|  |  |
| --- | --- |
| Title: |  |
| Attendees |  |
| Absent |  |

This is how the GitHubPageGenerator works:

The documentation of a GitHubProject is made in a word document.The GitHubPageGenerator will parse this manual and generate web pages wiki pages for GitHub.

The advantages:

* All your project documentation will be consistent (manual, web site and wiki)
* All your project documentation can be spelling and grammar checked when created (default M$ Word functionality)

Before you can do this you must first

* Create a GitHub account via <https://github.com/join>
* Create a GitHub project via <https://github.com/new>
* Clone the GitHub project repository in eclipse (i.e. https://github.com/ntenhoeve/Introspect-Framework.git)
* Clone the Github wiki repository in eclipse (i.e. https://github.com/ntenhoeve/Introspect-Framework.wiki.git)
* Create a GitHub website repository called (USERNAME.github.io) via <https://github.com/new> (see <https://pages.github.com/>)
* Clone the GitHub USERNAME.github.io project repository in eclipse (i.e. https://github.com/ntenhoeve/ntenhoeve.github.io.git)

Then create a M$ Word document that documents your project. You can use:

* Normal text Code text (use “code” style which you need to create your self)
* Chapter titles (use heading styles up to 3 levels deep)
* Lists (use bullets. Only one level deep)
* Hyperlinks (external or references to chapters within the word document)

Good examples of chapters that you put in your M$ Word document are:

Every time you update the M$ Word document start the GitHubPageGenerator. The GitHubPageGenerator will parse the M$ Word document and generate web pages wiki pages for GitHub into the local repositories.

Than open and refresh the repositories in eclipse and commit and push them to GitHub.

You can now review the pages:

* Web page: http://USERNAME.github.io
* Wiki: https://github.com/USERNAME/REPOSITORY/wiki

Done.