Coding in the Open Reviews

Use this checklist as a guide of what to look for when migrating your code to GitHub.com and doing open-source assessments.

> **Note**
>
> This also relies significantly on your common sense as we probably haven't thought of everything, and add anything you think of here

Why are we doing this again?

- To show the tax-paying public what we do with all their money
- To show other software developers the amazing stuff we do at HMRC
- To enhance our career prospects by showing in public the high-quality code we produce here every day
- To possibly end up with something that looks like this: http://netflix.github.io

Our strategy and principles for writing code in the open is well defined: See http://hmrc.github.io/requirements-for-commiters/

Checklist

Security	
	The repository should not reveal any information that could aid an attacker in getting access to sensitive information or performing some kind of attack. None of the following should be included in the repo or any commit history: • References to any internal URLs such as jenkins, qa orchestrator, Kibana etc. • DNS names as they appear in puppet • IP addresses • Secrets used in prod (e.g. encryption keys, certificates, etc.) • Credentials (e.g. usernames, passwords, API keys, etc.) • URLs to JIRA tickets (ticket IDs are OK) • Names of datacenter providers • References to, or copies of, internal service configuration
	There should be no known security implementation vulnerabilities that might pose a non-trivial risk to HMRC. This can be confirmed through security testing (both manual and automated).
	There should be no known security design flaws that might pose a non-trivial risk to HMRC. This can be confirmed through both security design review and risk assessments.
	If the project contains any security-enforcing behaviour (e.g. risk rules, authentication / authorisation behaviour, fraud detection rules), this functionality must either be moved into a separate private repo, or the project may not be open sourced. For avoidance of doubt, calls to an authentication / authorisation service is fine to be open sourced, but the auth service itself may not be open sourced.
Quality	
	The repository must compile and tests run outside of our HMRC environment.
	There must be no user data that could possibly belong to a real person. This includes tax identifiers (NINOs, UTR etc) - we have a library to generate these (https://github.com/hmrc/domain). Names, postcodes, phone numbers, company names and address should be as test-like as possible. EG "ABC XYZ" as name, "1 ABCDE Street, FGHIJ Town, AA1, 1AA" as address. Phone number could be 07000111222. This is a directive form our security consultant. See Test Data in the Open for more details.

	Tests must not start or use a remote service which should not have been open-sourced.
	The code quality is at least reasonable. Remember this will be publicly available and will reflect on HMRC's reputation.
Design	
	The repository should perform one thing well. Where possible (e.g. for libraries), aim for the code to be re-usable in non-HRMC applications (see play-breadcrumb for a good example).
	Libraries must not read configuration from config files. See this post for more details.
General	
	Is the repository's name descriptive? Use Microservice Naming Guidelines as a reference when choosing a name, and ensure these guidelines are followed before submitting for review.
	All dependencies must already be open sourced and on GitHub.com
	Licence headers should be added to files. See the first few steps here (the sbt-auto-build plugin sets things up so that headers are automatically added on compile).
	The project should contain a descriptive README file explaining what the project is and how it can be used.