Research Software Summer School: APIs & web scraping

Jelte van Boheemen
Research Software Engineer

Obtaining online data

- Ask nicely
- Interact with Application Programming Interface (API)
- Scrape

API vs web scraping

API	scraping
Neat data, methods	Messy data, methods
Legal, intended way	Legally grey
Possibly restricted, paid	Free, Libre, Gratis
Needs to exist	Remaining option

Application Programming Interface

- Software communicates with software
- Not necessarily online
  - App accesses camera through API
- Interface
  - Underlying system does not matter
  - Analogy: order at a restaurant

HTTP Request

- HyperText Transfer Protocol

- Client *requests* at *endpoint*
- API replies with *response*

- HTTP methods:
  - **GET, POST, PUT, DELETE**
  - more exist

HTTP Response

- 2xx: Succes
- 4xx: Client error (your fault)
- 5xx: Server error (not your fault)

- Common response data formats:
  - JSON
  - Image
  - HTML

JSON

- JavaScript Object Notation

- Nested data format
- Human readable

- Standard for data interchange
- Widely supported

- In Python: **dict** and/or **list**

API endpoints

- Location to access resource(s)

- REST
  - Representational State Transfer
  - convention for architecture of endpoints

  - GET /articles → list all articles
  - GET /articles/5 → get article 5
  - POST /articles → create article
  - DELETE /articles/5 → delete article 5

Additional information in requests

- GET: Query parameters
  - 0 or more
  - refine requests
  - https://www.fruityvice.com/api/fruit/fat?min=5
- POST:
  - send data

Headers

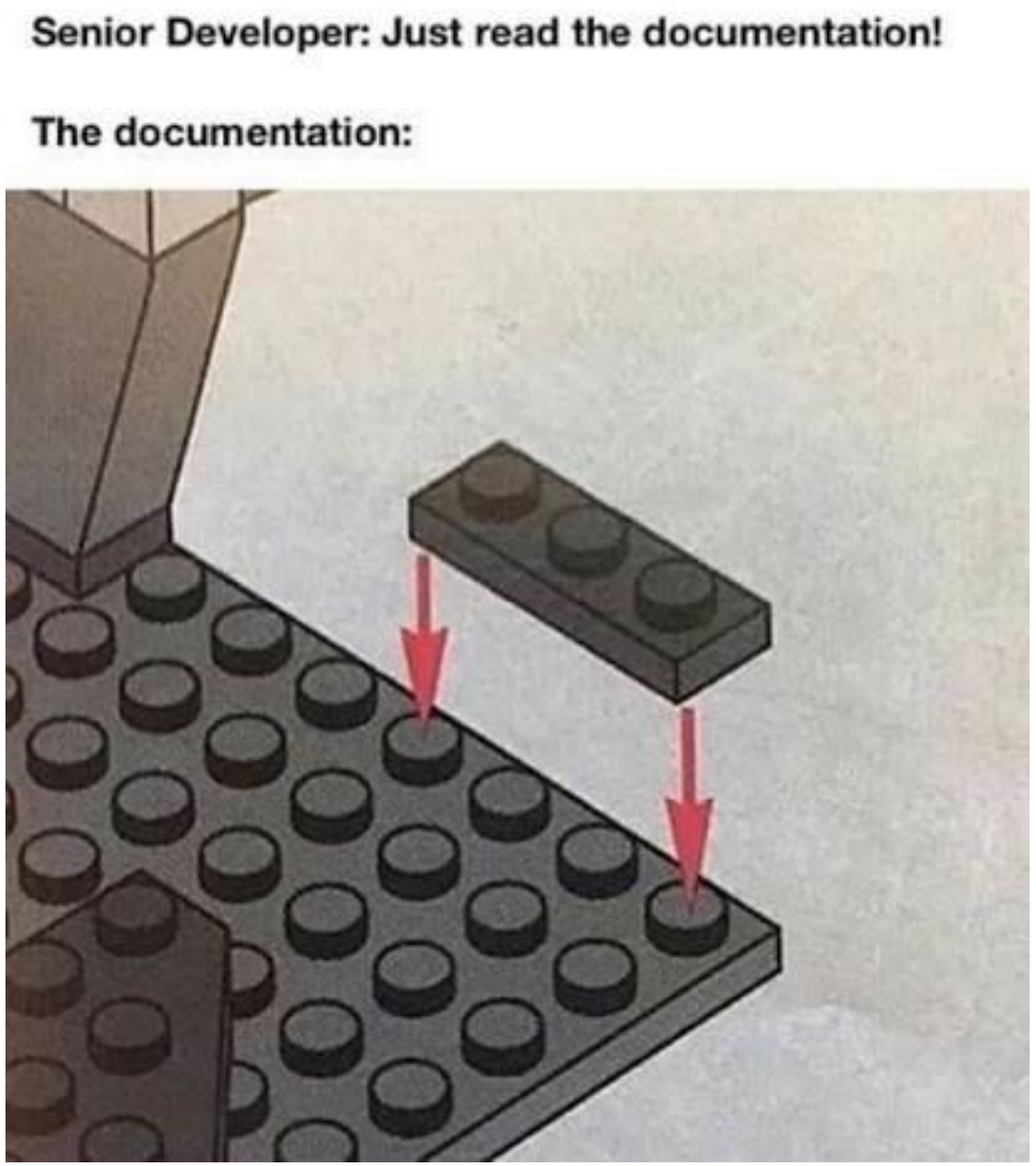
- ”settings” for HTTP requests and responses
- examples:
  - Authorization
  - content-type
  - Accept
  - User-Agent
- defaults will work for today

Making a GET request

- curl <url>
  - MacOS/Linux or Git Bash
- Browser renders **JSON**
- Displaying a website
  - **HTTP GET** request
  - **HTML** response
  - Browser renders

API access

- API needs to exist
- Sometimes: **API key** required
- Sometimes: payment required
- Sometimes: access limited
- Documentation: present and correct?



Exercise time!

- Day 4 > APIs > Interacting with an API

    - Play around with **fruityvice.com**
    - Find the docs first

    - 10 minutes

APIs using Python

- Python can make HTTP requests
- Standard library **urllib** is meh

- **requests** package is great

- Demo time

Exercise time!

- Day 4 > APIs > Using Python to interact with API

  - Play around with **fruityvice.com**
    - But now in Python

  - 15 minutes

Utrecht University

Sharing science,
shaping tomorrow