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
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. [trec]

[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.

[res]

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.

2.2 JSON

All the responses to Trello API calls use JSON. JSON means *Javascript Object Notation*. It's not directly related to JavaScript, but it was first developed for the use with JavaScript. **TODO: verify** JSON an easy markup language like XML. But JSON consists of only two conecpts: arrays and hashes. An array is

a list of values. A hash is a list of key-value pairs. 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. **TODO: JSON Wikipedia**

2.3 Ruby

Ruby is a modern scripting language. It's very similar to Pathon and fills the same purposes as PHP, which is very popular since years. The big difference to older scripting languages like PHP is, that it's much more easier to read. Ruby doesn't require brakets. One can write `do` and `end` instead. Thats much more understandable than { and } like they are used in PHP. But `do` and `end` are replacable by { and }.

2.3.1 Ruby concepts

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. There are two different kinds of extensions: *packeges* and *gems*. Ruby gems are small plugins for Ruby which provide additional methods and classes. Gems can be added to an existing Ruby installation at any time. Some Ruby distributions are delivered with several gems. **TODO: Wie heißen die vorinstallierten packages?** Ruby packages are ...

Chapter 3

Applications

3.1 Trello API wrapper

3.1.1 Markdown

Maybe it's better under "Export to HTML"

3.2 Trello framework

3.3 Export to HTML

3.3.1 Twitter Bootstrap Framework

3.3.2 HTML 5

3.3.3 CSS 3 / SASS

3.3.4 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

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
```

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.

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.

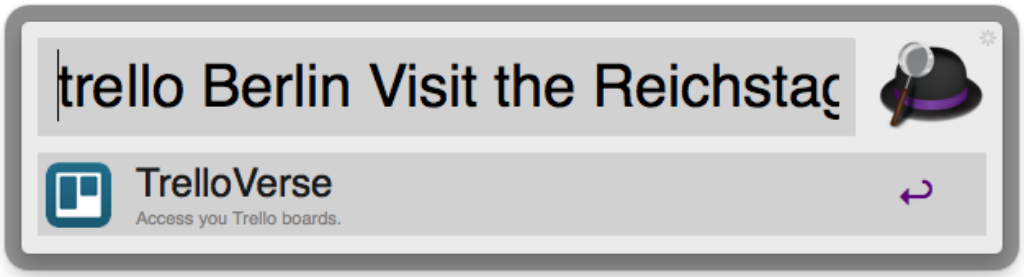

Figure 5.1: Alfred Extension for Trello: This command would add a card with the name *Visit the Reichstag* to the board called *Berlin*.

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 [trea] and Android [treb] 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.

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

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

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

[treb] Trello android app available for download! — trello blog. http://blog.trello.com/trello-android-app-available-for-download/.

[trec] Trello documentation — trello documentation. https://trello.com/docs/index.html.

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