Browser Automation and Web Modification Introduction

•••

edde.yang@activenetwork.com

Agenda

- Why we need to automate browsers and modify web pages
- Common tools to automate browsers and modify web pages
    - Selenium
    - Headless browsers
        - SlimerJS, PhantomJS
        - CasperJS, nightmare
    - iMacros
    - Browser plugins
        - GreaseMonkey, TamperMonkey
        - Stylish

Why we need to automate browsers and modify web pages

- Repetitive tasks are not interesting and time-consuming
- Humans tend to make errors when doing repetitive tasks
- We can save time to do more important stuff, like improve your product
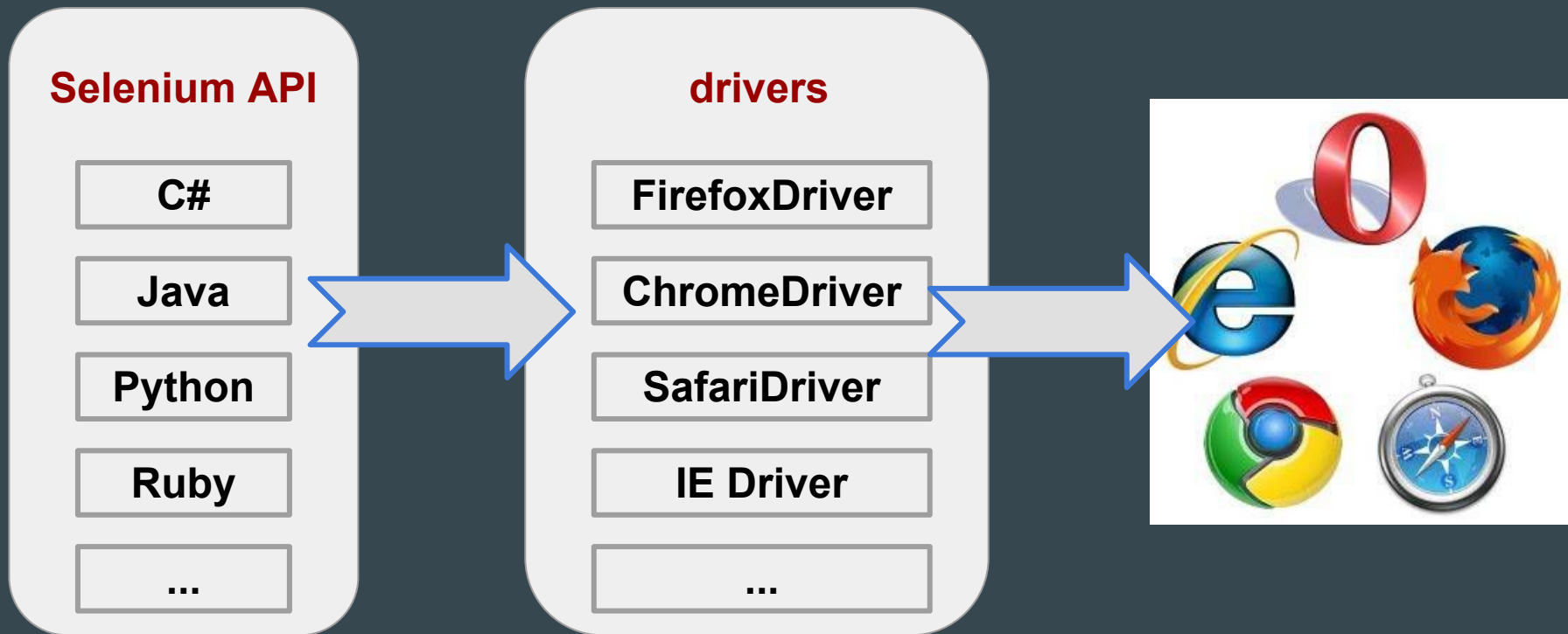- We hope the web page displays and behaviors the way we want it to

Problems we might face

- We are going to make a trip and we have several destinations. Find the cheapest flight ticket.
- After each production release we need to do a quick smoke test to ensure the application works as expected. The process is exactly the same. Log into the system, create a season, create several sessions under the season, maybe create a product, buy something and checkout.
- We were given an Excel file with 100 rows of customer information. We were asked to create users in a system using these info. That means to need to repeat the same process 100 times.

Selenium

- A testing framework created in 2004 from ThoughtWorks
- Supports multiple programming languages
- Supports all major browsers
- Free and powerful automation solution

Selenium Architecture

Selenium and Drivers

- Selenium
  - http://www.seleniumhq.org/download
- Chrome driver
  - https://chromedriver.storage.googleapis.com/index.html?path=2.27/
- Firefox driver
  - https://github.com/mozilla/geckodriver/releases

Selenium Demo

Headless Browsers

- A headless browser is a web browser without GUI
- You can automate headless browsers using the interfaces they provided
- Useful for testing
- Famous headless browsers
  - PhantomJS
  - SlimerJS
  - CasperJS
  - nightmare

PhantomJS

- A WebKit-based browser with no GUI (headless browser)
- Provides API that allows us to do anything we would do with a normal browser
- Allows us to programmatically manipulate web page content or scrape data
- It supports web standards which means you can use web familiar technologies like CSS selector, DOM manipulation, JSON, HTML5, Canvas, SVG etc.
- People use PhantomJS to do many different tasks like:
  - Repetitive browser related task
  - Automation testing and JavaScript unit testing
  - Continuous integration
  - Scraping data
  - ...

SlimerJS

- Is a headless browser
- Very similar to PhantomJS
- Differences between the two
  - PhantomJS is based on WebKit (Safari, Chrome), SlimerJS is based on Gecko (Firefox)
    - WebKit powers Safari
    - A fork of WebKit named Blink powers Chrome
  - Most of the API is the same, but there are still some differences
  - SlimerJS is not completely headless
  - You can use coffeescript only with SlimerJS
  - SlimerJS supports Flash
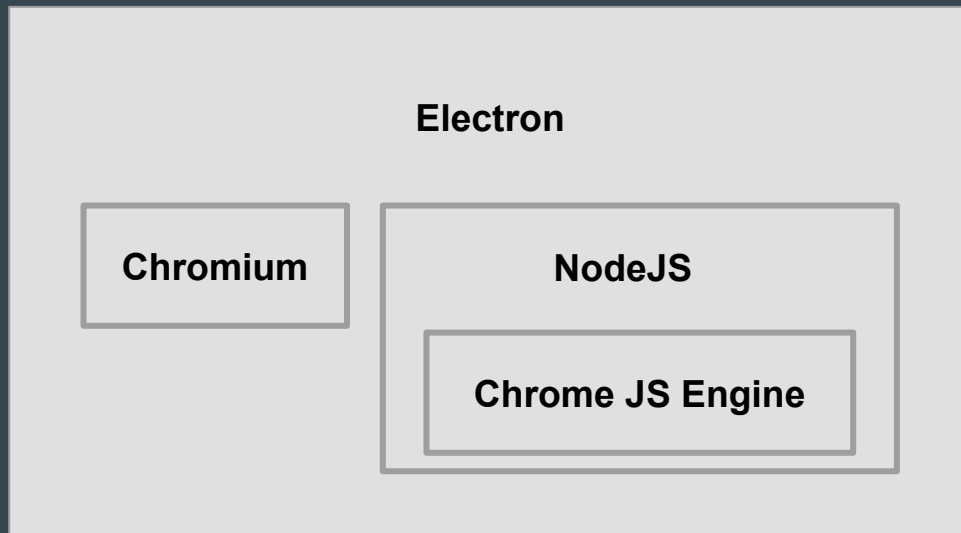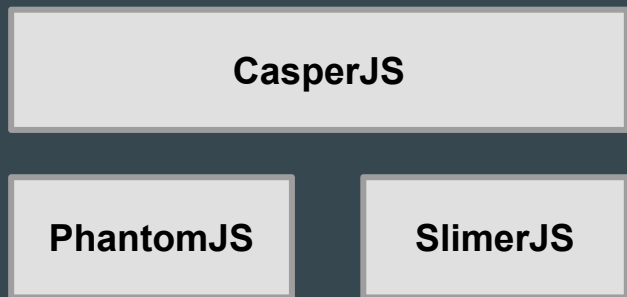 - PhantomJS is more popular

CasperJS

- PhantomJS's API is not easy to use
- CasperJS uses PhantomJS or SlimerJS under the hood
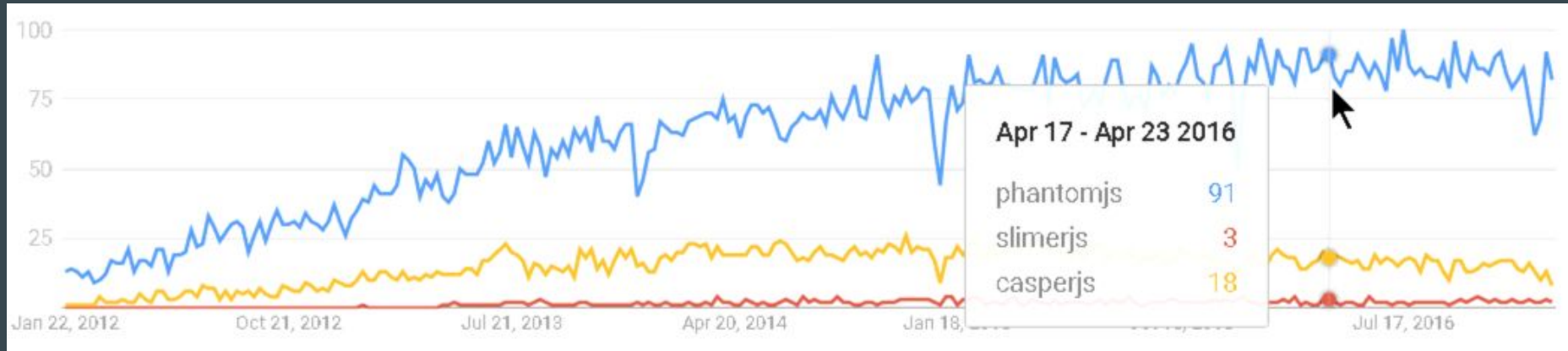- CasperJS provides a set of easy-to-use API

nightmare

- nightmare is similar with CasperJS
- It provides a set of easy-to-use API
- nightmare 1.0 use PhantomJS under the hood
- nightmare 2.0 use Electron underneath
  - Electron = Chromium + NodeJS
  - Build cross-platform desktop apps using web technologies (JS, CSS, HTML)
  - Comes from GitHub
  - Used by many famous projects
    - Atom
    - Visual Studio Code
- daydream Chrome extention

Headless Browsers

CasperJS

PhantomJS

SlimerJS

Electron

Chromium

NodeJS

Chrome JS Engine

Google Trends Comparison

Headless Browsers Download and Installation

- NodeJS
  - https://nodejs.org/en/download/
- PhantomJS
  - npm install phantomjs
- SlimerJS
  - npm install slimerjs
- CasperJS
  - npm install capserjs
- Nightmare
  - npm install nightmare
  - Electron will be install by running this command

Nightmare Demo

iMacros Introduction

- iMacros is browser plugins from Ipswitch, inc.
- iMacros can record and replay repetitive tasks in browsers
- Supports major browsers: Chrome, IE, Firefox
- The project starts in 2007

iMacros related scripts

- Ask Firefox to run a iMacros script
  - "C:\Program Files (x86)\Mozilla Firefox\firefox.exe" imacros://run/?m=Demo-FillForm.iim
  - http://wiki.imacros.net/iMacros_for_Firefox#Command_Line_Support
- Save web page data to a file
  - SAVEAS command
  - http://wiki.imacros.net/SAVEAS

iMacros Demo

GreaseMonkey, TamperMonkey & Stylish

- Web pages are displayed locally in our browsers, which means we can modify them if needed
- Modify appearance
    - Stylish
- Modify behavior
    - GreaseMonkey
    - TamperMonkey

Demo

Thank you!