
Stars
Nova 2020

Turning a static website more dynamic using GitHub Actions

foosel / Turning a static website more dynamic using GitHub Actions

Nova 2020

Gina “foosel” Häußge

Project lead & main developer of OctoPrint

Development

test

Connection

State

State: **Printing**
Resend ratio: **5 / 15.1K (0%)**

File: **3DBenchy_.25.gcode**
Uploaded: **2018-08-08 12:53:39**
User: **test**
Timelapse: **-**
Filament (Tool 0): **1.74m**
Approx. Total Print Time: **00:43:29**

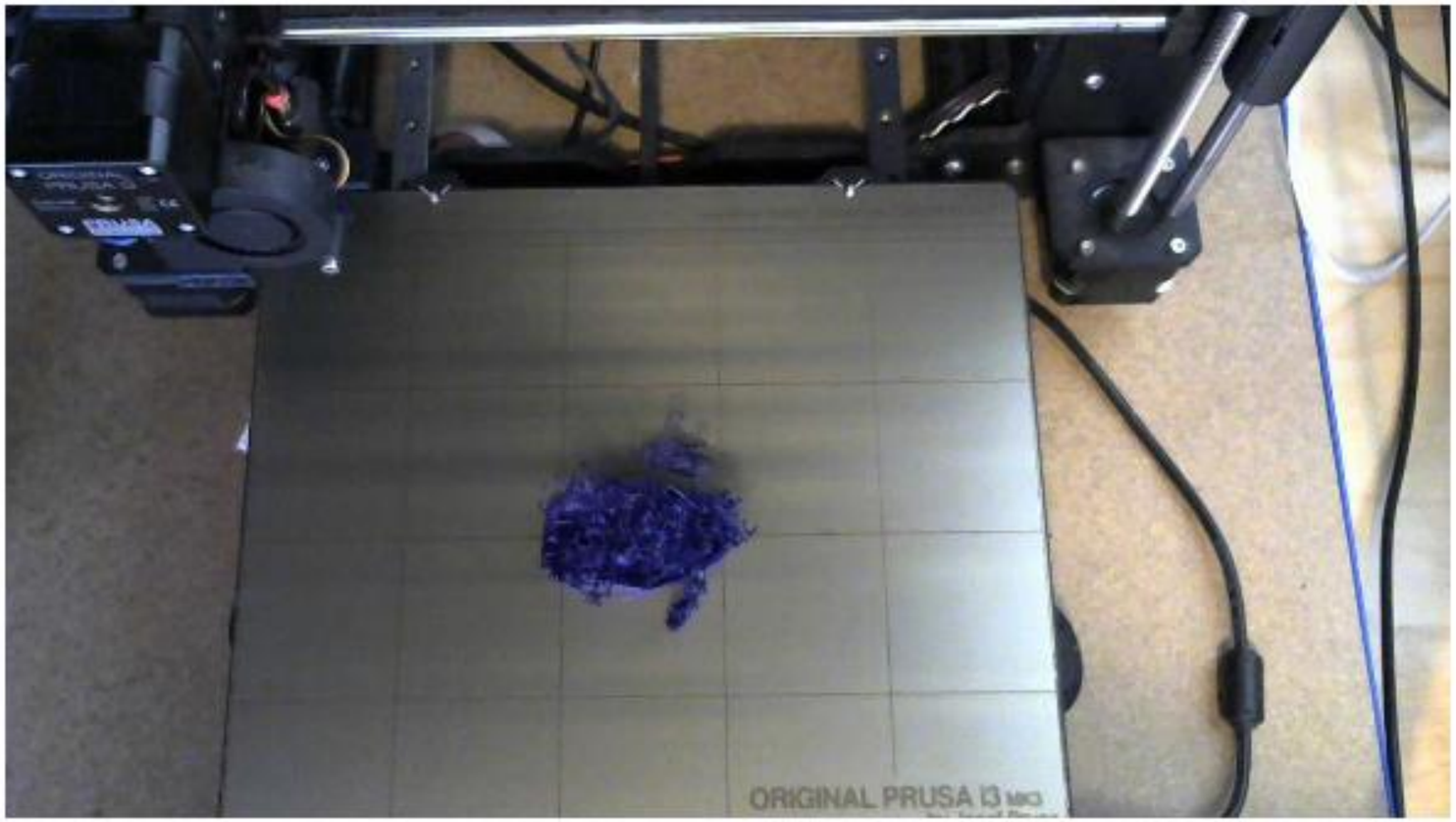
Print Time: **00:04:40**
Print Time Left: **00:25:12**
Printed: **462.7KB / 2.8MB**
15%

Print | Pause | Cancel

Files

Search...

Temperature | Control | GCode Viewer | Terminal | Timelapse



X/Y | Z | Tool (E) | General

5 mm

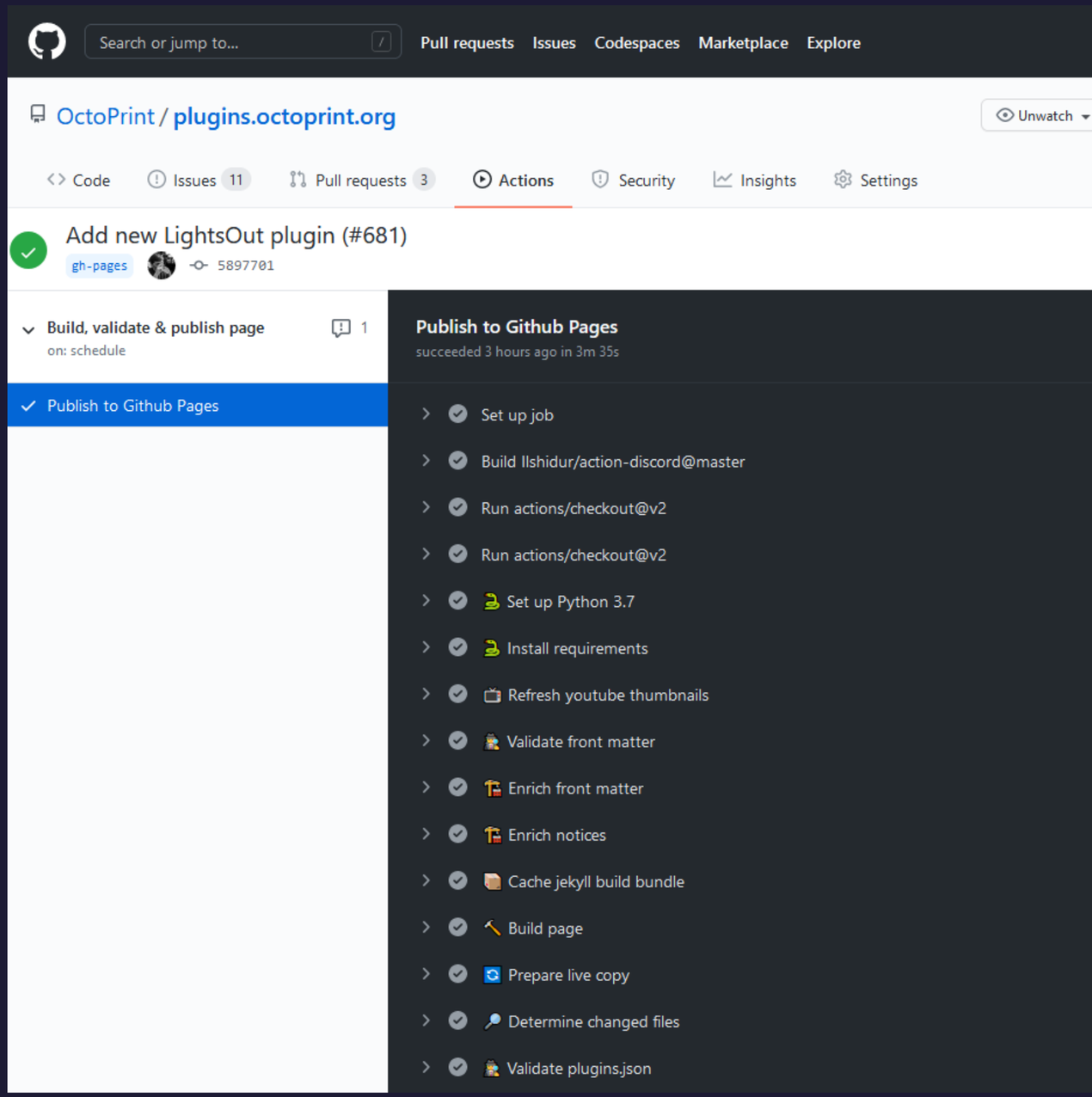
Motors off

Starting point

- Plugin repository for OctoPrint at **plugins.octoprint.org**
- Over 260 plugins by 180 authors, most of them hosted on GitHub
- Static page: Jekyll (w/ collections) on GitHub Pages
- Problem: Which plugins are popular, hot, well maintained, …
- Problem: Outdated compatibility information (Python 2 vs 3)

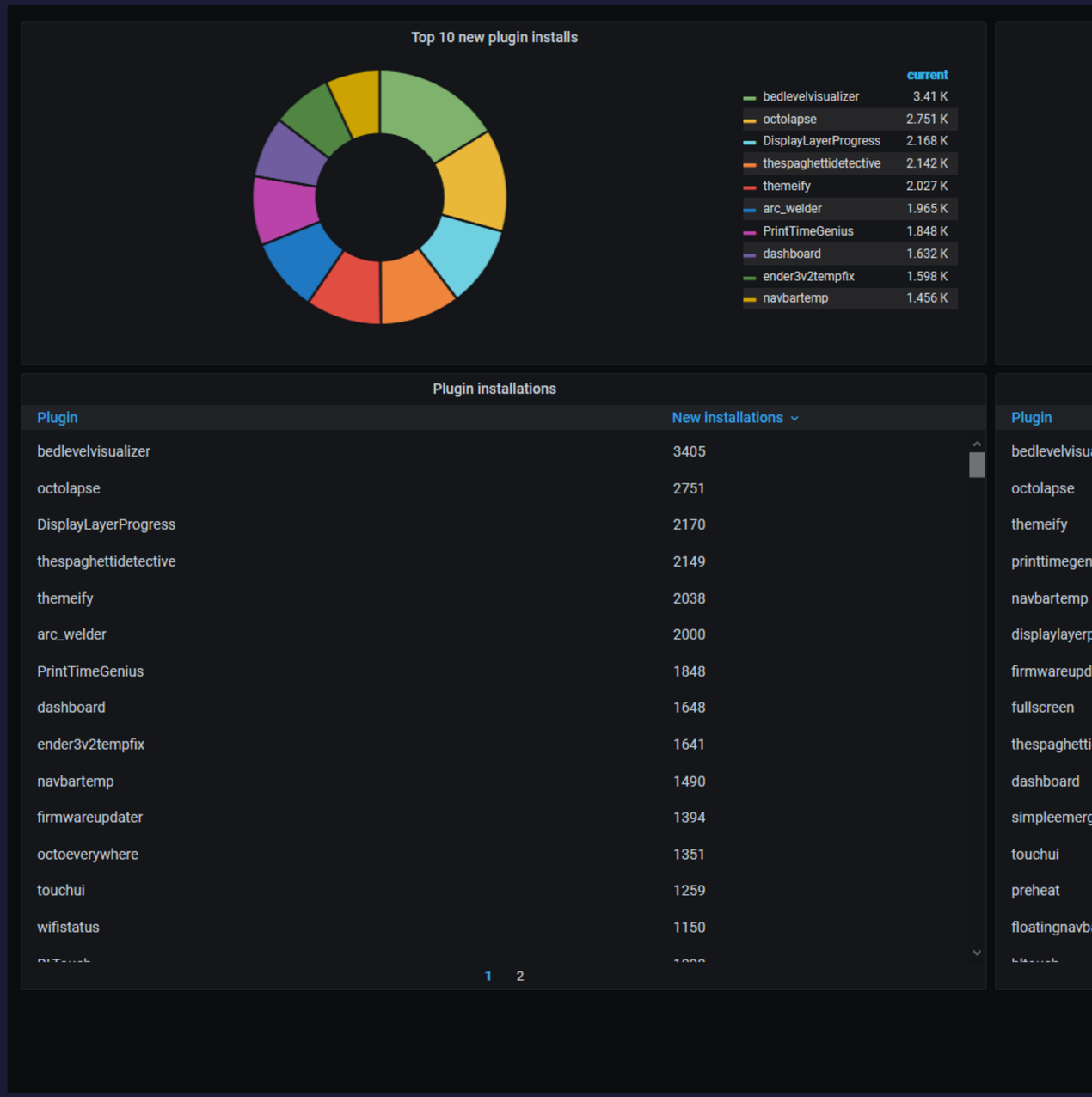
Enter: GH Actions

- Page build workflow
- Caching
- Data enrichment



Usage data

- JSON export from ElasticSearch
- Python build step populates Jekyll collection prior to page build
- frontmatter, requests



GitHub Metadata & raw files

- Detect GitHub hosted plugins (source URL)
- Fetch latest release, number of releases, number of open/closed issues, number of stars, last push
- Fetch main plugin file, parse into AST & extract compatibility information
- Long runtime: Parallelization & not on pull requests

Challenge: API rate limiting

- API calls with personal access token
- Use GraphQL where possible, all in one query
  - Problem: releases (no prerelease filter)
  - Fetch via GraphQL, if prerelease fetch via API
- Refresh page only twice per day

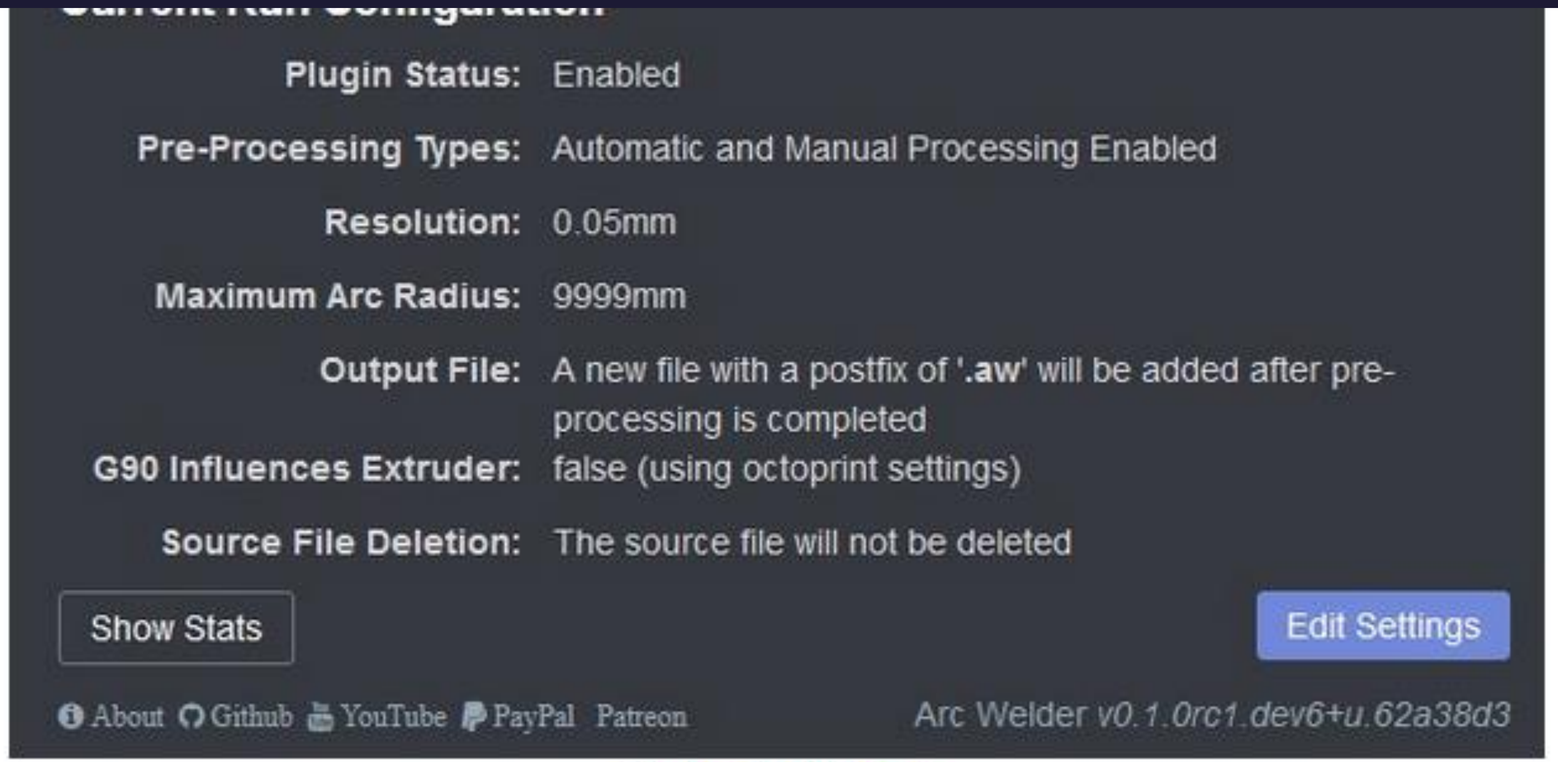
```
PLUGINSTATS_30D_URL = "https://data.octoprint.org/export/plugin_stats_30d.json"

GITHUB_PREFIX = "https://github.com/"
GITHUB_GRAPHQL_URL = "https://api.github.com/graphql"
GITHUB_GRAPHQL_QUERY = """
query {
  repository(owner: \"{{user}}\", name: \"{{repo}}\") {
    openIssues: issues(states: OPEN) {
      totalCount
    },
    closedIssues: issues(states: CLOSED) {
      totalCount
    },
    releasesCount: releases(last: 100){
      totalCount
    },
    lastRelease: releases(last:1){
       nodes {
        name,
        publishedAt,
        url,
        tagName,
        isPrerelease
      }
    },
    lastPush: defaultBranchRef {
      target {
        ... on Commit {
          history(first: 1){
            edges{
              node {
                committedDate
              }
            }
          }
        }
      }
    },
    watchers(last:100){
      totalCount
    },
    stargazers(last:100){
      totalCount
    }
  }
}"""

GITHUB_REST_URL = "https://api.github.com"
```

Star gazing ☆

- OAuth starring proxy
  - Python + flask + flask-dance
  - Docker
- Some JS magic allows starring right from the repository
- Challenge: OAuth Scope needed for starring unclear



The Arc Welder Tab

How To Use Arc Welder

Please read the readme file in the Github Repository for installation and usage instructions. I am planning to add a complete wiki for the plugin in the near future and will link to that here when it is complete.

Please note that if you are using Python 3, you may need to install the *python3-dev* package before *Arc Welder* will install. This is detailed in the prerequisites section of the readme file linked to above.

Registration date
24 Oct 2020

Active instances the past month
Installed on at least **2.04k** instances

New installs the past week
At least **489** new installations

GitHub stats (Logout)

- Latest Release: 1.0.0
  released on 24 Oct 2020
- Releases: 7
- Stars: 142
- Issues: 22 open, 49 closed
- Last push: 25 Oct 2020

Tags

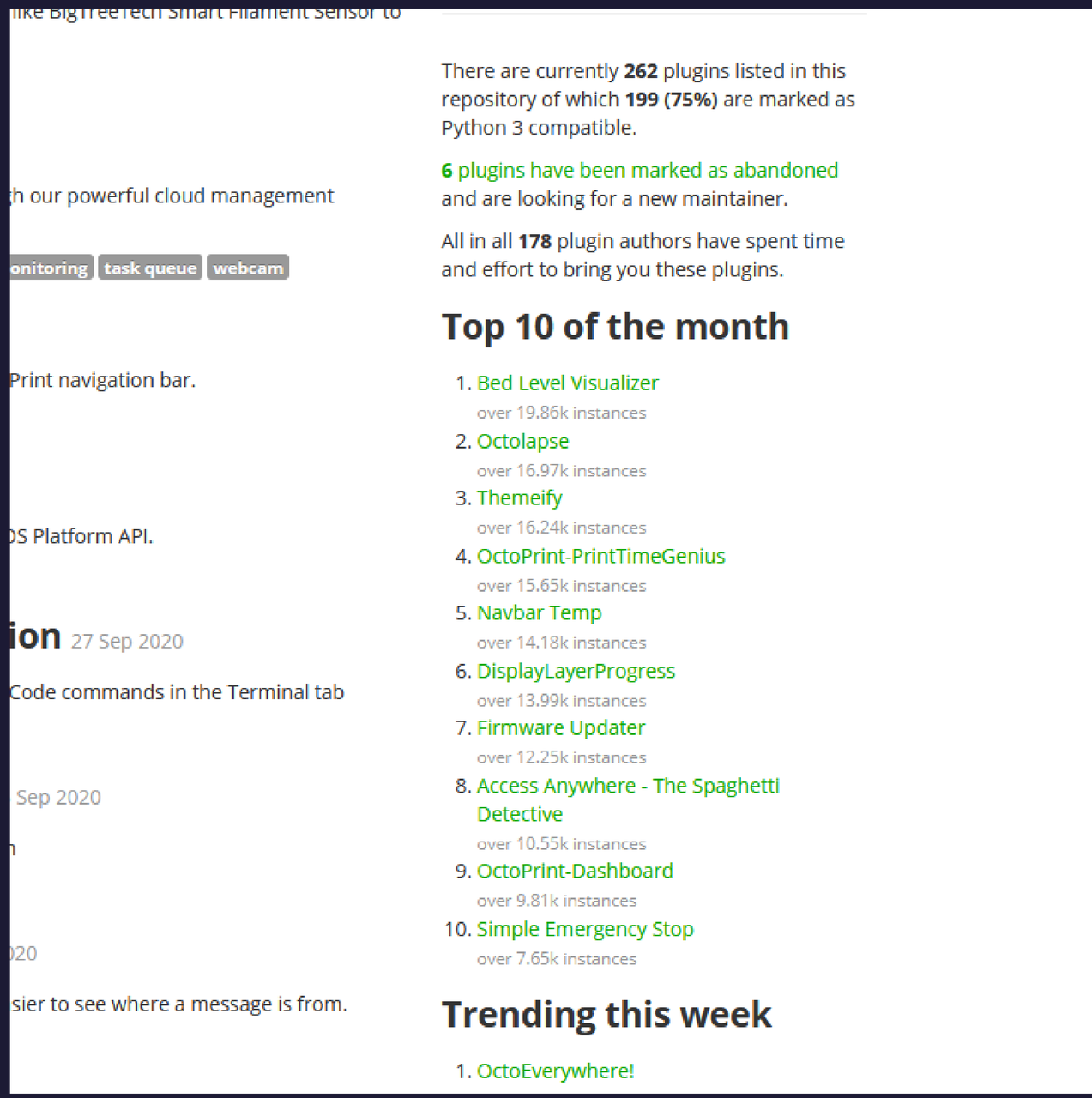
arc compression g2 g3 gcode preprocessing stutter

Compatibility Information

- OctoPrint: 1.4.0+
- Operating Systems: Linux, Windows, FreeBSD
- Python: >=2.7,<4

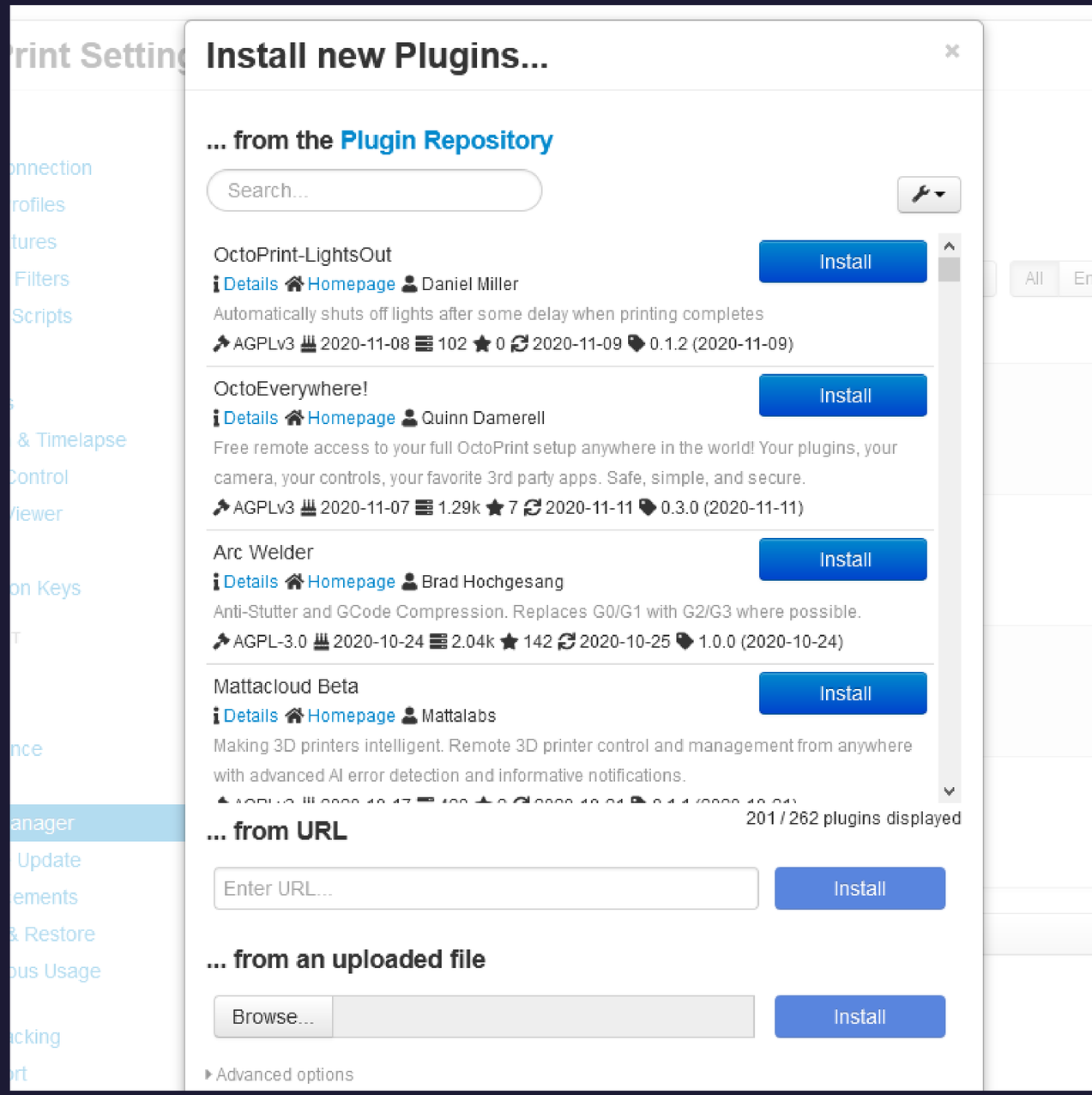
Repository…

- Users can easily find plugins that are popular, trending, …
- Users get an indicator of the development activity on a plugin
- Users get more accurate compatibility information
- Users can show appreciation by starring plugins



… & OctoPrint

- Added to export to OctoPrint
- Availability in built-in repository browser
- Sorting, filtering, …
- No starring (yet?)



Code?

Workflow, enrichment & validation scripts, page

Page source

OctoPrint/plugins.octoprint.org

Star gazing

Star proxy

OctoPrint/github-star-proxy

Usage data JSON

Data exports

data.octoprint.org/export/

Live in action

Plugin repo

plugins.octoprint.org