

Wildhainweg 9
Case postale 6935
3001 Berne
Tél. 031 633 59 00
Fax 031 633 59 99
www.be.ch/oio
info.kaio@fin.be.ch



Code source et documentation

Check-lists

Date d'édition	27 novembre 2018
Version	1.0
Statut du document	Validé
Classification	-
Auteur	Joos Thomas, FIN-KAIO-AP-SW
Numéro du document	270319

Sommaire

1	Fonction	3
2	Check-lists	4
2.1	Check-list pour la documentation.....	4
2.2	Check-list pour le code source.....	6
3	Documentation associée à la check-list « Documentation »	7
3.1	Documentation avec le code source.....	7
3.2	Destinataires de la documentation et structure.....	7
3.3	Description de l'objet.....	8
3.4	Indication du caractère libre du logiciel et dépôt de la licence.....	8
3.5	Principales fonctionnalités du projet.....	9
3.6	Stade de développement.....	9
3.7	Version de démonstration.....	9
3.8	Interlocuteur / interlocutrice spécialiste du domaine et propriétaire du projet.....	9
3.9	Documentation pour l'installation.....	9
3.10	Guide du développeur.....	10
3.11	Documentation du développeur.....	12
4	Documentation associée à la check-list « Code source »	13
4.1	Historique du code source.....	13
4.2	Données sensibles, dignes de protection ou confidentielles.....	13
4.3	Suppression des informations sur les auteurs et auteures.....	13
4.4	Prise en compte des licences de bibliothèques.....	14
4.5	Utilisation de gestionnaires de paquets.....	15
4.6	Déclaration de la licence de projet dans le descripteur du gestionnaire de paquets.....	15
4.7	Attributions et informations sur les droits de propriété intellectuelle.....	15
	Sources et licence	17
	Procès-verbal du document	18

1 Fonction

Ces check-lists sont des outils destinés aux spécialistes TIC. Elles les aident à vérifier la documentation et le code source d'un projet. Elles leur permettent de s'assurer que la qualité de la documentation est suffisante pour la distribution d'une application sous licence libre et que le code source et tous les composants de tiers utilisés dans le projet ne présentent aucun risque juridique.

Elles garantissent en outre que la documentation du projet s'adresse tout aussi bien aux utilisateurs finaux qu'aux spécialistes.

2 Check-lists

2.1 Check-list pour la documentation

	Cf. point
<input type="checkbox"/> La documentation est enregistrée avec le code source.	3.1
<input type="checkbox"/> La documentation s'adresse aux utilisateurs finaux et aux spécialistes.	3.2
<input type="checkbox"/> La documentation de présentation est enregistrée dans un fichier <code>README.md</code> .	3.2
<input type="checkbox"/> La licence du projet est enregistrée dans un fichier <code>LICENSE</code> . La partie variable du projet de licence a été adaptée.	3.2
<input type="checkbox"/> Le fichier <code>README.md</code> contient une description de l'objet.	3.3
<input type="checkbox"/> Le fichier <code>README.md</code> indique que le projet concerne des logiciels sous licence libre et comporte un lien vers le fichier <code>LICENSE</code> .	3.4
<input type="checkbox"/> Les principales fonctions du projet sont indiquées dans le fichier <code>README.md</code> .	3.5
<input type="checkbox"/> Le stade actuel du développement est exposé dans le fichier <code>README.md</code> .	3.6
<input type="checkbox"/> Une version de démonstration des applications est disponible et le fichier <code>README.md</code> comporte un lien vers cette version.	3.7
<input type="checkbox"/> L'interlocuteur / interlocutrice technique pour ce projet a été nommé-e et son nom est indiqué dans le fichier <code>README.md</code> .	3.8
<input type="checkbox"/> Le propriétaire du projet, par exemple une association spécialement fondée à cet effet, est indiqué dans le fichier <code>README.md</code> .	3.8
<input type="checkbox"/> Une documentation d'installation des applications est disponible et le fichier <code>README.md</code> comporte un lien vers elle.	3.9
<input type="checkbox"/> Le fichier <code>README.md</code> renvoie au Guide du développeur (Developer Guidelines).	3.10
<input type="checkbox"/> Le fichier <code>README.md</code> renvoie à la documentation du développeur (Developer Documentation).	3.11
<input type="checkbox"/> Le guide du développeur est enregistré dans un fichier <code>CONTRIBUTING.md</code> .	3.10
<input type="checkbox"/> Un code de conduite est défini pour le projet et est enregistré avec le code source dans un fichier Markdown ; le fichier <code>CONTRIBUTING.md</code> comporte un lien vers ce code de conduite.	3.10
<input type="checkbox"/> Le guide du développeur précise que les audits de code sont réalisés par pull-request.	3.10

- | | |
|---|------|
| <input type="checkbox"/> Le guide du développeur précise que la liste des mentions des droits d'auteur et des droits d'attribution doit être actualisée en cas de modification des bibliothèques logicielles utilisées. | 3.10 |
| <input type="checkbox"/> Le guide du développeur renvoie à la documentation pour les développeurs. | 3.10 |
| <input type="checkbox"/> La documentation du développeur présente les interfaces techniques du projet et indique quels outils sont nécessaires pour le développement. | 3.11 |
| <input type="checkbox"/> La documentation du développeur expose les règles formelles valables pour le code source du projet, concernant par exemple son formatage. | 3.11 |
| <input type="checkbox"/> La documentation du développeur explique comment est organisé le code source du projet et comment tester techniquement les fonctions. | 3.11 |
| <input type="checkbox"/> La documentation du développeur comporte une ébauche de l'architecture avec un descriptif sommaire. | 3.11 |
| <input type="checkbox"/> Si le projet est une application, la documentation du développeur explique comment démarrer l'application pour le développement et le débogage et quels composants de l'infrastructure cela nécessite. | 3.11 |

2.2 Check-list pour le code source

	Cf. point
<input type="checkbox"/> Le code source ne contient aucune donnée sensible, digne de protection ou confidentielle.	4.2
<input type="checkbox"/> L'historique du code source ne contient aucune donnée sensible, digne de protection ou confidentielle.	4.1 4.2
<input type="checkbox"/> La société logicielle a été informée que le nom et l'adresse e-mail de l'auteur du code source seront publics s'ils figurent dans le code source ou le dépôt de gestion de code source.	4.2 4.3
<input type="checkbox"/> Les bibliothèques logicielles utilisées par le projet sont compatibles avec la licence du projet.	4.4
<input type="checkbox"/> Le projet utilise un gestionnaire de paquets pour gérer ses dépendances.	4.5
<input type="checkbox"/> Le projet déclare sa licence dans le descripteur du gestionnaire de paquets correspondant.	4.6
<input type="checkbox"/> Le code source contient un fichier <code>THIRD-PARTY-LICENSES.md</code> indiquant, pour chaque bibliothèque utilisée, le nom du projet de la bibliothèque, la page d'accueil de la bibliothèque et l'identifiant SPDX de la licence de la bibliothèque et comporte un lien vers la licence de la bibliothèque.	4.7
<input type="checkbox"/> Pour les applications, une boîte de dialogue « A propos » renvoyant au fichier <code>THIRD-PARTY-LICENSES.md</code> est disponible.	4.7
<input type="checkbox"/> Le processus d'audit du code assure que le fichier <code>THIRD-PARTY-LICENSES.md</code> sera actualisé à chaque modification apportée à la bibliothèque.	4.7

3 Documentation associée à la check-list « Documentation »

3.1 Documentation avec le code source

La documentation doit être un élément du code source du projet, de sorte que les modifications qui lui sont apportées soient aussi conservées (journalisation des révisions) et que la documentation soit versionnée de la même manière que le projet. Ainsi, la documentation, pour autant qu'elle soit gérée correctement, correspondra automatiquement au stade de développement du projet, et on pourra à tout moment retrouver la documentation de versions plus anciennes du projet.

En outre, le projet donne ainsi aux tiers la possibilité de restituer aisément des modifications apportées à la documentation sous forme de contribution.

Markdown est utilisé comme langage de marquage, car il est simple à écrire, très répandu et peut être lu par une machine. En outre, toutes les plateformes courantes restituent les documents écrits en Markdown dans un format cible bien lisible. Au moyen d'un générateur de site statique, tel que Jekyll, MkDocs, Gatsby ou Sphinx, les documents en Markdown peuvent être affichés dans n'importe quel format cible, par exemple selon les normes de l'identité visuelle d'une administration, et ce, quelle que soit la plateforme utilisée.

3.2 Destinataires de la documentation et structure

Étant donné qu'elle s'adresse tant aux utilisateurs finaux qu'aux spécialistes de l'informatique, la documentation du projet ne doit pas être centrée sur les aspects techniques.

C'est le fichier `README.md` qui sert d'entrée en matière. Toutes les plateformes courantes reconnaissent et présentent les fichiers portant ce nom comme un fichier donnant accès à la documentation.

Le fichier `README.md` a pour objet de présenter succinctement le but, la portée, le stade, la licence et les groupes-cibles du projet. Concrètement, le fichier `README.md` doit avoir la teneur suivante :

- description de l'objet ;
- renvoi à la licence dans un fichier `LICENSE` ;
- principales fonctions de base du projet ;
- stade de développement ;
- renvoi à une version de démonstration ;
- indication de l'interlocutrice ou de l'interlocuteur technique et du propriétaire du projet ;
- renvoi à une documentation pour l'installation ;
- renvois au guide du développeur dans un fichier `CONTRIBUTING.md` et à la documentation du développeur.

Exemple de bloc de code¹ `README.md`

```
# Nom du projet
```

Description de l'objet

Nom du projet est un logiciel à code source ouvert qui fait l'objet d'une licence en vertu de la [licence BSD 3 clauses](LICENCE).

Nom du projet propose les fonctions suivantes :

- * Fonction n° 1
- * Fonction n° 2
- * Fonction n° 3
- * Fonction n° 4
- * Fonction n° 5

Stade de développement

Une démo du Nom du projet est disponible à l'adresse https://demo.projekt-name.ch.

Il est possible d'ouvrir une session avec le nom d'utilisateur « demo » et le mot de passe « demo ».

Le spécialiste ou le spécialiste du domaine à contacter concernant le Nom du projet est Prénom Nom. Il ou elle est joignable à l'adresse info.kaio@fin.be.ch.

Utiliser Nom du projet

* [Installation de Nom du projet](docs/install-instructions.md)

Contribuer au développement de Nom du projet

Veuillez lire les documents suivants pour contribuer au développement de Nom du projet :

- * [Guide du développeur](CONTRIBUTING.md)
- * [Documentation du développeur](docs/development.md)

3.3 Description de l'objet

Une brève description de l'objet doit figurer en introduction du fichier `README.md`. Elle permet au lecteur ou à la lectrice de décider en quelques instants si le projet l'intéresse ou pas. Autant dire que cette description doit être concrète et surtout concise.

Pour de plus amples explications sur la description de l'objet (avec des exemples), voir [Produire un logiciel libre, Fixer des finalités claires](#).

3.4 Indication du caractère libre du logiciel et dépôt de la licence

Il faut, immédiatement après la description de l'objet, préciser sans ambiguïté que le projet est libre. Il faut indiquer la licence utilisée par le projet en précisant son [identifiant SPDX](#) et ajouter

un lien avec le texte de la licence figurant dans le fichier `LICENSE`. Généralement, les modèles de licence contiennent une partie variable (nom du projet, année du droit d'auteur, nom de l'auteur, p.ex.) qu'il faut modifier dans le fichier `LICENSE`.

La plupart des plateformes reconnaissent les licences figurant dans un fichier `LICENSE` et proposent des informations claires sur la licence, comme un résumé de ses conditions.

Pour de plus amples explications, voir [Produire un logiciel libre, Préciser que le projet est libre](#).

3.5 Principales fonctionnalités du projet

Décrites dans le fichier `README.md`, les principales fonctionnalités complètent la description de l'objet.

Pour de plus amples explications, voir [Produire un logiciel libre, Lister les fonctionnalités et les pré-requis](#).

3.6 Stade de développement

Il faut préciser le stade de développement d'un projet dans le fichier `README.md` pour que les personnes intéressées puissent s'informer rapidement de son degré de maturité. Il est en effet important pour le lecteur ou la lectrice de savoir si le projet est déjà mûr ou ne se trouve qu'au début de son développement, s'il bénéficie d'une maintenance rapprochée et à quelle fréquence de nouvelles versions sont publiées.

Le titulaire peut aussi indiquer sous cette rubrique le soutien dont son projet a le plus besoin à ce stade. Il peut s'agir d'un développeur ou d'une développeuse qui dispose de connaissances techniques précises ou d'une personne qui revoit la documentation du projet.

Pour de plus amples explications, voir [Produire un logiciel libre, Informer sur le statut de développement](#).

3.7 Version de démonstration

Pour les applications, il est recommandé de proposer une version de démonstration afin que les utilisateurs et utilisatrices puissent l'essayer sans l'installer. Selon le genre de l'application, la version de démonstration peut être la version définitive ou une version test. Il est important que le lecteur ou la lectrice ait immédiatement accès, par eux-mêmes, à la version démo.

3.8 Interlocuteur / interlocutrice spécialiste du domaine et propriétaire du projet

Le fichier `README.md` indique l'interlocuteur ou l'interlocutrice spécialiste du domaine, afin que d'autres administrations ou organisations intéressées puissent le contacter le plus directement possible.

Les adresses électroniques personnelles sont à éviter. A défaut d'une adresse convenable dans une DIR/CHA/JUS ou dans l'association responsable, il est aussi possible d'utiliser à cet effet l'adresse générale de l'OIO (info.kaio@fin.be.ch).

Le fichier `README.md` doit également préciser l'organisation à qui appartient le projet, en particulier lorsqu'une association a été fondée pour gérer celui-ci.

3.9 Documentation pour l'installation

Pour les applications, il convient que le fichier `README.md` fasse référence à un document qui précise les exigences en matière de logiciel et de matériel, les composants nécessaires (comme une base de données) et les instructions d'installation.

3.10 Guide du développeur

L'accès au Guide du développeur figure dans le fichier Markdown `CONTRIBUTING.md`, vers lequel le fichier `README.md` contient un lien. Ce guide du développeur fait partie de la documentation générale conçue pour les développeurs et les développeuses désireux de contribuer au projet et s'occupe davantage de la collaboration et de la communication que des aspects techniques.

A cet effet, il doit aborder les aspects suivants et, le cas échéant, mentionner les liens menant vers les documents mentionnés :

- un renvoi au code de conduite du projet, qui énonce les normes sociales à respecter au sein de celui-ci ;
- il est conseillé de choisir comme code de conduite la [Charte Code de conduite contributeurs](#) sous licence [CC-BY-4.0](#) ou de la prendre comme base ;
- l'indication selon laquelle les audits du code se font dans le projet sous la forme de pull requests de GitHub, avec un renvoi à la documentation de GitHub sur ce sujet ;
- en dernier lieu, il faut mentionner la « Documentation développeurs ».

Exemple de bloc de code2 CONTRIBUTING.md

```
# Collaborer au développement de Nom du projet

Nous nous réjouissons de recevoir vos contributions à ce projet et vous prions de
tenir compte des aspects suivants.

# Audits du code

Nous réalisons un audit pour toutes les modifications du code, y compris celles
réalisées par les membres du projet.
Nous utilisons à cet effet les pull requests de GitHub. Lisez
[l'Aide GitHub](https://help.github.com/articles/about-pull-requests/) pour en sa-
voir plus sur les pull requests.

# Code de conduite

Veuillez lire et respecter notre [Code de conduite](CODE-OF-CONDUCT.md).
```

Exemple de bloc de code 3 CODE-OF-CONDUCT.md ([Charte Code de conduite contributeurs](#))

```
# Charte Code de conduite contributeurs

## Notre engagement

Dans l'intérêt de favoriser un environnement ouvert et accueillant, nous nous en-
```

gations, en tant que responsables et en tant que personnes contribuant à ce projet, à faire de la participation une expérience exempte de harcèlement pour tout le monde, quel que soit le niveau d'expérience, le sexe, l'identité ou l'expression de genre, l'orientation sexuelle, le handicap, l'apparence personnelle, la taille physique, l'origine ethnique, l'âge, la religion ou la nationalité.

Nos critères

Exemples de comportements qui contribuent à créer un environnement positif :

- * l'utilisation d'un langage ouvert et accueillant ;
- * le respect des différents points de vue et expériences vécues ;
- * accepter poliment les critiques constructives ;
- * se concentrer sur ce qui est meilleur pour la communauté ;
- * faire preuve d'empathie envers les autres membres de la communauté.

Exemples de comportements non acceptables :

- * l'utilisation de langage ou d'imagerie sexualisés et les avances sexuelles non sollicitées ;
- * le *trolling*, les commentaires insultants ou désobligeants, et les attaques personnelles ou d'ordre politique ;
- * le harcèlement en public ou en privé ;
- * la publication d'informations privées de tierces personnes, telles que des adresses physiques ou électroniques, sans permission explicite ;
- * toute conduite qui pourrait être raisonnablement considérée comme inappropriée dans le milieu professionnel.

Nos responsabilités

Les responsables du projet doivent clarifier les critères de comportement acceptables de ce projet : il est attendu que ces personnes prennent les mesures correctives justes comme réponse à tout comportement inacceptable.

Les personnes qui assurent la maintenance du projet ont le droit et la responsabilité de supprimer, modifier ou rejeter les commentaires, *commits*, code, modifications du wiki, questions et autres contributions qui ne respectent pas ce Code de Conduite, ou de bannir temporairement ou définitivement quiconque, suite à des comportements jugés inappropriés, menaçants, injurieux, ou nuisibles.

Objectifs

Ce Code de Conduite s'applique à la fois au sein des espaces du projet ainsi que dans les espaces publics lorsqu'un individu représente le projet ou sa communauté. Font parties des exemples de représentation d'un projet ou d'une communauté le fait d'utiliser une adresse email propre au projet, de poster sur les réseaux sociaux avec un compte officiel, ou d'intervenir pour représenter le projet au cours d'un événement en-ligne ou hors-ligne. La représentation du projet pourra être autrement définie et clarifiée par les responsables du projet.

Application

Les cas de comportements abusifs, harcelants ou tout autre comportement inacceptable peuvent être signalés en contactant l'équipe du projet à [INSÉRER UNE ADRESSE EMAIL]. Toutes les plaintes seront examinées et étudiées et se traduiront par une réponse appropriée aux circonstances. L'équipe du projet s'engage à garder confidentielles les informations de la personne qui remonte un incident. Plus de détails sur la politique de mise en application des règles peuvent être publiés séparément.

Les membres du projet qui ne suivent ou qui n'appliquent pas le Code de Conduite de bonne foi s'exposent temporairement ou de façon permanente à des répercussions définies par d'autres membres de la direction du projet.

Attribution

Ce Code de Conduite est adapté du [Contributor Covenant][homepage], version 1.4, disponible à <https://www.contributor-covenant.org/fr/version/1/4/code-of-conduct>

[homepage]: <https://www.contributor-covenant.org>

3.11 Documentation du développeur

Contrairement au guide du développeur, centré sur la collaboration et les règles de conduite, la documentation du développeur présente les caractéristiques techniques qui conditionnent la participation au développement du projet. Cette documentation doit aborder au minimum les aspects suivants :

- interfaces techniques du projet et outils nécessaires à son développement ;
- règles formelles applicables au code source du projet, s'agissant par exemple de son formatage ;
- organisation du code source du projet et procédé de test technique des fonctions ;
- ébauche de l'architecture avec un descriptif sommaire de celle-ci, y compris, si possible, une délimitation du contexte et le premier niveau de la vision modulaire selon [arc42](#) ;
- pour les applications : procédure visant à démarrer l'application à des fins de développement ou de débogage et les éléments d'infrastructure requis.

4 Documentation associée à la check-list « Code source »

4.1 Historique du code source

Le code source d'un projet est d'habitude enregistré dans un système de gestion de code source (SCM), comme GitHub. Ces systèmes n'enregistrent pas seulement la dernière version du code, mais aussi toutes les modifications qu'il a subies au fil du temps, en particulier les éléments effacés, y compris les fichiers.

Le SCM comprend par ailleurs les méta-informations sur les modifications du code source, comme le nom et le courriel de l'auteur ou de l'auteure ou les signatures PGP des étiquettes et des *commits* signés.

4.2 Données sensibles, dignes de protection ou confidentielles

Le code source et l'historique du code source du projet peuvent contenir des données à caractère public, comme une liste des numéros postaux d'acheminement, des données aléatoires ou des données de synthèse.

En revanche, ils ne peuvent pas contenir de données ou d'informations sensibles, dignes de protection ou confidentielles, et en particulier les suivantes :

- noms d'utilisateurs et mots de passe ou d'autres données secrètes, comme des clés privées, des jetons d'accès ou des certificats ;
- noms DNS internes, URL, noms d'hôte, adresses IP, structures des réseaux ou désignations d'unités de disque ;
- données anonymisées ou pseudonymisées, car il subsiste toujours une possibilité d'annuler l'anonymisation ou la pseudonymisation.

Même si la dernière version du code source ne contient pas de données ou d'informations sensibles, dignes de protection ou confidentielles, ces données ou ces informations peuvent être récupérées de l'historique du système SCM lorsqu'elles figuraient par le passé dans le code source ou dans les méta-informations. Dès lors, la mise au net de la dernière version du code source ne les supprime pas totalement.

En cas de doute, il est dès lors recommandé, avant de publier le code source, de supprimer totalement son historique après l'avoir mis au net.

Les noms et les courriels des auteurs et auteures du code source figurent généralement dans le code source ou dans les méta-informations du SCM, par exemple dans les *commits* ou dans les étiquettes annotées de GitHub. Si cette pratique ne pose pas de problème du point de vue juridique (les règles applicables à la publication de ces informations étant du ressort de la société logicielle), il est cependant recommandé d'attirer l'attention de celle-ci sur le fait qu'il est possible que les noms et les courriels des auteurs et auteures du code source soient publiés en même temps que le code source.

4.3 Suppression des informations sur les auteurs et auteures

Il est possible de modifier l'historique de GitHub de sorte à rendre méconnaissables les informations sur les auteurs et auteures. C'est la fonction du script ci-dessous publié par GitHub :

<https://help.github.com/articles/changing-author-info/>

4.4 Prise en compte des licences de bibliothèques

Tous les projets ou presque utilisent des bibliothèques (logithèques, dépendances, etc.) de tiers pour faire appel à des fonctions générales sans devoir les programmer. La plupart de ces bibliothèques sont elles-mêmes dépendantes d'autres bibliothèques, ce qui engendre des dépendances multiples (dépendances transitives). Des centaines de dépendances sont la règle même pour des projets de petite taille, leur nombre dépendant du langage de programmation utilisé.

Il faut s'assurer, pour chaque dépendance, que sa licence est compatible avec la licence du projet. Des problèmes particuliers se posent lorsque les dépendances n'ont pas de licence open source ou de licence déclarée ou utilisent une licence virale qui ne concorde pas avec la licence du projet. Certaines bibliothèques ont plus d'une licence et permettent à l'utilisateur ou à l'utilisatrice de choisir celle qui lui convient.

Tous les langages de programmation courants permettent d'établir automatiquement des listes des dépendances du projet (y compris des dépendances transitives) et des licences des dépendances. Il s'agit notamment des outils suivants :

- le [Project Info Reports Plugin](#) du gestionnaire de paquets Apache Maven pour les projets Java, qui crée, dans le rapport Dépendances, une liste de toutes les bibliothèques utilisées, y compris la licence déclarée, au format HTML ;
- le [NPM License Checker](#) pour le gestionnaire de paquets NPM des projets JavaScript, qui peut établir une liste de toutes les bibliothèques utilisées, y compris la licence déclarée, en divers formats, comme CSV, et
- le [License Finder de Pivotal](#), compatible avec divers gestionnaires de paquets et langages de programmation, comme Ruby Gems, Python Eggs ou Godeps.

4.4.1 Utilisation de bibliothèques sous licences propriétaires

Les bibliothèques sous licences propriétaires posent des problèmes de fond dans les projets open source et ne peuvent en principe pas être utilisées, en particulier lorsque le projet de logiciel libre a été distribué sous licence virale, comme AGPL.

Nous vous prions de prendre contact avec le fabricant, de concert avec le service juridique de l'OIO, afin d'établir si la bibliothèque propriétaire peut être utilisée dans votre projet et à quelles conditions.

4.4.2 Pour les projets sous licence BSD 3-Clause, MIT ou Apache 2.0

Pour les projets sous licence BSD 3-Clause, MIT ou Apache 2.0, il est possible d'utiliser les bibliothèques créées sous les licences suivantes (sélection) :

Apache 2.0, Apache 1.0, BSD 3-Clause, BSD 2-Clause, ISC, LGPL (lié), MIT et MPL.

Les bibliothèques créées sous les licences suivantes ne peuvent pas être utilisées (sélection) :

AGPL, BSD 4-Clause, EUPL, GPL, GPL-SIK et LGPL (non lié).

Si la licence d'une bibliothèque ne figure pas dans les listes, prière de contacter le service juridique de l'OIO pour plus d'informations.

4.4.3 Pour les projets sous licence EUPL

Pour les projets sous licence EUPL, il est possible d'utiliser les bibliothèques créées sous les licences suivantes (sélection) :

Apache 2.0, BSD, EUPL, LGPL (lié), MIT et MPL.

Les bibliothèques créées sous les licences suivantes ne peuvent pas être utilisées (sélection) :

AGPL, GPL et LGPL (non lié).

Si la licence d'une bibliothèque ne figure pas dans les listes, prière de contacter le service juridique de l'OIO pour plus d'informations.

4.5 Utilisation de gestionnaires de paquets

Tous les langages de programmation courants ou presque proposent des gestionnaires de paquets pour gérer leurs dépendances, comme Apache Maven pour les projets Java, NPM pour les projets JavaScript ou NuGet pour les projets .NET.

Lorsqu'un gestionnaire de paquets est utilisé, les dépendances utilisées par le projet ne font plus directement partie du code source du projet, ce qui est le cas lorsque, par exemple, le code source de la dépendance est copié dans le projet ou que les dépendances sont copiées dans le projet au format binaire. En lieu et place, le projet se procure toutes les dépendances par le biais de déclarations du gestionnaire de paquets.

On s'assure ainsi que le gestionnaire de paquets peut établir la liste de toutes les dépendances du projet et on évite que le code source des dépendances soit modifié dans le projet sans notifier cette modification aux auteurs et auteures de la dépendance.

4.6 Déclaration de la licence de projet dans le descripteur du gestionnaire de paquets

Le projet devrait déclarer sa licence dans le descripteur ad hoc du gestionnaire de paquets au moyen de l'[identifiant SPDX](#), afin que sa licence puisse être publiée par le biais du gestionnaire de paquets. Cela est particulièrement important pour les bibliothèques, afin que les utilisateurs et utilisatrices de la bibliothèque puissent automatiquement consulter sa licence par le biais du gestionnaire de paquets.

4.7 Attributions et informations sur les droits de propriété intellectuelle

De nombreuses bibliothèques sont soumises à une licence qui exige de son utilisateur ou de son utilisatrice qu'il indique le droit de propriété intellectuelle (*original copyright notice*) ou reconnaissent l'utilisation (*attribution*) de la bibliothèque.

Il s'agit notamment des licences suivantes :

[Apache 2.0](#), [ASL 1.1](#) (Apache 1.1), [BSD](#), [BSD 3-Clause](#), [Creative Commons "Attribution" \(CC BY\)](#), [MIT](#) et [ISC](#).

Pour d'autres licences, consulter [Google Open Source Documentation, Licenses - Notice](#).

Cela veut dire que, en pratique, presque toutes les bibliothèques utilisées sont concernées. Même les projets de faible envergure utilisent une multitude de bibliothèques, de sorte qu'il est très laborieux d'indiquer les droits de propriété intellectuelle ou l'utilisation des bibliothèques dans le respect des dispositions légales.

Dès lors, il est conseillé de procéder comme suit :

- établir une liste des bibliothèques utilisées en faisant appel aux fonctions du gestionnaire de paquets ou à des outils tels que NPM License Checker ou License Finder de Pivotal ;
- mettre la liste en forme afin qu'elle contienne les informations suivantes : nom officiel du projet de la bibliothèque, site Internet du projet de la bibliothèque, licence utilisée de la bibliothèque, sous la forme d'un identifiant SPDX avec un lien vers le texte de la licence ;
- enregistrer la liste dans le code source du projet, par exemple dans un fichier `THIRD-PARTY-LICENSES.md` ;
- pour les applications : ajouter un lien sous la rubrique « A propos de » ou dans une boîte de dialogue « A propos de cette application » pour renvoyer au fichier `THIRD-PARTY-LICENSES.md` ;
- l'utilisation de la rubrique « A propos de » est très répandue dans ce but, par exemple sur Google Chrome (sous la rubrique « A propos de Chrome ») ;
- pour les bibliothèques : mettre dans le fichier `README.md` un renvoi au fichier `THIRD-PARTY-LICENSES.md` ;
- s'assurer, lors de l'audit du code, que les modifications des bibliothèques ont été reportées dans le fichier `THIRD-PARTY-LICENSES.md`.

Exemple de bloc de code 4 THIRD-PARTY-LICENSES.md

```
Ce projet utilise des logiciels open source :  
  
* Spring Boot, http://projects.spring.io/spring-boot/, sous licence [Apache-2.0] (http://www.apache.org/licenses/LICENSE-2.0)  
* caniuse-db, https://github.com/Fyrd/caniuse, sous licence [CC-BY-4.0] (https://creativecommons.org/licenses/by/4.0/)
```


Sources et licence

La check-list, la documentation associée et les modèles de documents correspondants sont tirés en partie des sources suivantes :

- [Google Open Source Docs](#) de [Google LLC](#), sous licence [CC-BY-4.0](#)
- [Produire un logiciel libre](#), [How to run a Successful Free Software Project](#) de [Karl Fogel](#), sous licence [CC-BY-SA-4.0](#).

Compte tenu des sources utilisées, le résultat correspondant accessible au public est sous licence libre [CC-BY-SA-4.0](#).

Procès-verbal du document

N° de document #287254

Auteur Joos Thomas, FIN-KAIO-AP-SW

Suivi des modifications

Version	Nom	Date	Remarques
1.0	Ferdinand Hübner	27.02.2018	Création
0.2	Mario Siegenthaler	14.03.2018	Compléments et transfert dans Word
0.3	Thomas Joos	01.05.2018	Compléments et adaptations mineurs
0.4	Ferdinand Hübner	14.05.2018	Suppression des CLA et des en-têtes de licence dans les fichiers. Remplacement d'AGPL par EUPL. Correction de la remarque concernant la publication du nom et du courriel des auteurs et auteures du code source. Intégration du feed-back de Cornelia Nussberger
0.5.-0.7	Thomas Joos	13.06.2018	Révisions
0.8	Thomas Joos	13.06.2018	Finalisation
1.0	Thomas Joos	04.07.2018	Version définitive après approbation par le PB
1.1	Thomas Joos	12.09.2018	Corrections des erreurs, modifications mineures et formulation inclusive
1.2.-1.3	Stefan Schneider	13.09.2018	Révision
1.4	Thomas Joos	18.09.2018	Finalisation

Contrôle

Version	Service	Date	Visa	Remarque
0.3	TTE	09.05.2018	C. Nussberger	Remarques et adaptations mineures
0.8	Thomas Joos	13.06.2018	Tjo	---
0.9	PB	02.07.2018	PB	Approbation par le Portofolio board
1.2	État-major	13.09.2018	Ssc	Vérification orthographique

Validation

Version	Service	Date	Visa	Remarque
1.0	Service / responsable de domaine	25.06.2018	mwe / rae	---
0.9	PB	02.07.2018	PB	Validation par le Portofolio board