



How to self-assess your codebase towards the Standard for Public Code

Standard for Public Code

- Developed to make it easier to collaborate on a codebase
- Originally developed by Foundation for Public Code
- <https://standard-compliant.publiccode.net>
- Now community maintained

Simple self-assessment

Standard for a Public Code self-assessment

Check if your codebase is ready for a full assessment.

Standard for Public Code

The [Standard for Public Code](#) is a set of criteria that supports public organizations in developing and maintaining software and policy together. Anyone developing software or policy for a public purpose can use this standard to work towards higher quality public services that are more cost effective, with less risk and more control. This self-assessment will let you know how ready your codebase are for a full assessment.

Start now

Content for this site is generated by [publiccodenet/assessment-eligibility](#) which builds upon [Choices](#).
Contact: info@publiccode.net

<https://publiccodenet.github.io/assessment-eligibility/>

Full assessment

Open-Zaak and the Standard for Public Code version 0.7.1

Link to commitment to meet the Standard for Public Code: [CONTRIBUTING](#)

Code in the open

criterion met.

Meets	Requirement	Notes and links
yes	All source code for any policy in use (unless used for fraud detection) MUST be published and publicly accessible.	code , VNG/GEMMA2 policy linked in README
yes	All source code for any software in use (unless used for fraud detection) MUST be published and publicly accessible.	code
yes	The codebase MUST NOT contain sensitive information regarding users, their organization or third parties.	2020-05-12 review by @EricHerman ; ISO 27001
yes	Any source code not currently in use (such as new versions, proposals or older versions) SHOULD be published. Documenting which source code or policy underpins any specific interaction the general public may have with an organization is OPTIONAL.	releases , docker hub tags

Bundle policy and source code

criterion met.

Meets	Requirement	Notes and links
yes	The codebase MUST include the policy that the source code is based on. If a policy is based on source code, that source code MUST be included in the codebase, unless used for fraud detection.	API specs embedded in the codebase , and linked from documentation "yes" or "not applicable", do we consider it based on the API?
yes	Policy SHOULD be provided in machine readable and unambiguous formats.	OpenAPI is in machine readable yaml format
yes	Continuous integration tests SHOULD validate that the source code and the policy are executed coherently.	GitHub workflow , API Test Platform

Make the codebase reusable and portable

Full self-assessment

- Testing policy and documentation for consistency with the source and vice versa is OPTIONAL.
- Testing policy and documentation for style and broken links is OPTIONAL.
- Testing the software by using examples in the documentation is OPTIONAL.
- Publish with an open license
 - All source code and documentation MUST be licensed such that it may be freely reusable, changeable and redistributable.
 - Software source code MUST be licensed under an OSI-approved or FSF Free/Libre license.
 - All source code MUST be published with a license file.
 - Contributors MUST NOT be required to transfer copyright of their contributions to the codebase.
 - All source code files in the codebase SHOULD include a copyright notice and a license header that are machine-readable.
 - Having multiple licenses for different types of source code and documentation is OPTIONAL.
- Make the codebase findable
 - The name of the codebase SHOULD be descriptive and free from acronyms, abbreviations, puns or organizational branding.
 - The codebase SHOULD have a short description that helps someone understand what the codebase is for or what it does.
 - Maintainers SHOULD submit the codebase to relevant software catalogs.
 - The codebase SHOULD have a website which describes the problem the codebase solves using the preferred jargon of different potential users of the codebase (including technologists, policy experts and managers).
 - The codebase SHOULD be findable using a search engine by codebase name.
 - The codebase SHOULD be findable using a search engine by describing the problem it solves in natural language.
 - The codebase SHOULD have a unique and persistent identifier where the entry mentions the major contributors, repository location and website.
 - The codebase SHOULD include a machine-readable metadata description, for example in a publiccode.yml file.
 - A dedicated domain name for the codebase is OPTIONAL.
 - Regular presentations at conferences by the community are OPTIONAL.
- Use a coherent style
 - The codebase MUST use a coding or writing style guide, either the codebase community's own or an existing one referred to in the codebase.
 - Contributions SHOULD pass automated tests on style.
 - The style guide SHOULD include expectations for inline comments and documentation for non-trivial sections.
 - Including expectations for understandable English in the style guide is OPTIONAL.
- Document codebase maturity
 - The codebase MUST be versioned.
 - The codebase MUST prominently document whether or not there are versions of the codebase that are ready to use.
 - Codebase versions that are ready to use MUST only depend on versions of other codebases that are also ready to use.

Standard for Public Code version 0.8.0 checklist

- Code in the open
 - All source code for any software in use (unless used for fraud detection) MUST be published and publicly accessible.
 - All source code for any policy in use (unless used for fraud detection) MUST be published and publicly accessible.
 - The codebase MUST NOT contain sensitive information regarding users, their organization or third parties.
 - Any source code not currently in use (such as new versions, proposals or older versions) SHOULD be published.
 - Documenting which source code or policy underpins any specific interaction the general public may have with an organization is OPTIONAL.
- Bundle policy and source code
 - The codebase MUST include the policy that the source code is based on.
 - If a policy is based on source code, that source code MUST be included in the codebase, unless used for fraud detection.
 - Policy SHOULD be provided in machine-readable and unambiguous formats.
 - Continuous integration tests SHOULD validate that the source code and the policy are executed coherently.
- Make the codebase reusable and portable
 - The codebase MUST be developed to be reusable in different contexts.
 - The codebase MUST be independent from any secret, undisclosed, proprietary or non-open licensed software or services for execution and understanding.
 - The codebase SHOULD be in use by multiple parties.
 - The roadmap SHOULD be influenced by the needs of multiple parties.
 - The development of the codebase SHOULD be a collaboration between multiple parties.
 - Configuration SHOULD be used to make source code adapt to context specific needs.
 - The codebase SHOULD be localizable.
 - Source code and its documentation SHOULD NOT contain situation-specific information.
 - Codebase modules SHOULD be documented in such a way as to enable reuse in codebases in other contexts.
 - The software SHOULD NOT require services or platforms available from only a single vendor.
- Welcome contributors
 - The codebase MUST allow anyone to submit suggestions for changes to the codebase.
 - The codebase MUST include contribution guidelines explaining what kinds of contributions are welcome and how contributors can get involved, for example in a "CONTRIBUTING" file.
 - The codebase MUST document the governance of the codebase, contributions and its community, for example in a "GOVERNANCE" file.
 - The contribution guidelines SHOULD document who is expected to cover the costs of reviewing contributions.
 - The codebase SHOULD advertise the committed engagement of involved organizations in the development and maintenance.
 - The codebase SHOULD have a publicly available roadmap.
 - The codebase SHOULD publish codebase activity statistics.
 - Including a code of conduct for contributors in the codebase is OPTIONAL.

Full self-assessment

and the Standard for Public Code version 0.8.0

Link to commitment to meet the Standard for Public Code:

[Code in the open](#)

criterion met.

Meets

Requirement

Notes and links

All source code for any software in use (unless used for fraud detection) **MUST** be published and publicly accessible.
All source code for any policy in use (unless used for fraud detection) **MUST** be published and publicly accessible.
The codebase **MUST NOT** contain sensitive information regarding users, their organization or third parties.
Any source code not currently in use (such as new versions, proposals or older versions) **SHOULD** be published.
Documenting which source code or policy underpins any specific interaction the general public may have with an organization is **OPTIONAL**.

[Bundle policy and source code](#)

criterion met.

Meets

Requirement

Notes and links

The codebase **MUST** include the policy that the source code is based on.
If a policy is based on source code, that source code **MUST** be included in the codebase, unless used for fraud detection.
Policy **SHOULD** be provided in machine-readable and unambiguous formats.
Continuous integration tests **SHOULD** validate that the source code and the policy are executed coherently.

[Make the codebase reusable and portable](#)

criterion met.

Meets

Requirement

Notes and links

The codebase **MUST** be developed to be reusable in different contexts.
The codebase **MUST** be independent from any secret, undisclosed, proprietary or non-open licensed software or services for execution and understanding.
The codebase **SHOULD** be in use by multiple parties.
The roadmap **SHOULD** be influenced by the needs of multiple parties.
The development of the codebase **SHOULD** be a collaboration between multiple parties.
Configuration **SHOULD** be used to make source code adapt to context specific needs.
The codebase **SHOULD** be localizable.
Source code and its documentation **SHOULD NOT** contain situation-specific information.
Codebase modules **SHOULD** be documented in such a way as to enable reuse in codebases in other contexts.
The software **SHOULD NOT** require services or platforms available from only a single vendor.

Standard for Public Code

- <https://github.com/standard-for-public-code/>
- Review template available
- Community discussion on how to surface usage of the standard