Bot Academy

2015

André Costa

@Lokal_Profil

Stockholm, 2015-11-28

bit.ly/BotAcademy2015

WIKIMEDIA
SVERIGE

__TOC__

1. Bot Philosophy
2. Bot Etiquette
3. The API(s)
4. Tools & Frameworks
5. Tool Labs
6. Housekeeping

Bot Philosophy

Why, what and When (not) to bot

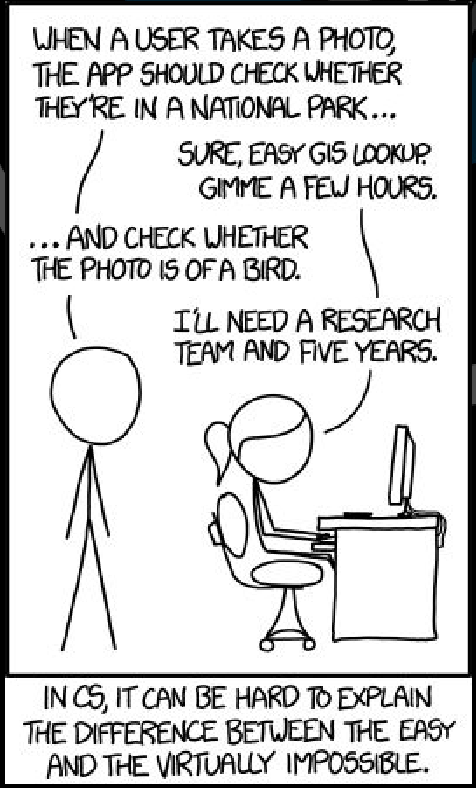
What is a (ro)bot?

Bots are:

- Quick
- Tireless
- Carefull
- Stupid

Can it be don with a bot?

Many things which are obvious to a human are impossible for a bot.

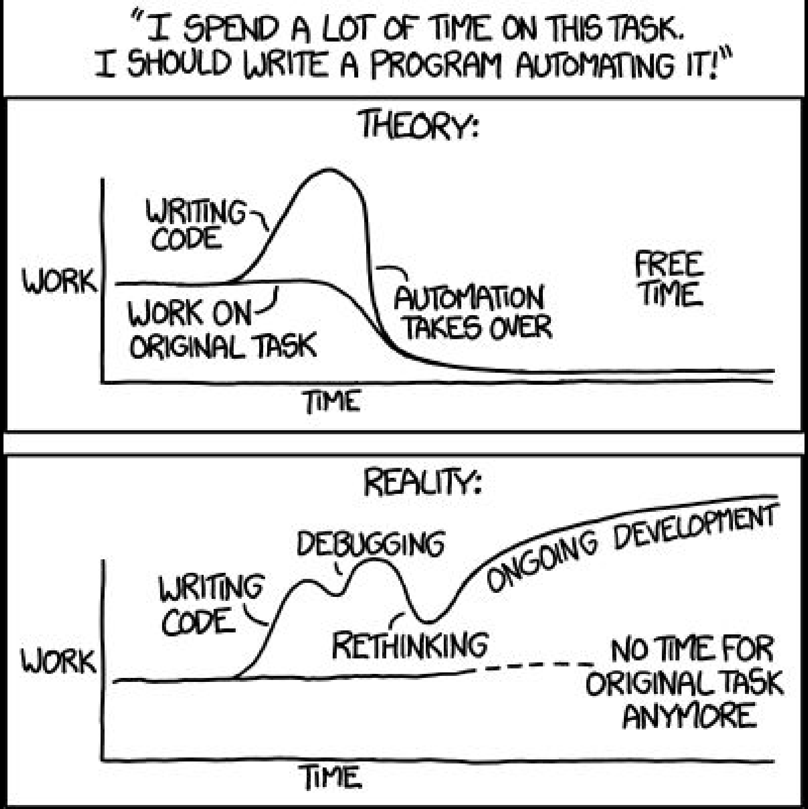

WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
... AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.
IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

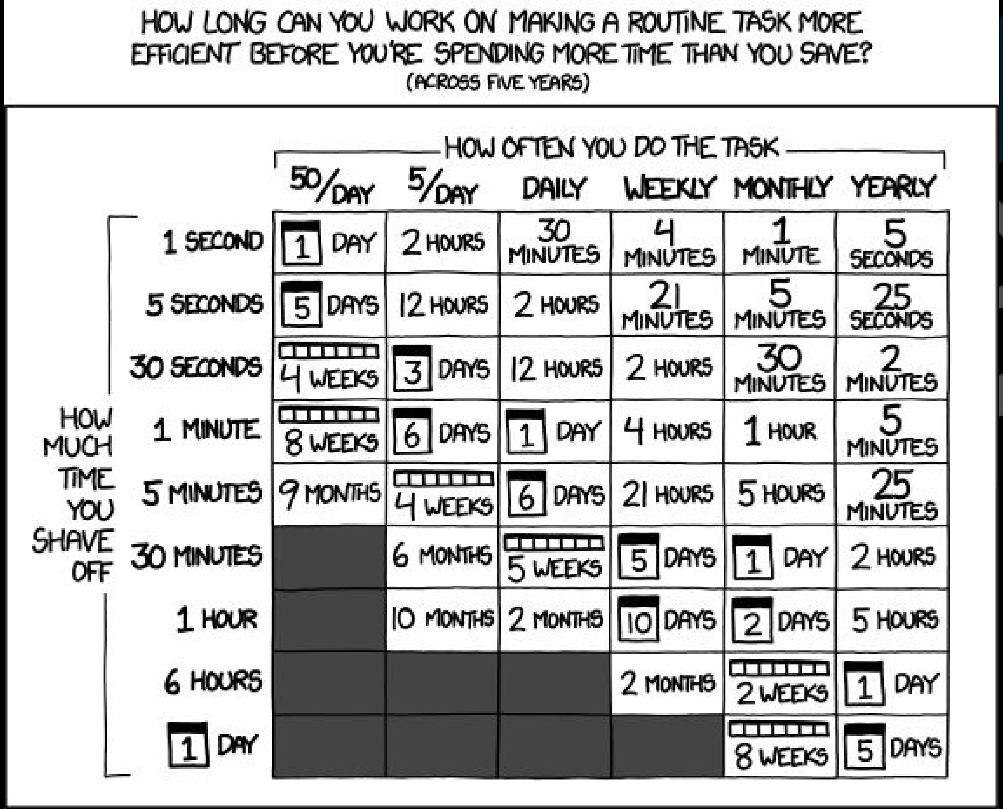
Is it suitable to do with a bot?

Different types of tasks:

- Quicker than doing it manually - repetitive
- Fewer mistakes than doing it manually
- Simpler than doing it manually
- Cannot (reasonably) be done manually

Is it really quicker?





HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

	HOW OFTEN YOU DO THE TASK					
HOW MUCH TIME YOU SHAVE OFF	50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
6 HOURS				2 MONTHS	2 WEEKS	1 DAY
1 DAY					8 WEEKS	5 DAYS

Controlled vs. automatic

- Controlled (one time run)
  - Mark all links as dead after a webpage goes down
- Automatic
  - Welcome all new users after their first edit
- Regular
  - Reset the sandbox every third day
- Semi-manual
  - Identify typos which a user then handles

Bot Etiquette

Rules, permissions and flags

Wiki(p|m)edia specific

- All wikis have different rules and policies
- Some require that each bot project is approved in advance
- Ensure that there is always community support for your activities (on the concerned wiki)
- Overarching: meta:Bot policy and mw:API:Etiquette

Bot account

- Most wikis recommend a separate account
- The user page should make it clear who is responsible for the bot
- “bot” should be part of the name

The need for a bot account depends on type, frequency and number of edits.

Bot flag

In theory:

1) a tag in the logs
2) more powerful queries of the API

I practice:

a trusted user whose edits are seldom audited

Speed

Use built in “limitations”

- Be aware of (local) rules/recommendations
- Start slow

Test, test, test

You are responsible for your bot!

Don’t be bold?

Test so that you don’t have to clean-up!

- Skip edge cases
- Be aware of different scripts and character encodings

Sandboxes and test wikis

Do edit testing in

- Sandboxes
  - Q4115189
  - Wikipedia:Sandbox
- Your userspace (e.g. Special:MyPage/Sandbox)
- Test wikis, e.g.:
  - test.wikimedia.beta.wmflabs.org
  - wikidata.beta.wmflabs.org

The API(s)

The basis for automation

But I wanted bots!

A good indication of what you can/can’t get out of Wiki(p|m)edia

Your basis if you don’t use a framework

A stand alone tool for simpler tasks or tools

Where?

Endpoint:

en.wikipedia.org/w/api.php

commons.wikimedia.org/w/api.php

etc.

- In addition to automatic documentation: mediawiki.org/wiki/API
- Formats: **json** / xml / php and more

An example

Categories of the API and Wikimedia articles

```
action=query
prop=categories
titles=API|Wikimedia
redirects=
cllimit=10
```

/w/api.php?action=query&prop=categories&titles=API|Wikimedia&redirects=&cllimit=10

The Sandbox

At Special:ApiSandbox there is a composing-/test environment for API-queries

Live! so test on your own pages or a test wiki if you are editing

Errors and warnings

Warnings - non-critical / answers possible

- Example

Errors - critical / no answer

- Example
- Error code and info
- HTTP-header: MediaWiki-API-Error

Wikidata (under development)

Most important:

- `action=wbgetentities`
  - Information on specific objects (e.g. Open data)
- `action=wbsearchentities`
  - Objects which give search hits for a given word ina a given language (e.g. ”Öppna” in Swedish)

API Etiquette (short)

- Combine your queries
- Don’t run too quick
- Use the `maxlag` parameter if possible
- Use a descriptive `User-Agent` header
  - Name of the app
  - Contact details
  - meta.wikimedia.org/wiki/User-Agent_policy

Other APIs

WDQ: Query engine for Wikidata

WDQS: SPARQL endpoint for wikidata

Database access: SQL (on labs), try on Quarry

RestBase: For read access and internal use

Dumps: For when you want to crunch all of Wikipedia

Tools & Frameworks

From point’n’click tools to coding

- ApiSandbox:
  - Run simpler processes straight from your browser
- AutoWikiBrowser (Windows):
  - Requires no programming skills
- Pywikibot (formerly Pywikipedia)
  - Powerful framework in python with a library of pre-made scripts
- Purpose specific tools
  - Plenty around, especially for Wikidata
- Frameworks in many other languages

AWB: Quick to learn and use

en.wiki/Wikipedia:AutoWikiBrowser

- Windows only
  - In theory also Linux / Mac
- Simple but can be made very powerful through Regular Expressions and plugins
- Wikipedia-centric
- Whitelisting (e.g. en.wiki)

AWB: Regular expressions (Regex)

en.wiki/Regular expression

- Find text strings with a certain syntax
    - t.ex. *.mp3, wiki(p|m)edia
        - all .mp3, wikipedia or wikimedia
    - \[(http|https):\/\/wikimedia.se ([^\]]*)\]
        - All wiki formated links to wikimedia.se which also have a link text
- Not only for AWB

AWB: demo

AutoWikiBrowser

The semi-automated wiki editor

Available under the GNU General Public License.

Pywikibot (core)

Framework + library of scripts

Also supports Wikidata and Commons

Actively developed

Built in safety valves (edit conflicts, throttling etc.)

Cross platform (because it’s Python)

Pywikibot: demo

Purpose specific tools

- For Wikidata
  - Autolist: example
  - QuickStatements
- Anti vandalism: Huggle, STiki, …
- Url2Commons: image uploading (piped example)
- Page Piles: pipe output from one tool into another

check out the directory for many more tools

Tool Labs

Hosting environment for bots and tools

When do you need Tool Labs?

Access to replicas of Wikimedia databases

Simplifies collaboration

Web server for:

- Recurring (Cron) and self-running jobs
- Tools initialised by users
- Handles load balancing etc.

Caveats

Not a production server, outages occur

Code must be freely licensed and open source

Almost any bot must be run from a tool account (not your user account)

This ensures abandoned tools can be saved

Housekeeping

Copyrights, sources and links

Images / Copyrights

Image sources:

- Wikimedia Sverige logo.svg / Wikimedia Foundation / CC BY-SA 3.0
- BotAcademy.svg / Jan Ainali / CC0 1.0
- XKCD #1425, #1319, #1205 / Randall Munroe / CC BY-NC 2.5
- AutoWikiBrowser2.png / AutoWikiBrowser team / GPL v2
- Pwb icon.svg / Xqt et. al / CC BY-SA 3.0
- CC-BY-SA icon.svg / Creative Commons / CC BY-SA 2.5

Useful links (in no particular order)

- Wikipedia:Creating a bot
- Wikidata:Creating a bot
- Robot creating of articles with AWB, for dummies (Swedish)
- Pywikibot - Python 3 Tutorial (for Wikidata)
- API presentation (2014, English)