

Proposal for GSoC 2022

Email plugin

Project Abstract

Create a plugin to fetch mail via IMAP and convert messages to notes (including attachments). The plugin should be able to filter what messages it downloads, e.g. based on the folder.

Project Overview

Description

Create a plugin to fetch mail via IMAP and convert messages to notes (including attachments). The plugin should be able to filter what messages it donwloads, e.g. based on the folder.

Additional features to consider:

- support more than one account
- convert HTML to Markdown
- delete/move received emails

Expected Outcome: Email plugin (with the features described above) is available to install from the plugin repo.

Difficulty Level: Medium

Skills Required: TypeScript, JavaScript.

Potential Mentor(s): Roman, Laurent

Expected size of project: 350 hours

Prototype

We'll be using Joplin's API, and I'd like to refer to the following two Node/browser-based JS plugins to create an appropriate plugin. 1. Format conversion function 2. Selector function, which can set retention rules, common tags and how to convert elements 3. Support multiple accounts 4. Delete and move email functions 5. If time is enough, I will add more functions, such as HTML custom add conversion rules, or even convert to other formats.

The difficulty with this plugin is converting the HTML and TXT formats of the message, so we can focus on that. There are several ways to write a plug-in that converts HTML to markdown:

Convert HTML to Markdown using Node.js JavaScript.

HTML syntax is converted to markdown display by switching CSS. (can be referred)

Python library that converts HTML online to Markdown. (can be referred)

Java library that converts HTML online to Markdown. (can be referred)

Node.js JavaScript's Implementation

First install Turndown, then use Turndown's simple API to convert HTML to markdown. You can use an interactive Turndown demo to test out its capabilities. Turndown has a number of options that allow you to use filters to preserve elements that you think might be at risk of incorrect conversion.

reference material:

[mixmark-io/turndown](https://github.com/mixmark-io/turndown):  HTML to Markdown converter written in JavaScript (github.com)

[mdnice/sitdown](https://github.com/mdnice/sitdown): turn html to wonderful markdown (github.com)

CSS' Implementation

To make a CSS converter, you first need to introduce CSS display, enabling Markaown display if necessary. [download link](#):

```
<link rel="stylesheet" href="../css/markdown.css">
```

Python' Implementation

There is a Python library that converts HTML online to Markdown, the way it works is download the package, then use cmd, and we can use it to make plug-ins.

pip install pandoc

pandoc -f html -t markdown -o b.md b.html

```
1 import os
2 path = r'your path'
3 all = os.listdir(path)
4 for i in all:
5     if i[-4:] == 'html':
6         i = i[:-5]
7         print(i)
8         os.system('cd {} && pandoc -f html -t markdown -o {}.md {}.html '.format(path,i,i))
9
```

Java' Implementation

```
1 import com.overzealous.remark.Remark;
2 import com.overzealous.remark.Options;
3 // PHP Markdown Extra
4 Remark markdownExtraRemark = new Remark(Options.markdownExtra());
5 // MultiMarkdown
6 Remark multiMarkdownRemark = new Remark(Options.multiMarkdown());
7 // Github Flavored Markdown
8 Remark githubMarkdown = new Remark(Options.github());
9 // Pegdown with all extensions enabled
10 Remark pegdownMarkdown = new Remark(Options.pegdownAllExtensions());
```

Development Process

Community Bonding (May 20 - June 12)

The task at this stage is to get to know the project in depth, and maybe solve some bugs along the way. Do some research on the required technology, talk to community developers and mentors, and change some technical steps or plans.

Coding Phase 1 (June 13 - July 25)

There are 6 weeks in Coding Phase 1 and an evaluation after the phase. The goal of stage 1 is to complete the Format conversion function on Android or ios. It needs to write some key code.

Week 1-3 (June 13 - July 4)

I will write out the Format conversion function.

Week 4-6 (July 5 - July 25)

The next step is to realize the Selector function, which can set retention rules, common tags and how to convert elements, and the Joplin will get an exciting release update. We will pass the first assessment at the same time. Once the goals of this phase have been achieved, we review the code and fix any known bugs in the previous code. And writing code should be documented.

Coding Phase 2 (July 25 - September 04)

There are 6 weeks in Coding Phase 2 and an evaluation after the phase. We will continue to complete some key tasks and more as we see fit.

Week 1-2 (July 25 - Aug 8)

We will finish “Support multiple accounts” and “Delete and move email functions”

Week 3-4 (Aug 8 - Aug 22)

If time is enough, I will add more functions, such as HTML custom add conversion rules, or even convert to other formats.

Week 5-6 (Aug 22 - September 04)

Maybe pick up where we left off the last few weeks. Some problems should be solved in this week. Once the goals of this phase have been achieved, we review the code and fix any known bugs in the previous code. And writing code should be documented. And then our work is basically done, and then the final evaluation.

Buffer Time (7 days)

There are 9 buffer days in case something didn't go as planned in the weeks before. I think some of the work might take less than two weeks, so we'll actually move faster and have more fault tolerance

More about Me

Applicant Info

Name:	Sen Fang (Wirth)
Email:	wirth.fang@foxmail.com
Github:	github.com/FangSen9000
Time Zone:	UTC+08:00 (China)
Location:	Zhengzhou, China
Education:	Henan University, Victoria University, Australia, double major in Computer Science and Technology. sophomore.
Telephone:	+86 18143465655
CSDN blog:	Wirth's blog (Chinese)

Self-introduction

As for me, I am a pathfinder, I love open source, and really enjoy the atmosphere of GSoC. When I worked in a virtual reality lab, I enjoyed applying algorithms to practice. It is worth mentioning that I have experience in bytedance youth training camp, which is just like Microsoft Exploration Program or Google Step Program. As the Contributor guide says, I get to know the project as quickly as possible and create branches to get familiar with the project. I am very happy to participate in the JOPLIN project, I have full confidence in its prospects, and I will continue to contribute to the project even after the GSoC ends.