Jira Querying Script Quickstart

A quick overview of the basic functions provided in the qjira tool.

The tool can run under both Python 2.7+ and Python 3 runtimes. The module dependencies are handled with the standard setup.py file. And both pip and easy\_install installers should work without any problem.

There are several commands implemented and you can see them all in the README file or at https://github.com/andrew-hamlin-sp/jira\_reporting\_scripts.

The two used every week for status reporting are the tech debt command and the summary command.

# Tech Debt Command

This produces a simple table calculating percentage of bug points to story points across a set of projects.

Note: the command name in the tools is actually “debt” not “techdebt”.

Producing a table, such as

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name | Bug Points | Story Points | Tech Debt % |
| IIQ - Chambers Bay | 69 | 69 | 50% |
| IIQ - Magnolia | 87 | 11 | 89% |
| IIQ - Pebble Beach | 48 | 58 | 45% |
| IIQ - Sawgrass | 76 | 29 | 72% |
| Grand Total | 280 | 167 | 63% |

A basic example including all current 7.3 projects is as follows.

$ qjira debt -o debt\_73.csv -f 7.3 IIQCB IIQMAG IIQSAW IIQPB

The output file debt\_73.csv can be imported in Excel and formatted as a table to be copied into the status report.

# Summary Command

This produces an HTML fragment document, by default, that can be copied into Harbor using the HTML formatting option.



Note: There is a –csv option that will produce the original Excel formatted CSV file.

A basic example of summarizing Chambers Bay status is as follows.

$ qjira summary -o summary\_73\_cb.txt -f 7.3 IIQCB

Simply replace the IIQCB with your projects key. I suggest using the txt extension to avoid opening the file as an actual HTML page in a browser or rich text editor. That makes copying the HTML source to Harbor more difficult.

The command will accept multiple project keys but the results will be a single table of interleaved stories from all projects.