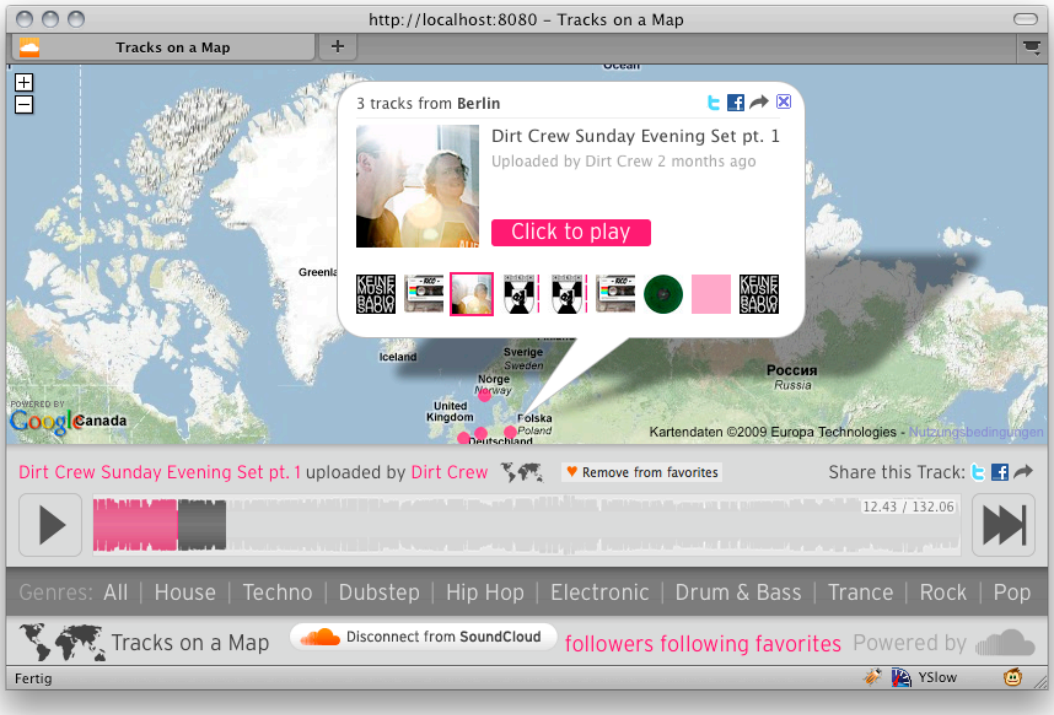
Tracksonamap

Extending a legacy project
with oAuth on Google App Engine



The SoundCloud Connect feature in Tracksonamap and this document have been created by Johan Uhle during the seminar „Social Web Application Engineering“ under the supervision of Christian Willems in the winter term 2009/2010 at the Hasso Plattner Institute in Potsdam / Germany.

A **live version** of Tracksonamap can be found here
http://www.tracksonamap.com

The **source code** has been released under Apache 2 License on Github here
http://github.com/freenerd/SoundCloud-Map/

Hasso Plattner Institute
http://www.hpi-web.de

Johan Uhle
http://www.freenerd.de
http://www.twitter.com/freenerd

Introduction

This paper is going to give an insight into the development process of Tracksonamap. The challenges and solutions on the way to implement the SoundCloud Connect feature using oAuth are described. An extra focus is put on the efforts of working with a legacy project.

About Tracksonamap

Tracksonamap is making the latest tracks uploaded to SoundCloud available on a map. Users can discover new music from all around the world by browsing the map, opening cities and playing tracks. When a track is finished the next one is played automatically. The music can be filtered by provided genres via explicit filter at the bottom and by browsing to locations.

Tracksonamap is based on SoundCloud Map, which has been developed by this papers author in the seminar „Webprogrammierung / Web 2.0 Technologien“ in the summer term 2009 at the HPI. If you want to know more about the foundations of Tracksonamap, please read the paper „*SoundCloud Map: A Web 2.0 Mashup with SoundCloud and Google Maps*“ [1].

After the seminar had finished SoundCloud Map was not ready to be launched. Further steps were made towards finishing it: The backend processes were refactored to wide extent to be more error-proof and work efficiently. Sharing functionality to Twitter and Facebook was implemented and the user interface was completely redesigned.

A major development jump happened at Music Hackday Berlin [2] [3] on September 18th - 20th 2009. The application was finally launched on October 9th 2009 [4] [5]. Since then only minor bug fixes mainly for the backend have been deployed.

The communication about Tracksonamap with the users happens via a dedicated Twitter Account [6]. Google Analytics is used for collecting visitor statistics. Since the launch about 40.000 to 50.000 visitors come to the site every month.

---

[1] http://www.freenerd.de/soundcloudmap-paper/

[2] http://berlin.musichackday.org/index.php?page=Tracks+on+a+Map

[3] http://berlin.musichackday.org/

[4] http://twitter.com/tracksonamap/status/4739363404

[5] http://blog.soundcloud.com/2009/10/09/tracksonamap/

[6] http://www.twitter.com/tracksonamap

SoundCloud Connect

The idea for this semesters course was to implement a SoundCloud Connect [7] functionality. The idea is to have a „Connect with SoundCloud“ button in Tracksonamap. When the user clicks on it, a popup opens where the user can authorize Tracksonamap to connect with the users SoundCloud account. After the authorization Tracksonamap is able to display the users followers, followings and favorite tracks (further referenced as „network information“) on the map. Furthermore the currently playing track can be favorited or unfavorited by the user.

This feature set offers fun for users because it is a new approach to the users network on SoundCloud. Listening to the own network is something that is currently not easily possible on SoundCloud. Mapping the own followers also embodies real value for artists on SoundCloud since it is a geographical representation of the artists fans.

These features have been requested several times by the users of Tracksonamap.

Refactoring

Before the development of the SoundCloud Connect features could start a refactoring of the existing code had to be done. This was mainly because SoundCloud Connect affects all modules of Tracksonamap and uses a lot of functionality already available in the code. During the refactoring modules were split into submodules and existing methods were refactored for being more generic.

Testing

The goal of refactoring is to improve the codes structure without changing its behavior. One of the best practices in refactoring is to have a big test suite with a high code coverage. If all tests pass after the refactoring the new code is likely to behave as intended.

Tracksonamap is not having any tests. Therefore I can not trust the refactored code. Testing the application by hand both locally and deployed in development mode exposed no problems. But the experience I got since the launch taught me, that a lot of errors especially in the back-end processes only surface on the long run. Be it because of Google App Engine specific environment parameters (as depicted later) or the problems occurring because of the work with remote APIs. Particularly these edge cases should be covered in tests.

As long as the refactored code is not tested I am not going to launch the new features. The tests will be written with the help of Nose[8] and executed with the help of the Nose Plugin NoseGAE[9]. The lesson from this for me is that that I would not start a software project again without thinking about and implementing testing at the appropriate point.

---

[7] http://wiki.github.com/soundcloud/sc-connect/

[8] http://somethingaboutorange.com/mrl/projects/nose/

[9] http://pypi.python.org/pypi/NoseGAE/

Implementing SoundCloud Connect

After the refactoring was finished the actual implementation of the SoundCloud Connect features started.

SoundCloud Connect is based on oAuth 1.0a [10] . A python wrapper[11] is provided by SoundCloud under the LGPL License which wraps around the oAuth authorization process and the API requests to make working for the programmer more comfortable.

When the user enters the site, a „Connect to SoundCloud“ button is displayed on the bottom. On click a pop-up by SoundCloud opens where the user has to allow Tracksonamap to be connected to the SoundCloud account. Afterwards the user is returned to Tracksonamap where 3 new buttons are displayed on the bottom. They provide access to displaying the users followings, followers and favorite tracks. Furthermore a new button appears in the audio player enabling the user to see and change the favorite setting for the currently playing track.

The network information is fetched asynchronously in the backend after the oAuth authorization is finished. This may take some time depending on the size of the users network. To avoid exceeding the execution limit of 30 seconds which Google App Engine imposes on scripts this part is implemented with the help of the Task Queue API[12]. Every network item is fetched in its own task. Several tasks are processed in parallel which speeds up execution. Google App Engine guarantees, that every item is processed. This structure leads to a fast and secure fetching of the users network.

The internal API provides access to the fetched data in the JSON[13] format. It is queried by the front end which is written in JavaScript using the jQuery library[14].

The API wrapper provided by SoundCloud had to be customized for running in the restricted Google App Engine Python environment.

The „web debugging proxy application“ Charles[15] was used during development to debug the local and remote APIs.

A surprising fact for me was that the oAuth 1.0a specification is not providing a way for the consumer application to programmatically revoke an access token at the service provider. Currently each time the user connects to SoundCloud a new access token is issued resulting in a lot of unused access token saved at SoundCloud. Whereas Google has implemented an own extension to allow remote access token revoking[16] SoundCloud is currently not supporting this.

---

[10] http://oauth.net/core/1.0a/

[11] http://wiki.github.com/soundcloud/python-api-wrapper

[12] http://code.google.com/appengine/docs/python/taskqueue/overview.html

[13] http://www.json.org/

[14] http://jquery.com/

[15] http://www.charlesproxy.com/

[16] http://code.google.com/intl/de-DE/apis/accounts/docs/AuthSub.html#AuthSubRevokeToken

Outlook

Before the SoundCloud Connect features can be launched some things have to be done:

As mentioned already the refactoring introduced significant changes to the backend which have to be tested thoroughly. This is needed to assure a trouble-free launch of the new features and to provide a solid base for future extensions of Tracksonamap.

A problem that has to be dealt with in the future is the handling of data that has been changed on SoundCloud but is still saved in the old state on Tracksonamap. This is especially important for deleted tracks and users. The most practical way to solve this problem would be to detect the failure when a user tries to access such wrong data, remove it from the datastore and redirect the user to another resource.

The launch of the new features is likely to attract more visitors and increase the servers load and traffic. This might cause problems with the Quotas limiting the use by Google App Engine. Therefore optimizing the runtime performance and reducing the amount of stored data will become important in the near future to allow further scaling.

The SoundCloud API could improve by offering a possibility for programmatically revoking access tokens and in general becoming faster and more reliable.

Experience with Google App Engine

My experience with Google App Engine comes from developing and maintaining Tracksonamap for over six months.

Very pleasant about Google App Engine is the ease of developing and deploying applications. The documentation is well written and the Google App Engine and Python community are building a sustaining ecosystem. The runtime environment and the internet connection of the hosting at Google are very fast. Within the Quotas using Google App Engine is free.

On the other hand some downsides emerged. Only the outdated version 2.5 of Python is available. Furthermore some native Python modules and methods are disabled. This is especially a problem when working with external libraries which have to be customized to work on Google App Engine. Developers also have no access to the configuration of the server. Additionally developers have to program against the some APIs only provided by Google. This makes it hard to transfer the code to other runtime environments.

The admin interface is not providing enough functionality for developers to effectively meter and manage deployed applications. This was especially severe when Tracksonamap exceeded Quotas. I was not able to conduct counter-measures and not even able to find the cause for the increased traffic and cpu time usage.

It is important to keep in mind that Google App Engine is still a beta product. Google is constantly improving it. Google App Engine has become an important part of the Google service infrastructure. It is well imaginable that Google App Engine or similar cloud-based hosting solutions will grow in significance over the coming years.

Conclusion

The oAuth protocol enables web applications to connect with one another and get remote access to the users data. Therewith it enables the programming of new mashups that offer tight integration between services. SoundCloud Connect is an implementation of the oAuth protocol. The provided Python wrappers makes working with oAuth faster and easier.

Tracksonamap is using these tools to integrate tightly with SoundCloud. It has been implemented on the Google App Engine web application platform which is an environment well-suited for social web applications.