NOTE: This is a **public** document that integrates some prior work at WireMock Inc.Contributions are welcome

WireMock Plugin Site - Evaluation Notes

By Oleg Nenashev, Oct-Nov 2023

This document contains evaluation notes for https://github.com/wiremock/wiremock.org/issues/24 - a new Documentation engine for the WireMock website. TL;DR: I recommend going with MkDocs + Multi-Repo Plugin + Material Theme

Requirements

- Documentation as Code
- Ability to have private forks and staging, e.g. for security releases
- Support for Markdown - the most of WireMock ecosystem is built around it
- Nice2Have: Asciidoc support. Asciidoc is a professional documentation markup language and stack which is MUCH more powerful than GitHub Flavoured Markdown
    - Note: Doc engines like Jekyll/Hugo/MkDocs/etc overcome many limitations
- Support for building the documentation site from multiple sources. We want to keep documentation for components and extensions along with their source codes, and build docs from there
    - Nice2Have: Simplified cross-referencing
- Support for including code snippets and document sections from external files
- Support for Multi-tab code samples and flexible syntax highlighting

Considered tools

I started from Antora vs. Docusaurus, and later added MkDocs after spending some time with it for Python WireMock. Hugo and Jekyll were ignored, because it is much weaker than MkDocs and Docusaurus.

NOTE: This is a **public** document that integrates some prior work at WireMock Inc.Contributions are welcome

Summary / Recommendation

- I recommend going with MkDocs + Multi-Repo Plugin + Material Theme. It is compliant with all the requirements, but does not natively support AsciiDoc which would be a strong plus
- Key reasons:
    - Easy to implement, PoC is ready: https://github.com/wiremock/components-site, https://wiremock.github.io/components-site/
    - Minimal effort to migrate the Markdown docs, flexibility of the structure
    - Low maintenance complexity, just a GitHub action and a set of simple config file
    - MkDocs is built around Python, and it is more aligned with the current WireMock roadmap. We want to be more present in the Python ecosystem, and it could be one of the ways to grow visibility and build some collaboration with Python communities
- Cons:
    - Reliance on not so popular extensions like Multi-Repo. We might have to contribute back the cross-rep[o link resolution, and it might become an obstacle
    - When using Multi-repo plugins, GitHub links to the included external repos do not get replaced to in-site relative links. This is something we will need to implement
    - No Asciidoctor support, no stable extension for that. Asciidoctor is much more capable than
        - NOTE: We can still use Some Asciidoc pages by rendering them and including them into the site. Material theme for Antora allows doing it transparently to users if styling is done properly
- If we go with the approach, the implementation would be:
    - Keep the landing in Jekyll, there is no so much content
    - Move the documentation to the new engine
    - Move some Jekyll pages like events to documentation, just to keep it simple
    - Still build as a single site by using relative links, similarly to how WireMock 2.x archive docs run now

Option 1: Antora

- Oleg's wishlist but might be complex to maintain
- Not all developers are ready to AsciiDoc, we want to keep Markdown as a first class citizen at least. It is possible with Antora

Notes

- https://github.com/oleg-nenashev/wiremock-on-antora-poc
- As I discovered, we do not have build-time support for MkDocs at the moment. It was a custom patch at one of the companies I worked for
- I tried to re-create an MVP patch for Markdown import support, but it was rejected by the Antora maintainer Dan Allen. They want to see a much more complicated extension that

would not require patches to the Antora core. It is justified, but I hit timebox when trying it::

- Discussion: https://antora.zulipchat.com/#narrow/stream/282400-users/topic/Antora.20site.20with.20Markdown-ish.20sources
- Patch: https://gitlab.com/antora/antora/-/merge_requests/1000 , extension: https://github.com/oleg-nenashev/antora-markdown-extension
- The more complex implementation is a lot of work, and the maintainer experience still won't be perfect. Also, a lot of issues are likely to happen with metadata, including contributing backlinks

Cons:

- No native Markdown support, one has to use the https://www.npmjs.com/package/@antora/collector-extension to convert files during the build
- "antora.yml" must be present in all imported sources. There is no way around it (proof?). The collector will need
- Fixed an opinionated repository structure (docs/modules/…). We will not be able to migrate docs while preserving the existing GitHub links

Summary:

- Antora is much more powerful than MkDocs, though multirepo extension and the Material theme close the gap considerably
- Moving to Antora requires metadata files to be added in all repositories we include into the documentation (or generated by the converter). Not great for markdown-imported repos where we would rather keep data inside the generator repo
- Markdown import can be considered only as a temporary solution, and this is not what we want given Markdown being a default for open source projects within WireMock ecosystem. We cannot enforce migration of whole ecosystem to Asciidoc, and unlikely want to put much core team effort in that.
-

Option 2: MkDocs Engine (RECOMMENDED)

Idea:

- Using a powerful theme like Material for MkDocs with a lot of features embedded (code browsing and copy, flexible footers, nav)
- Using https://github.com/jdoiro3/mkdocs-multirepo-plugin to source docs

Cons:

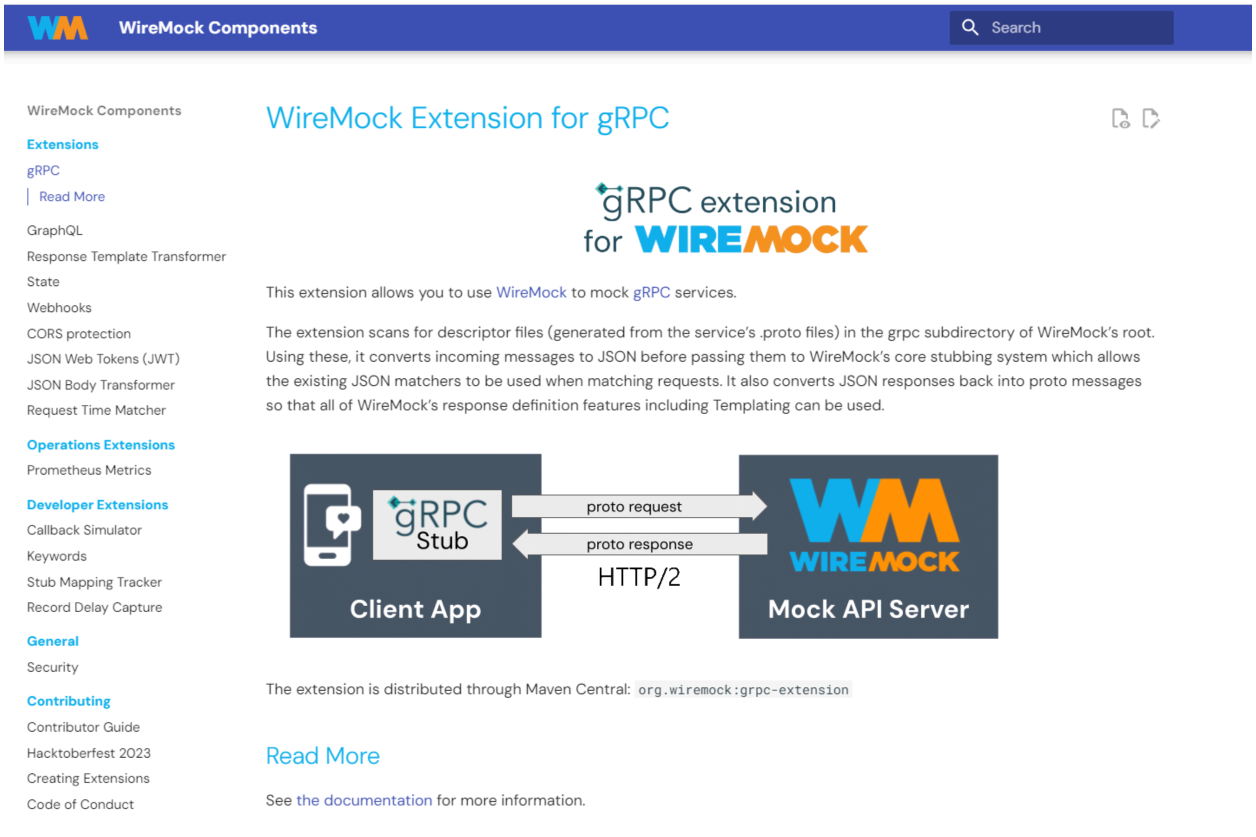
- No Asciidoc support
- The idea is to use Vanilla Markdown where possible. Extensions can be done via themes and the options there are limited. No Macro scripting language embedded

NOTE: This is a **public** document that integrates some prior work at WireMock Inc.Contributions are welcome

- Everything is in Python under the hood, so not a core tech stack for the WireMock community

Prototype

https://github.com/wiremock/components-site/



WireMock Components

Search

WireMock Components

Extensions

gRPC

Read More

GraphQL

Response Template Transformer

State

Webhooks

CORS protection

JSON Web Tokens (JWT)

JSON Body Transformer

Request Time Matcher

Operations Extensions

Prometheus Metrics

Developer Extensions

Callback Simulator

Keywords

Stub Mapping Tracker

Record Delay Capture

General

Security

Contributing

Contributor Guide

Hacktoberfest 2023

Creating Extensions

Code of Conduct

WireMock Extension for gRPC

gRPC extension for WIREMOCK

This extension allows you to use WireMock to mock gRPC services.

The extension scans for descriptor files (generated from the service's .proto files) in the grpc subdirectory of WireMock's root. Using these, it converts incoming messages to JSON before passing them to WireMock's core stubbing system which allows the existing JSON matchers to be used when matching requests. It also converts JSON responses back into proto messages so that all of WireMock's response definition features including Templating can be used.

The extension is distributed through Maven Central: `org.wiremock:grpc-extension`

Read More

See the documentation for more information.

Evaluation notes

- No officially supported scripting language. One would have to use JavaScript for everything custom, and expose YAML data to client side scripting
    - It is quite okay actually, JavaScript with proper styling and framework is no worse than HAML
    - Python extensions are also possible
- There is an extension for including external markdowns, not tested. It might work across the repositories from how the multi-repo plugin works
- Some features require sponsoring the project, but the cost is symbolic and well justified
- Mkdocs-multirepo-plugin does not resolve cross-links between the repositories (changing GitHub references to relative). It can be done via post-processing extensions and maybe contributed upstream
- Multi-version support will work, as long as all links in the documentation are relative. We can import multiple versions of the same site via branches/tags

NOTE: This is a **public** document that integrates some prior work at WireMock Inc.Contributions are welcome

- Mkdocs.yml is likely to grow very complex with Multi-repo. We might want to implement a better format to simplify maintenance. Not for MVP

References

- https://github.com/testthedocs/awesome-docs
- https://squidfunk.github.io/mkdocs-material/alternatives/