Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik

Diplom Thesis Informatics

Mit der Trello-API rummuckeln

Sebastian Engel

24th July 2012

Referees

Name Erstgutachter
(Bioinformatik)
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen

Name Zweitgutachter
(Biologie/Medizin)
Medizinische Fakultät
Universität Tübingen

Engel, Sebastian:
Titel der Arbeit
Diplom Thesis Bioinformatics
Eberhard Karls Universität Tübingen
Thesis period: 13th march to 11th September 2012

Abstract

Trello is a collaboration webservice to manage projects and assign their todo items to co-workers. There are many collaboration tools today, but most of them are very basic. Trello is very extensive and it is optimal fo small businesses. But although it works fine like it's supposed to it has its limits. Trello as its state now is a closed system. Nothing gets in or out unless you use Trello itself. But sometimes it would be handy if you were able to get content from Trello out into other applications. For example a CMS which should contain completed theses which you are already managing in Trello.

So this thesis addresses small scripts which let Trello interact with other webservices and applications. For this purpose I wrote a wrapper of the Trello API in Ruby to accomplish this task in the most dynamic way possible.

Acknowledgements

Write here your acknowledgements.

Contents

List of Figures

List of Tables

Nomenclature

API	Application Programming Interface
CLI	Command Line Interface
CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
JSON	JavaScript Object Notation
REST	Representational State Transfer
URI	Uniform resource identifier
XML	Extensible Markup Language

Chapter 1

Introduction

Blablabla.....

Die Arbeit gliedert sich dazu wie folgt: Die Grundlagen von BlaBlaBla werden in Kapitel 1 erarbeitet. ... Eine Diskussion und ein kurzer Ausblick im Kapitel **??** beschließen diese Arbeit.

Bevor wir uns der Auswertung bzw. Bewertung der gewonnenen Primärdaten zuwenden, wollen wir zunächst einige grundlegende Begriffe der deskriptiven Statistik wiederholen.

Chapter 2

Principles

2.1 Trello

2.1.1 How Trello works

Trello is a webservice by the New York City based web corporation Fog Creek Software[1]. It is a collaboration tool where you can manage your projects. There is the concept of so called *boards* which contents several configurable lists. In these lists you can create todo items you're working on, these are called *cards*. You can add your co-workers to these boards and cards. So everone who's working on a project can see whats going on at the moment.

2.1.2 Why Trello

Trello is not just one of hundreds of thousands of todo applications. It is streamlined for the purposes of small businesses. So for our needs in the university with small groups of people working on the same things it was perfect. Trello has proofed its value several months already.

The first wish was to see the due dates of the cards one is assigned to in Google calendar. Because Google calendar is the calendar tool of our choice. But thinking about that there were many other use cases for small scripts which could run as a cron job on a server to serve several regular tasks.

2.1.3 Trello API

Trello has an API which is still in beta at the moment I'm writing this. But it is already very extensive. [tre12d]

[1] http://www.fogcreek.com

REST

The Trello API is a *RESTful* web API. That means that the API is conform to the REST design model. REST is a common style of software architecture for distributed systems. An implementation of a REST web service follows four basic design principles:

- Use HTTP methods explicitly.
- Be stateless.
- Expose directory structure-like URIs.
- Transfer XML, JSON, or both.

[res12]

Authentification

Though the scripts which are used here need access to private boards in Trello there has to be any kind of authentification. For user applications with a frontend the Trello API provides OAuth2. But because of the concept of OAuth2 the user is required to enter his Trello username and password. [oau12] My scripts are supposed to run on servers as cron jobs. There is no user who could manually enter data. For this kind of applications Trello provides a key/token-system. Every user has a private key. Whith this key the user can generate a token. This token will be send along every request to the Trello API. The token tells Trello which scope the request can see. While generating a token one can specify the scope of the token and when it will expire. The possible expriations of a token are between one day and never. In our case we will use *never*. To generate a token one has to visit a special URL: `https://trello.com/1/authorize?key=SUBSTITUTEWITHYOURPRIVATEKEY &name=My+Application&expiration=never&response_type=token &scope=read,write` In this example the token would never expire and could read and write everything the user can access with the API. Other valid values instead of `never` for expiration would be `1day`, `30days`. `30days` is the default value. [tre12b]

2.2 JSON

All the responses to Trello API calls use JSON. It is a subset of the JavaScript programming language. Despite its relation to JavaScript it's language independent. JSON is da data-interchange format like XML. But JSON is built

on two structures. One is a list of key/value pairs. In most programming languages this is realised as an hash, struct, object or associative array. The other structure is an ordered list of values. This is realised as array, list, vector or sequence in popular programming languages. In JSON itself these structures are called *object* and *array*. Objects start and end with braces. Each key is followed by a colon and the key/value pairs are separated by commas. Arrays start and end with brackets. The values are separated by commas. Both can be arbitrary nested. At every point one of my script saves content at any other place than Trello it's in the JSON format, too. That's because it guarantees easy compatibility with Trello. JSON can be saved in files, too. A JSON file has the suffix `.json`. [jso12]

2.3 Ruby

Ruby is a modern general-purpose object-oriented programming language. Its big difference to most other languages is that it focuses on humans rather than computers.

Yukihiro Matsumoto, the designer of Ruby, said once:

> Ruby is simple in appearance, but is very complex inside, just like our human body.[rub00]

That means, that Ruby is very easy readable and is intuitive for humans even so it can perform complex tasks. This is achieved with English keywords instead of brackets and braces. The result for the programmer of this consistent philosophy is a very easy to read language which is also very plain. Because of the English words instead of abstract characters Ruby is easy understandable. Even non-programmers understand mostly whats going on. So programmers produce way less errors while writing the code. A wrongly spelled word is more intuitive recognisable than a missing bracket or semicolon. [rub12a] More about Ruby can be found at `http://www.ruby-lang.org`.

2.3.1 RubyGems

Ruby has a good amount of methods and classes every Ruby installation provides. But there are hundreds of extensions for special use cases – to communicate with RESTful Web APIs for example – made by third party developers. In Ruby such extensions are called *gems*. To manage and publish these third party libraries theres is the standard *RubyGems*. It provides a standard format for third party libraries for Ruby, a tool to manage the installation of gems and a server for distributing the gems. [rub12b] Some Ruby distributions are delivered with several gems. Gems can be added to an existing Ruby installation at any time.

To install an additional gem on a Unix operating system the following command can be used:

```
gem install gemname
```

Where `gemname` is the name of the respective gem. If the installation performed without errors, the gem is ready to use. [rub12c]

To use an installed gem in a Ruby script the following code at the top of the script before the code starts is necessary:

Listing 2.1: Using the gem *gemname*

```
1 require 'gemname'
```

Again, *gemname* stands for the name of the respective gem. If any gems are used in these script which are not part of the Ruby standard distribution, they are listed at the beginning of the related description. Further information at http://doc.rubygems.org and http://rubyforge.org/projects/rubygems/.

Chapter 3

Applications

3.1 Trello API wrapper

These scripts fulfill very different tasks, but they have also much in common. For example almost every script loads single cards. At least potentially. So I wrote a set of functions and classes which represent Trello for Ruby. This is kind of a translation of Trello to Ruby and vice versa. Additionally now the scripts can use the functions and in consequence they can stay very lightweight and clean. Almost everything that's possible whith the Trello API is possible with this API wrapper, too. But due to the fact that the Trello API is still in beta, there can always be errors and missing features.

3.2 Command Line Interface

Almost every script needs some informations. An information which every scripts need is key and token of the user which account should be used for the access to Trello. The scripts have to know which cards, lists and boards they have to look at. So these information has to be passed to the scripts, too. At first we set this information at the top of the script. But it emerged that it's very unpractical to hard code this in each script. So it would be impossible to use the same Ruby file with several Trello accounts. For every Trello account the user has to generate the file. The solution for this problem is a command line interface (CLI). With a CLI the user can pass information to the script in a predefined format, so the script knows exactly what to do. For every other call the user can specify different information for one and the same script.

The CLI commands for the Trello scripts are as follows:

`-k privatekey` the key of the user

`-t token` the token of the user

`-c card-id[, card-id]` one or more card-ids

`-l list-id[, list-id]` one or more list-ids

`-b board-id[, board-id]` one or more board-ids

`-a` all boards with all their cards this account is able to see

These are the basic CLI commands which every script uses. For some script there are additional commands. They are explained in their respective sections.

3.3 Export to HTML

The export to HTML script is

Used gems:

- `erb`
- `json`
- `open-uri`
- `pp`
- `kramdown`

3.3.1 Markdown

3.3.2 Twitter Bootstrap Framework

3.3.3 HTML 5

3.3.4 CSS 3 / SASS

3.3.5 ERB / Templating

3.4 One way sny to Google Calendar

3.5 Export to iCal

3.6 One way sync to Joomla

3.6.1 For every card an article

3.6.2 All cards in one article

3.6.3 One way sync to WordPress

3.7 Backup

3.7.1 Export

3.7.2 Import

Filename option

The -n (or -name) argument for this script stands for the filename of the backup file which contains the exported Trello data. With -n the user can specify a file to import. While processing the script first checks if the user has passed this argument. If not, it aborts. If the -n argument is given, the scipt proofes if the file is a ZIP file. For that it soesn't use the filename but the MIME type of the file.

TODO: listing design

In line 1 the `file -Ib #{filename}` is a bash call for receiving the MIME type of a file. Ruby executes it and with the gsub-Method it cuts the MIME part out of the received string. This shell script part in a ruby file is a bit dirty. But only for this small case it would be elaborately to use a seperate gem.

Listing 3.1: Checking if the file has the MIME type “application/zip”

```
1 if `file -Ib #{@filename}`.gsub(/;.*\n/, "") != "
    application/zip"
2   puts "ERROR: The backup file has to be a ZIP file!"
3   abort
4 end
```

TODO: What’s a MIME type?

3.7.3 Member import

Chapter 4

Conclusion

Chapter 5

Outlook

5.1 Trello Alfred Extension

Alfred [alf12] is a small Mac application which simplifies the way one can search the web or access all sorts of applications. It constist just of a input field which one cann access with a keystroke combination. It's like an extended Spotlight (on Mac) or Windows Search (on Windows). Developers can write extensions to access other webservices and applications with Alfred. It's even possible to run scripts with Alfred. With that possibility given it's perfect for acessing Trello while working in a fast and easy way.

There are three commands to add or read cards with this extension:

1. `trello board-name` will return the card-names and statuses of this board.

2. `trello board-name list-name` will return card-names and statuses of this list in this board.

3. `trello board-name text for a new card` will add a new card with the specified text to the first list of this board.

4. `trello board-name list-name text for a new card` will add a new card with the specified text to this list of this board.

If you enter `trello Berlin Visit the Reichstag` in Alfred the extension looks for a board called *Berlin*. If it finds nothign it looks for *Berlin Visit* and so on. So your board names shouldn't end with an imperative. The thought behind this operating principle is that it's very unlikely that a board name ends with an imperative and that imperatives are often used for card titles because cards are sort of a command.

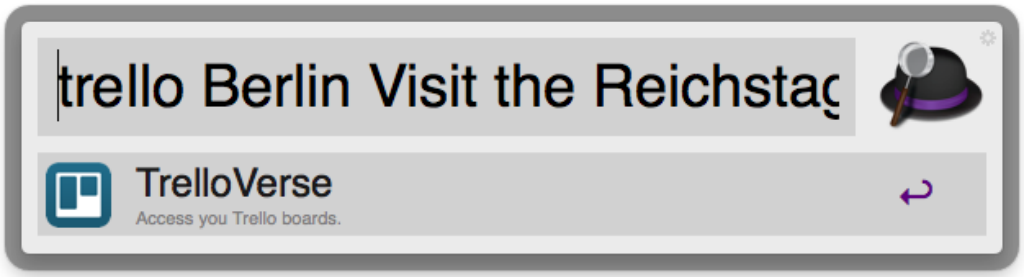

Figure 5.1: Alfred Extension for Trello: This command would add a card with the name *Visit the Reichstag* to the board called *Berlin*.

If you omit the text after the board name the extension will show you all card names of this board and its statuses.

Sometimes there are several boards with similar board names. In this case the extension will pick the "last"match. So if you have two boards called *Berlin* and *Berlin sightseeing* the extension will would pick *Berlin sightseeing*. This approach makes sense because if the extension would pick the first match, in this case *Berlin*, it wouldn't be possible to access *Berlin sightseeing*. In the case that one wants to access *Berlin* and add a new card beginning with *sightseeing* one has to put this board name betweet tick marks.

TODO: Code this and verify the practicability.

5.2 Native applications

Although Trello is an extremely good web-app, I'm of the opinion that a native application is always the better solution. The first reason is because it's a dedicated app and so it's integrated with the operationg system. Especially for todo-applications it's an advantage that they can access the systems notification system, or that they could completely vanish in the background so they don't bother the user while working. There are mobile applications for iOS [tre12a] and Android [tre12c] by Trello itself. But theres no Mac, Windows or Linux application.

A native application would even speed up the Alfred extension because the application could cache the data. So there hasn't to be an actual HTTP request for every command by the Alfred extension. And if a HTTP request necessary the user hasn't to wait because the application will handle the command in the background.

Bibliography

[alf12] Alfred app. http://www.alfredapp.com/, 08 2012.

[jso12] Json. http://www.json.org/, 08 2012.

[oau12] Oauth community site. http://oauth.net/, 08 2012.

[res12] Restful web services: The basics. https://www.ibm.com/developerworks/webservices/library/ws-restful/, 08 2012.

[rub00] [ruby-talk:02773] re: More code browsing questions. http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-talk/2773, 05 2000.

[rub12a] About ruby. http://www.ruby-lang.org/en/about/, 08 2012.

[rub12b] Rubyforge: Rubygems: Projektinfo. http://rubyforge.org/projects/rubygems/, 08 2012.

[rub12c] Rubygems manuals. http://docs.rubygems.org/, 08 2012.

[tre12a] App store - trello. http://itunes.apple.com/us/app/trello/id461504587?mt=8, 08 2012.

[tre12b] Getting started — trello documentation. https://trello.com/docs/gettingstarted/index.html, 08 2012.

[tre12c] Trello android app available for download! — trello blog. http://blog.trello.com/trello-android-app-available-for-download/, 08 2012.

[tre12d] Trello documentation — trello documentation. https://trello.com/docs/index.html, 08 2012.

Index

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Diplomarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum Unterschrift