Processing Web Data With XML And XSLT

Bogdan Dumitru, Syncro Soft

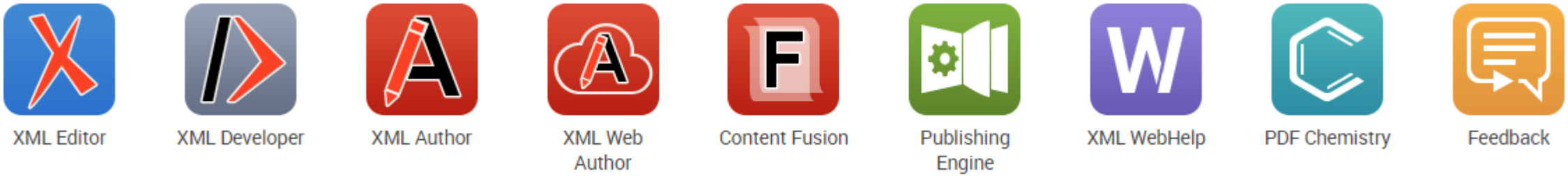
bogdan_dumitru@sync.ro

Open4Tech Summer School, 2020

About Us

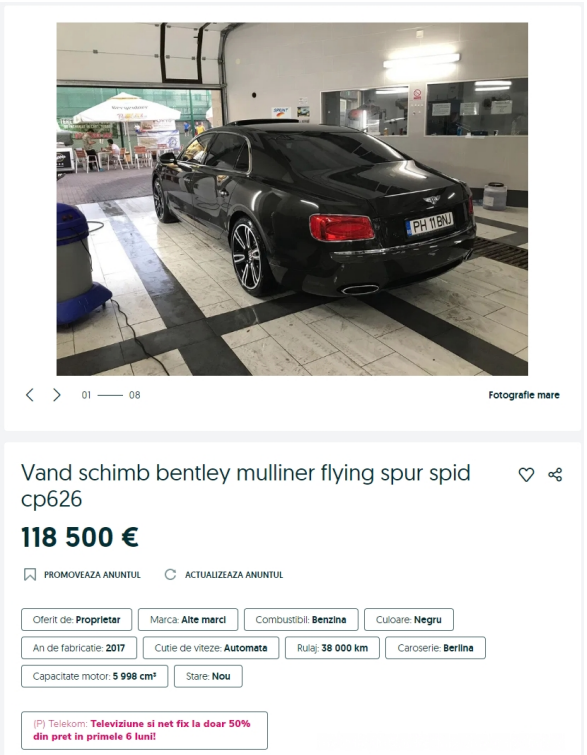
Syncro Soft

- Offer solution for XML authoring and publishing
- Have both stand-alone and web-based applications
- See more at sync.ro, oxygenxml.com



Goal

Extract data from a public site and store it in a well formatted structure.




```
<?xml version="1.0" encoding="UTF-8"?>
<ad>
   <title> Schimb Daewoo Cielo </title>
   <spec type="Oferit de">Proprietar</spec>
   <spec type="Marca">Daewoo</spec>
   <spec type="Model">Cielo</spec>
   <spec type="Culoare">Albastru</spec>
   <spec type="Combustibil">Benzina</spec>
   <spec type="Cutie de viteze">Manuala</spec>
   <spec type="An de fabricatie">2004 </spec>
   <spec type="Rulaj">145 264 km</spec>
   <spec type="Caroserie">Berlina</spec>
   <spec type="Capacitate motor">1 500 cm</spec>
   <spec type="Stare">Utilizat</spec>
   <description>

          Schimb Cielo mai ofer 1000 de lei diferenta
        . Masina se afla in stare buna
      VIN: WDDDJ72X97A116339
        . Consumabile recent schimbate
        . Ulei
        . Filtru de ulei
        . Filtru de aer
        . Filtru de combustibil
        . Pivoti
        . Bielete
        . Capete de bara
        . Bucsele
        . Rulmenti la roata
        . Placutele de frana
        . Discurile de frana
        . Baterie NOUA
        . Distributia schimbata la 143100km
        . Se vinde cu jentile de aluminiu cu cauciucurile de vara+inca un set de roti
        . Are Motor de 1.5 ecotec 8v de 80cp
        . Consuma foarte putin 5.5 mixt la proba!
        . Ca dotari are inchidere centralizata
        . Deschidere buson la buton
        . Sistem Audio HERTZ suna bestial
```

Note:

- GitHub repository: https://github.com/dumitrubogdanmihai/processing-web-data-with-xml-and-xslt
- Questions

About

The course will have two parts:

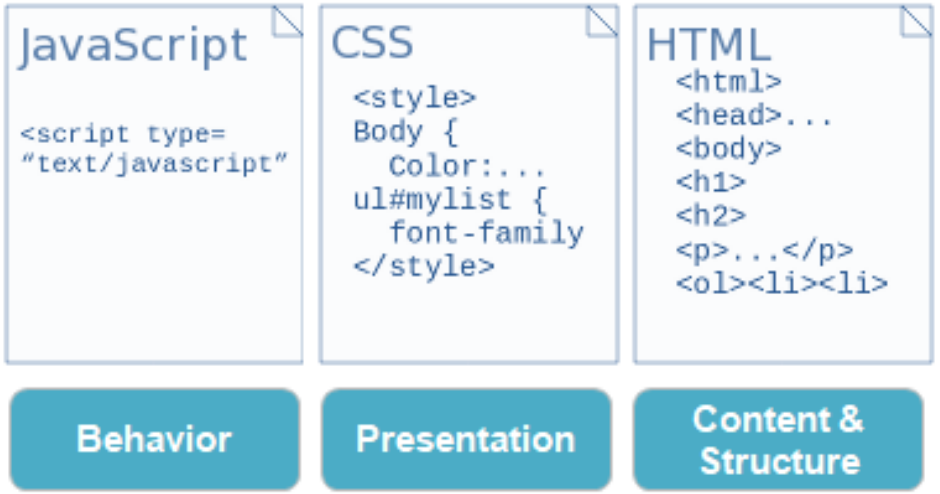
- Day 1: acquire data
  - design and implement the crawler
- Day 2: process data
  - store raw data in XML
  - process data with XSLT
  - query it by using XPath

Today's Agenda

- How Browsers Work
- What is a Web Crawler
- How to Control Browsers with Selenium
- Live Coding

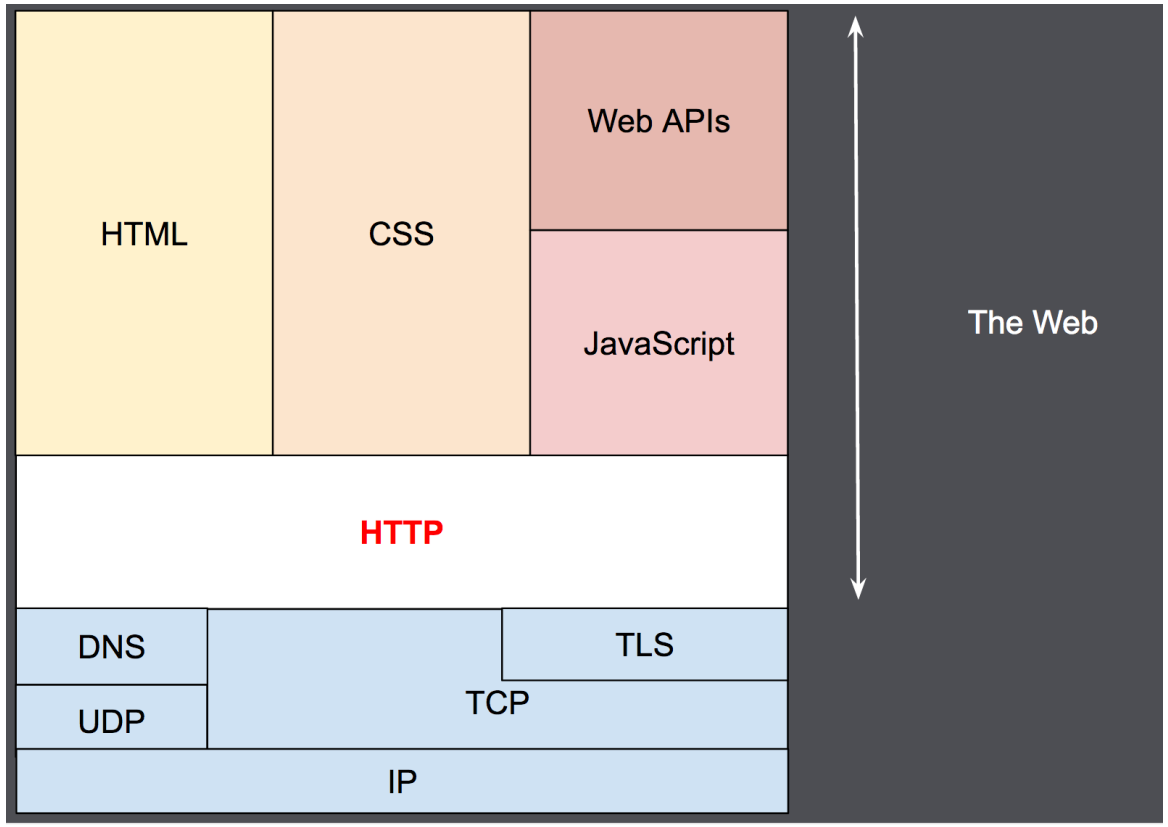
How Browsers Work?

- HTML (Hypertext Markup Language)
  - define structure and/or content
- CSS (Cascading Style Sheets)
  - define rendering
    - colors, sizes, fonts, etc
- JS (JavaScript)
  - define behavior
    - what function to call when a button is pressed



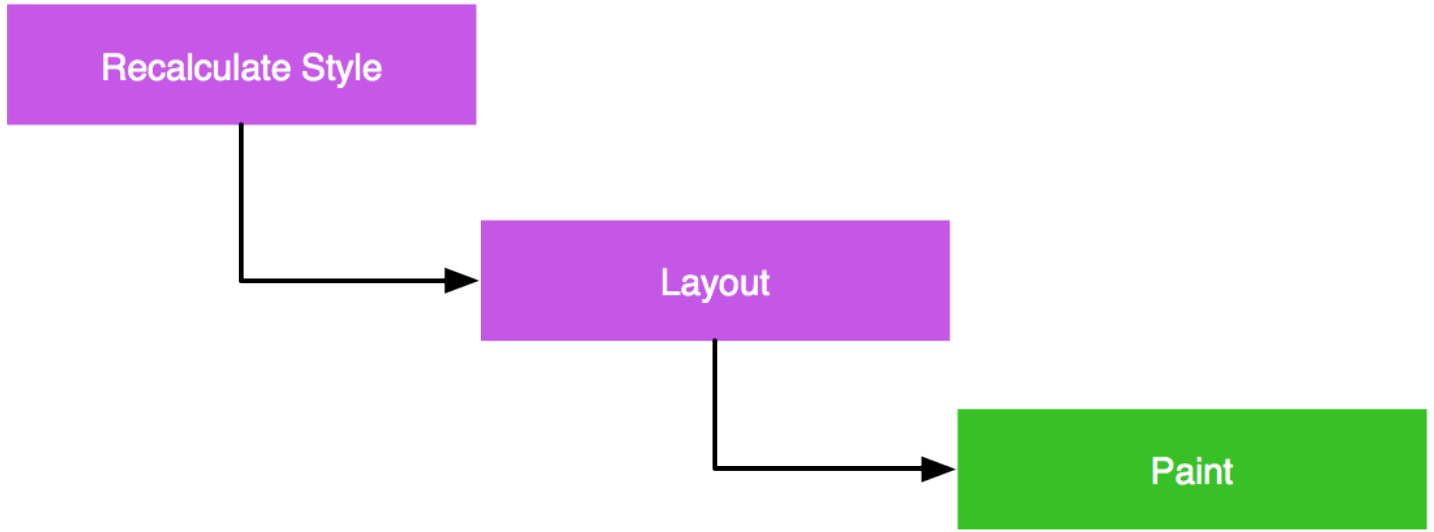
How Browsers Work?

- Rendering process:
  - retrieve HTML
  - retrieve CSS and JS
  - execute JS
  - compute styles
  - do layout
  - paint



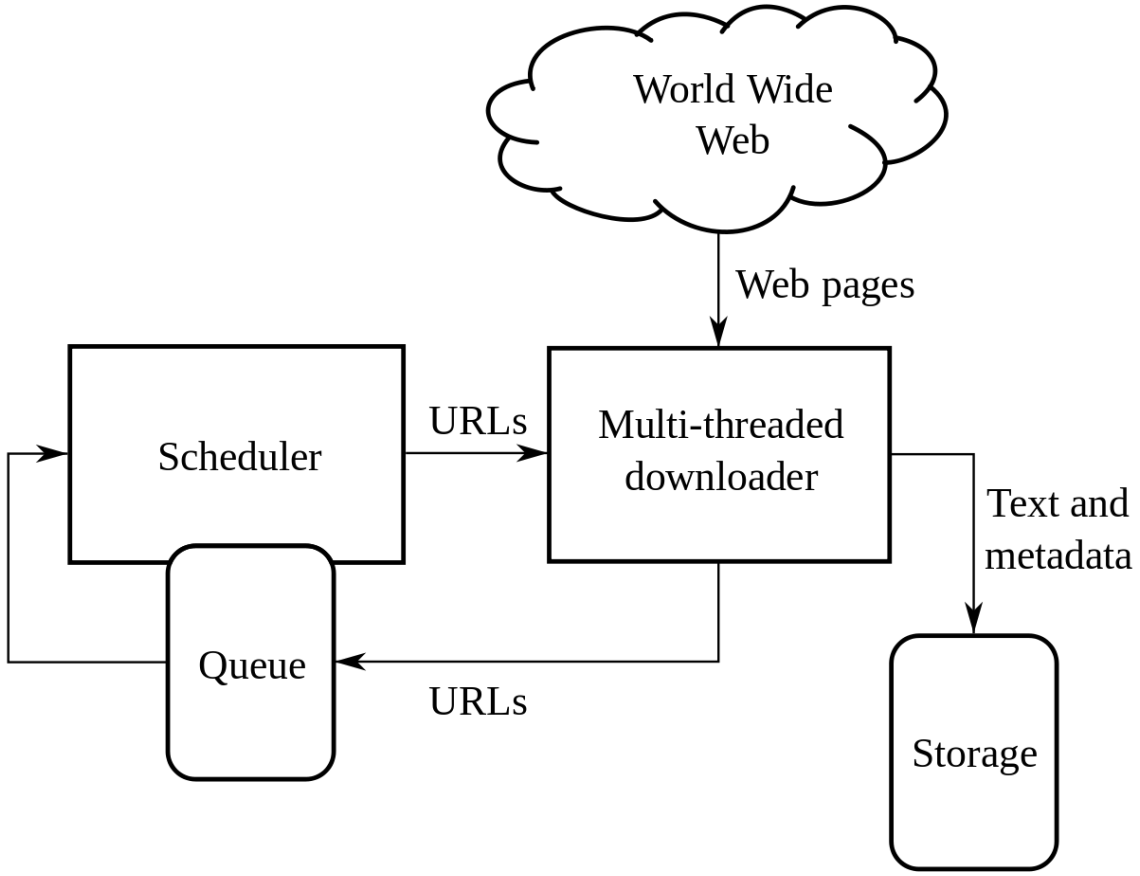
How Browsers Work?

- Rendering process



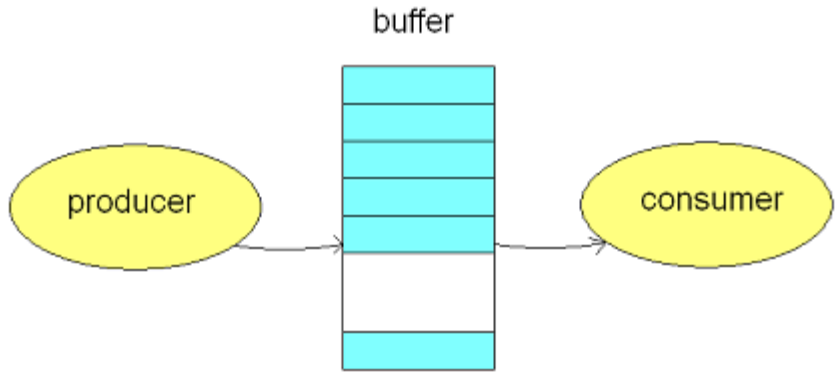
What is a Web Crawler

- A web crawler (or spider) is an agent that browse the web with the purpose of extracting or indexing data.
- It may or may not be bounded to a specific domain.



Crawler Design

- Producer
  - generates/supply new URLs of interest from where data is extracted
- Buffer
  - store the target URLs
  - must be thread-safe
  - usually it is a Queue data structure
- Consumer
  - extract data from each URL popped from buffer



How to get data from a web site?

Why basic HTTP requests aren't enough?

- Rarely websites sent data together with the HTML
  - usually the meaningful data is retrieved through async requests
  - to render First Meaningful Paint ASAP
  - to render data in chunks
- `curl -vvv https://www.olx.ro/`

How to get data from a web site?

The safe approach:

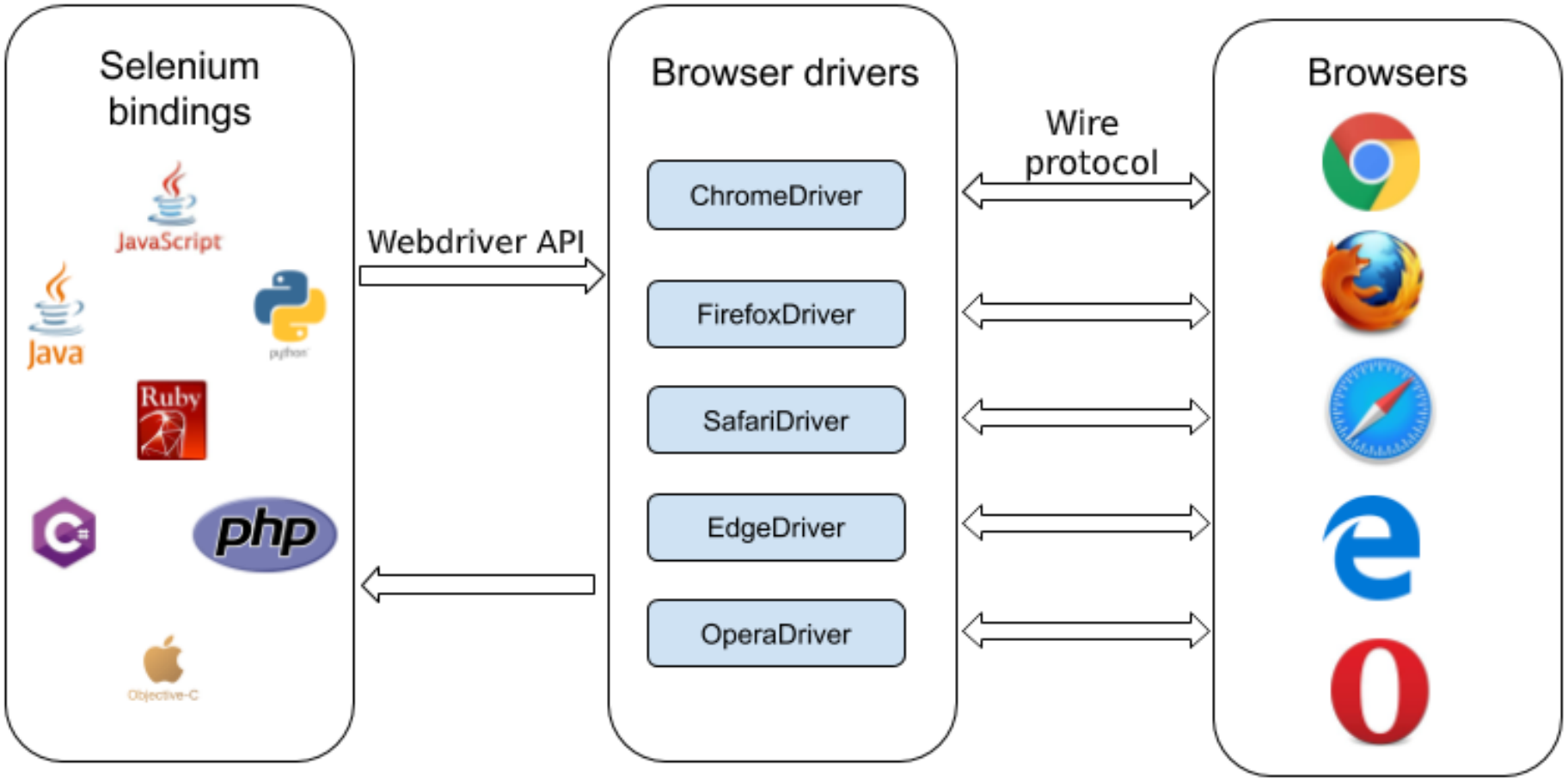
- Let the browser completely load the page
  - (all requests are finished)
- Get the whole HTML or just parts of it

How to Control Browsers

Selenium

- Can control browsers:
  - open an URL
  - click a button
  - etc.
- Retrieve data from browsers:
  - page content
  - position or size of an element
  - etc.

How Selenium Works



Important Note!

Each specific WebDriver version have a bounded interval of browser versions that is compatible with.

Firefox:

https://firefox-source-docs.mozilla.org/testing/geckodriver/Support.html

Chrome:
https://sites.google.com/a/chromium.org/chromedriver/downloads

Other Use Cases

- Web crawling
- Web indexing
- Automated testing

Configure Project

Prerequisites

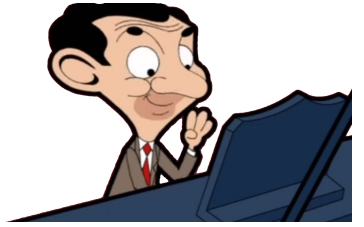
- JDK (1.8 or higher)
- Maven

Configure Project

- Google Chrome (v83.0)
  - https://www.google.com/chrome/
- Chrome Driver (v83.0)
  - https://chromedriver.chromium.org/downloads
  - set the `webdriver.chrome.driver` system property
- Selenium Java (v3.1)
  - pom.xml
    - `<dependency>`
    - `    <groupId>org.seleniumhq.selenium</groupId>`
    - `    <artifactId>selenium-java</artifactId>`
    - `    <version>3.141.59</version>`
    - `</dependency>`

Let's Code

- Define interfaces for producer and consumer
- Implement them



Let's Test It

We'll add a few unit tests

- JUnit
- pom.xml:
  - `<dependency>`
  - `<groupId>junit</groupId>`
  - `<artifactId>junit</artifactId>`
  - `<version>4.12</version>`
  - `<scope>test</scope>`
  - `</dependency>`

Made it faster

- run headless
- use fast selectors
- re-use browser instance
- pre-populate cookies
- do not load images

What can go wrong?

- Ban
  - IP ban
  - Geo-restrictions
  - Cookie restriction
  - Honeypot traps
- Captcha
- Changes in URL scheme or in HTML structure

What to prevent wrong things to happen?

- Ban
  - respect robots.txt
  - change User Agent
  - clear cookies
- Captcha
  - it's immoral to bypass captchas
  - it may be even illegal to do so
- Changes in URL scheme or in HTML structure
  - automated tests

Bonus

How to write tests using Selenium

- Open the page
  - `driver.get("https://localhost:8080")`
- Do an action
  - `driver.findElement(By.cssSelector(".class-name"))`
  - `element.click()`
- Assert the expected outcome
  - `assertEquals("Expected State", element.getText())`

THANK YOU!

Any questions?

Bogdan Dumitru

bogdan_dumitru@sync.ro