



Changes to the Website and Documentation

Stuart Pullinger
ICAT Face to Face Meeting
RAL, November 2017

Motivation 1

- To reorganize
 - Difficult for a newcomer to navigate and understand
 - Some gaps
 - ICAT concepts
 - Getting Started/Tutorial
 - Post-installation/using in production
 - Split across different documents, eg.
 - Page on icatproject.org website
 - Page on Maven repository
 - Generic ICAT component instructions



Motivation 2

- To move to different hosting system
 - Version control
 - Per-release documentation versioning
 - Ease of maintenance
 - By existing developers and contributors
 - Using existing tools and services
 - Ease of collaboration
 - Accepting contributions



How?

- Markdown in a Github repository
 - i.e. Text files under version control
 - Tools to convert to other formats eg. HTML, PDF
 - Easy to read
 - Easy to edit
 - Easy to understand/learn for contributors
 - Potential for packaging with software release



Tools & Services 1

- Website – github.io
 - Website written in Markdown
 - Stored in a Github repository
 - Already using GitHub
 - Versioning
 - Collaboration
 - Templates
 - Unfortunately, not very exciting
 - Potential for customisation at the expense of simplicity



Tools & Services 2

- Documentation – gitbooks.com
 - Documentation written in Markdown
 - Stored in a Github repository
 - Already using GitHub
 - Versioning – including tagged releases
 - Collaboration
 - WYSIWYG editors for non-techies
 - Online and offline



Website

- Templates
 - Limited choice
- Desktop preview possible before committing
- Possibility to customize
 - Could introduce build/compilation step
 - Possibility to use Travis-CI for this
 - Would lose ease of editing markdown directly
- Keeping It Simple – want to direct visitors to the correct part of the documentation
- [Demo](#)



Documentation

- Skeleton structure established
- Much to be filled in
 - Mostly by combining existing content
- Some new content:
 - Installation tutorial
 - ICAT Concepts
 - Post-installation instructions
 - Developer Howtos
- [Demo](#)



Documentation Sprint

- Rapidly combine existing documentation
 - From icatproject.org
 - From repo.icatproject.org
 - From previous face to face presentations
- Fill in gaps
 - Getting Started
 - Installation Overview
 - Etc.
- Practice the real-world documentation workflow



Continuous Integration

- Configure CI for all icatproject repos
- Using Travis-CI
 - Free for open source projects
 - Runs test suite on each commit
 - Easy configuration (YAML)
 - Runs in Ubuntu
 - May be ways of testing different OSes using Docker
- Provide reassurance
- Provide reference installation
 - Using



Ansible

- Application deployment and configuration
- Written and extended in Python
- Configured in YAML
- Keeps scripting to a minimum
 - No DSLs (like Quattor, Chef, Puppet)
- Create reference implementation
 - For installation
 - For testing – in Travis CI and elsewhere
- Currently being worked on by Jack Haydock at STFC



Releases Naming/Numbering?

- Problem: Which components are known to work together
 - Tested together
 - On which platform?
 - OS
 - Java + Application Server
- Solution?
 - We document ‘blessed’ releases which are known to work together
 - 2017.1 (icat.server-4.8.0 + others)
 - 2017.2 (icat.server-4.9.1 + others)
 - With tagged documentation release



Do we need...

- Collaboration Policy
 - For example, attribution/transfer of copyright
- Code of Conduct
- Licence for the documentation
 - Creative Commons
 - Attribution – CC BY – seems most similar to Apache2
 - “This license lets others distribute, remix, tweak, and build upon your work, even commercially, as long as they credit you for the original creation.”
- What else?



Discussion

- What are your views on
 - The Website
 - The Manual
 - The test platform
 - Release Naming/Numbering
 - Collaboration Policy
 - Code of Conduct
 - Documentation license CC BY

