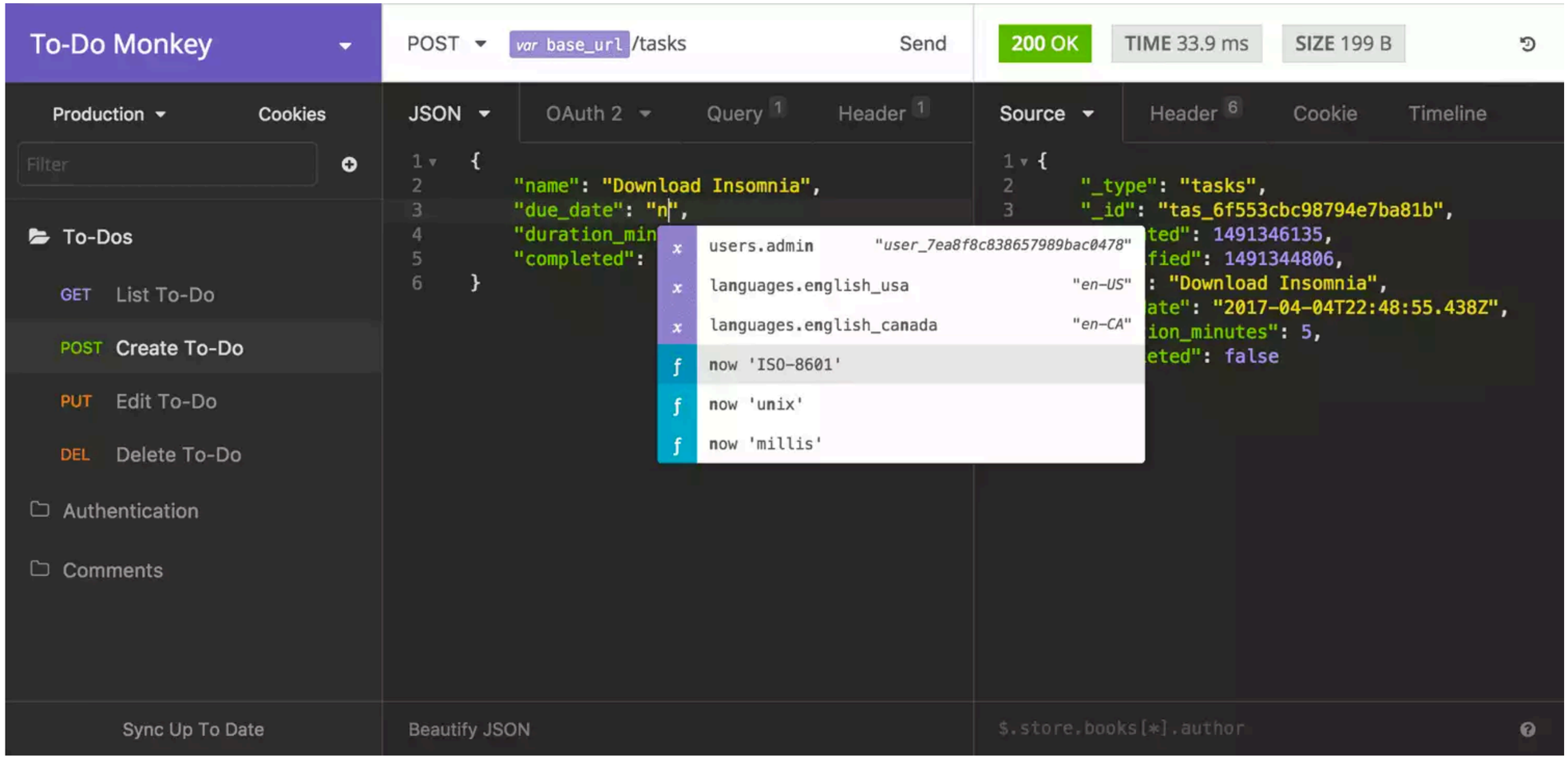
CHAPTER #5 -
TOOLS

UI-BASED TOOLS


BE OUR GUEST, BE OUR GUEST
PUT OUR POSTMAN TO THE TEST
imgflip.com

INSOMNIA

Debug APIs like a human, not a robot.



COMMAND LINE HTTP CLIENTS

HTTPie

cURL

CHAPTER #6 -

CORRECT APPROACH TO REST-API TESTING

ISTQB???

CHAPTER #6.2 -
XHR VS FETCH

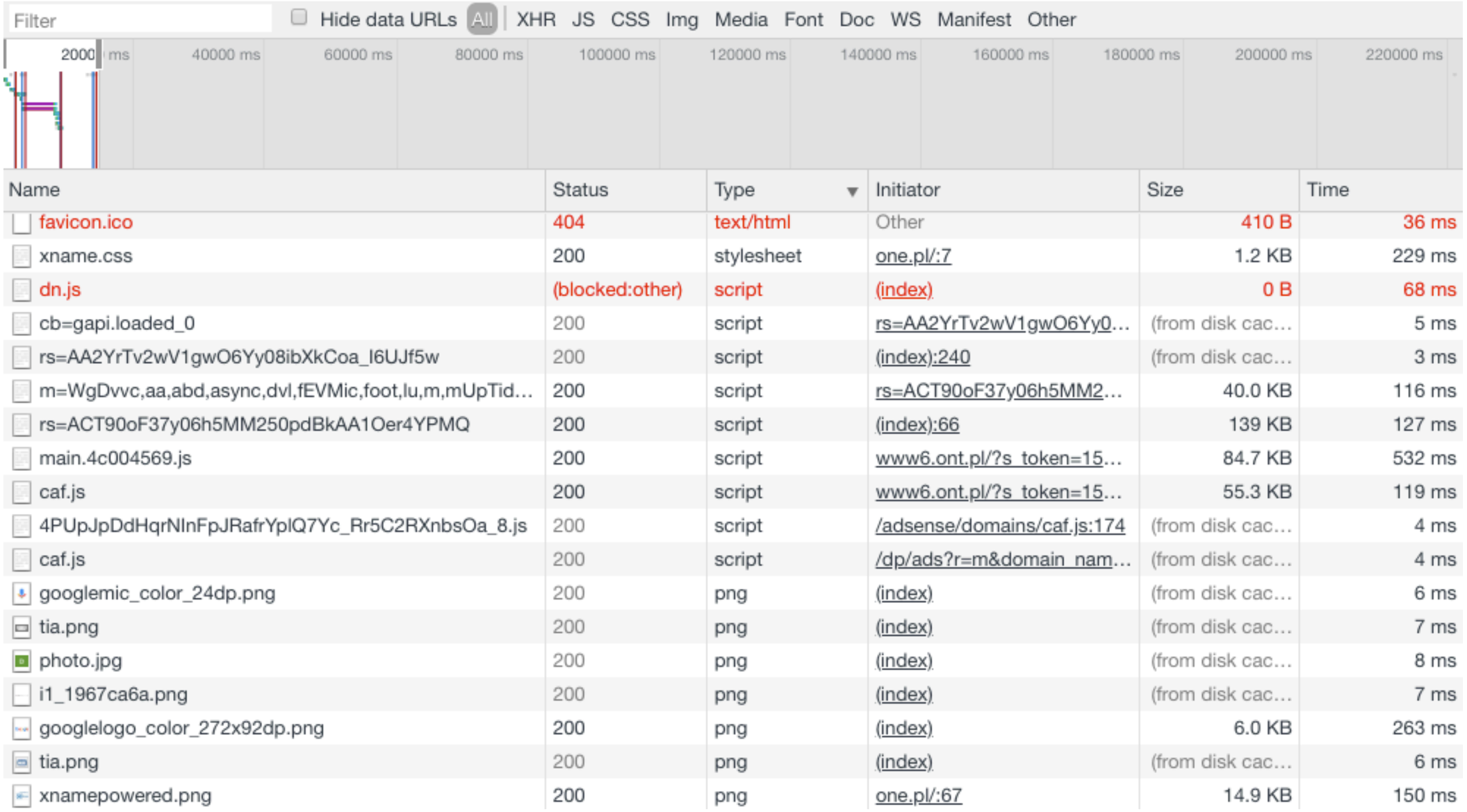
BUT FIRST…

WE SHOULD UNDERSTAND…

WHAT TYPE OF TESTERS YOU ARE?

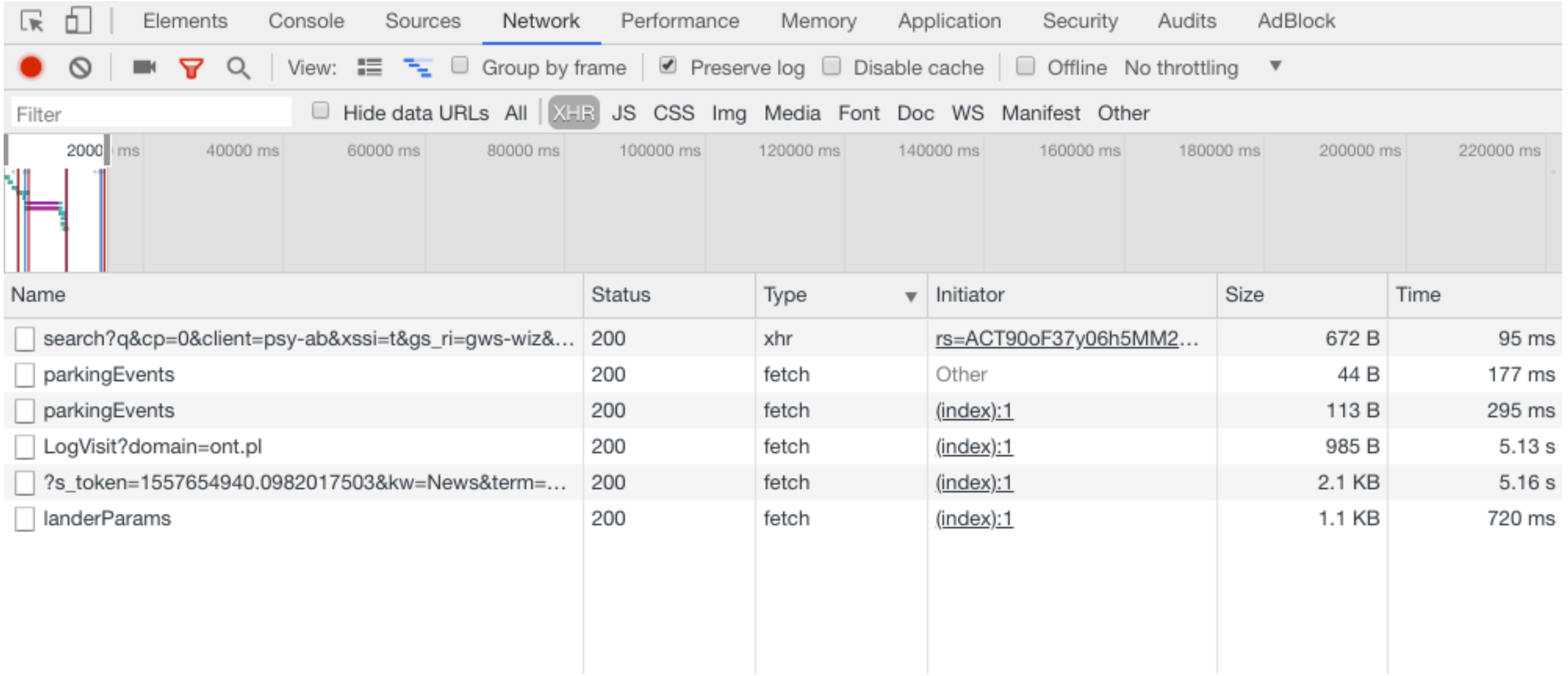
OPEN BROWSER
DEV.TOOLS

SMOKER'S DEV.TOOLS



Name	Status	Type	Initiator	Size	Time
favicon.ico	404	text/html	Other	410 B	36 ms
xname.css	200	stylesheet	one.pl/:7	1.2 KB	229 ms
dn.js	(blocked:other)	script	(index)	0 B	68 ms
cb=gapi.loaded_0	200	script	rs=AA2YrTv2wV1gwO6Yy0…	(from disk cac…	5 ms
rs=AA2YrTv2wV1gwO6Yy08ibXkCoa_I6UJf5w	200	script	(index):240	(from disk cac…	3 ms
m=WgDvvc,aa,abd,async,dvl,fEVMic,foot,lu,m,mUpTid…	200	script	rs=ACT90oF37y06h5MM2…	40.0 KB	116 ms
rs=ACT90oF37y06h5MM250pdBkAA1Oer4YPMQ	200	script	(index):66	139 KB	127 ms
main.4c004569.js	200	script	www6.ont.pl/?s_token=15…	84.7 KB	532 ms
caf.js	200	script	www6.ont.pl/?s_token=15…	55.3 KB	119 ms
4PUpJpDdHqrNInFpJRafrYplQ7Yc_Rr5C2RXnbsOa_8.js	200	script	/adsense/domains/caf.js:174	(from disk cac…	4 ms
caf.js	200	script	/dp/ads?r=m&domain_nam…	(from disk cac…	4 ms
googlemic_color_24dp.png	200	png	(index)	(from disk cac…	6 ms
tia.png	200	png	(index)	(from disk cac…	7 ms
photo.jpg	200	png	(index)	(from disk cac…	8 ms
i1_1967ca6a.png	200	png	(index)	(from disk cac…	7 ms
googlelogo_color_272x92dp.png	200	png	(index)	6.0 KB	263 ms
tia.png	200	png	(index)	(from disk cac…	6 ms
xnamepowered.png	200	png	one.pl/:67	14.9 KB	150 ms

HEALTHY PERSON DEV.TOOLS



CHAPTER #6.3 -
REST-API TEST STRATEGY

HAPPY PATH - FIRST

NEGATIVE CASES - SECOND

STATUS CODES

401 / 403 - UNAUTHORIZED & FORBIDDEN

404 - NOT FOUND

405 - METHOD NOT ALLOWED

400 - BAD REQUEST

409 - CONFLICT

422 - UNPROCESSABLE ENTITY

413 / 415 / 418 / 429 etc.

UNNEEDED DATA

CHAPTER #6.4 -
“FROM UP TO DOWN”

#DEMO

REST-API TESTING

#PRACTICE

REST-API TESTING

CHAPTER #7 -
LET’S review our BUG’s

CHAPTER #8 -
TIPS & TRICKS

HOW TO WORK PRODUCTIVELY?

HOW TO SAVE YOUR TIME?

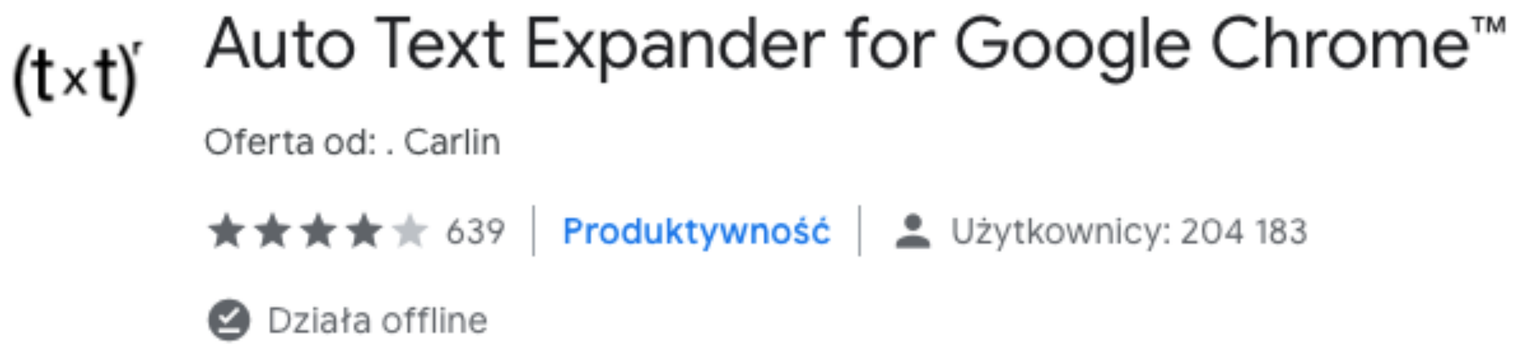
BUG-REPORTING AS ART

YOU SHOULD CARE ABOUT:

1. YOUR SELF.
2. YOU BACKEND-DEVELOPER.
3. YOU FRONT-END DEVELOPER.

BUT, WHAT I SHOULD TO DO?

1. INSTALL AUTO TEXT EXPANDER FOR GOOGLE CHROME



2. USE CORRECT APPROACH TO REPORT BUGS.

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Expected results: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Steps to reproduce:
1. Lorem ipsum dolor sit amet.
2. Lorem ipsum dolor sit amet.
3. Lorem ipsum dolor sit amet.
4. Lorem ipsum dolor sit amet.
5. Lorem ipsum dolor sit amet.
6. Lorem ipsum dolor sit amet.

Additional info:
Request URL: https://some-url.com/catalogue?page=1&size=6&tags=
Request Method: GET
Status Code: 400 Bad Request

```
curl 'http://35.225.108.2/catalogue?page=1&size=6&tags=' -H 'Cookie: _TRAEFIK_BACKEND=http://front-end:8079; md.sid=s%3AoXcYNVUyFEQPggko87dab4f9jcauZvwq.kRbPElYT5T4mPZsW3fxgLqEWi3LmfPGJc%2BfNjh%2FivSM' -H 'Accept-Encoding: gzip, deflate' -H 'Accept-Language: pl-PL,pl;q=0.9,en-US;q=0.8,en;q=0.7,ru;q=0.6' -H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36' -H 'Accept: application/json, text/javascript, */*; q=0.01' -H 'Referer: http://35.225.108.2/category.html' -H 'X-Requested-With: XMLHttpRequest' -H 'Connection: keep-alive' --compressed
```

ENVIRONMENT

Env: https://some-url.com [QA]
User: user1
Browser: Chrome Version 72.0.3626.121 (Official Build) (64-bit)
Cache: Removed

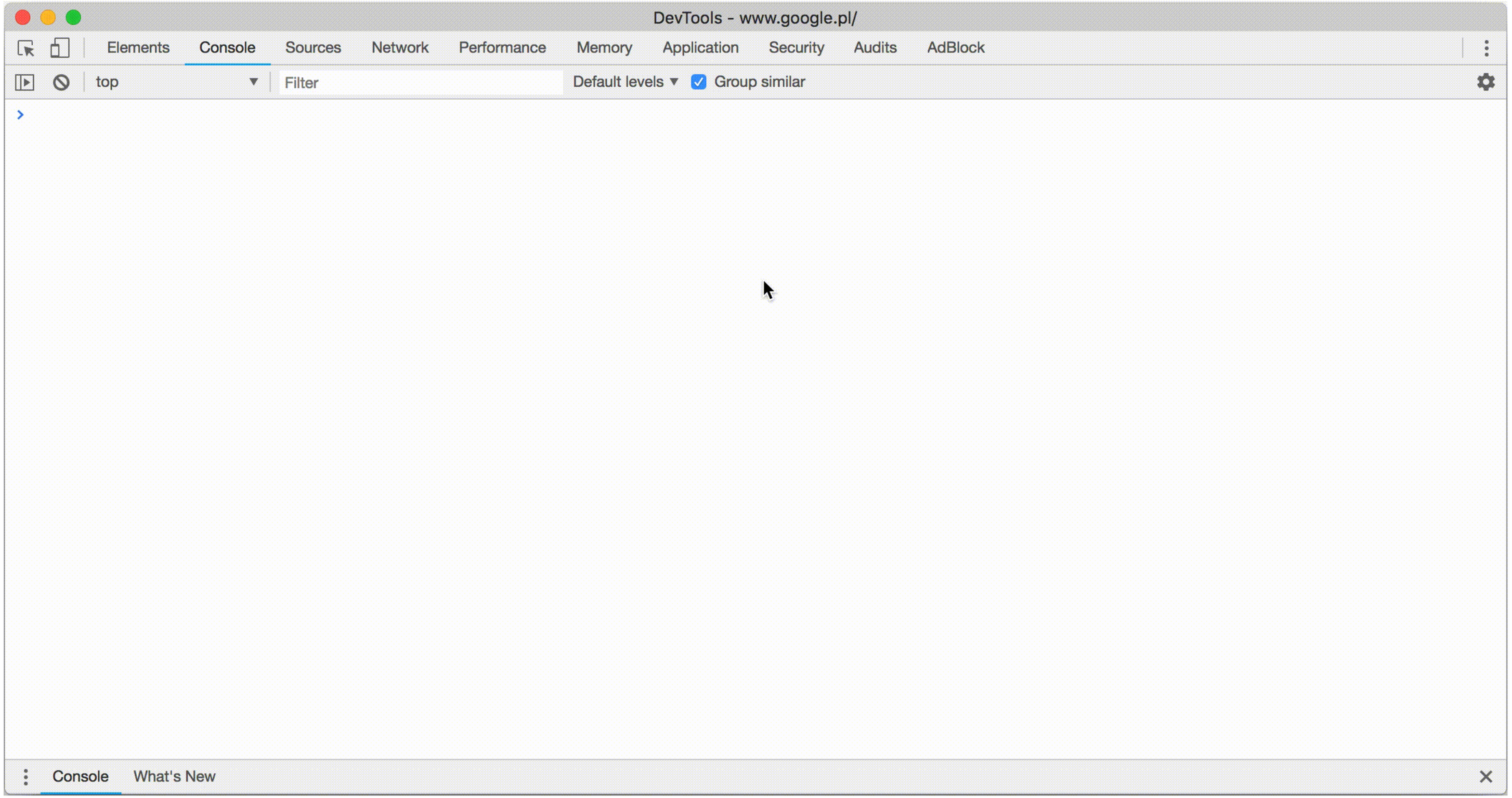
#klikajCurŁa

DON’T CHANGE YOUR CONTEXT

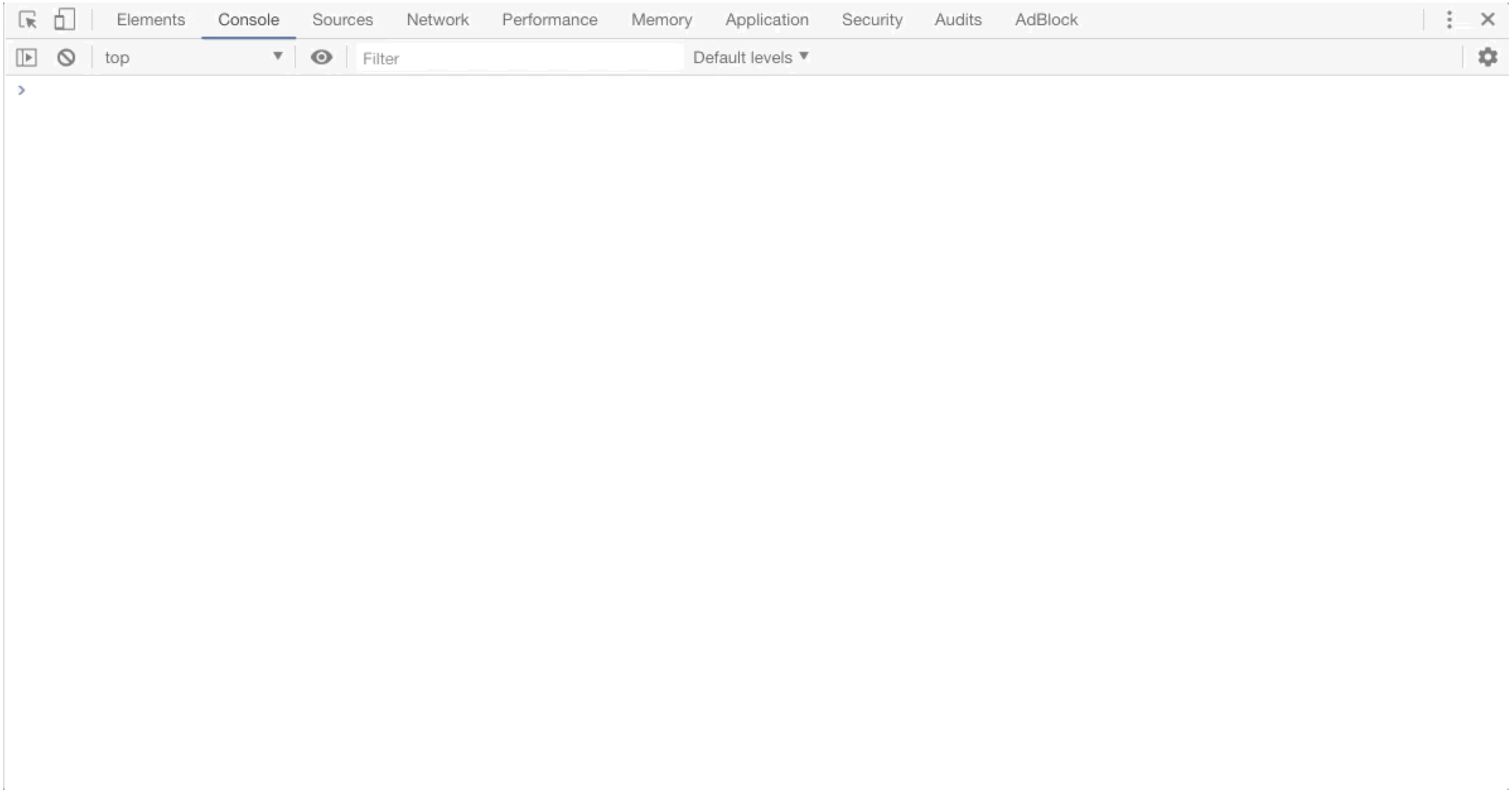
BROWSER DEV.TOOLS

USE COMMAND MENU

This tip allows you to display list of options available in chrome, such as show timestamp in console, frames per second, switch to dark theme and more!
You can display options list by pressing Ctrl + Shift + P (Cmd + Shift + P for iOS).



USE CONSOLE TO
SEND REQUESTS


Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
AdBlock
top
Filter
Default levels

THANK YOU!

HAPPY TESTING!

2019-12-01 - Kraków

Warsztat #327 Podstawy uczenia maszynowego: pierwszy projekt od A do Z

Warsztat płatny - od 79 PLN, 1 edycja warsztatu

ZAREJESTRUJ SIĘ

2019-12-07 - Kraków

Warsztat #331 Praktyczne sposoby wykrywania podatności w aplikacjach webowych

Warsztat płatny - od 79 PLN, 4 edycja warsztatu

ZAREJESTRUJ SIĘ