

MooX

– Manuel du développeur –

Table des matières

| | |
|--|---|
| I. Comprendre le logiciel..... | 3 |
| I.1 Objectifs du logiciel..... | 3 |
| I.2 Documentation..... | 3 |
| I.2.1 Documentation de conception..... | 3 |
| I.2.2 Javadoc..... | 3 |
| I.2.3 Autres ressources..... | 3 |
| II. Développer avec la communauté..... | 4 |
| II.1 L'accès au code source..... | 4 |
| II.1.1 GIT..... | 4 |
| II.1.2 Les branches..... | 4 |
| II.1.3 Dépendances..... | 5 |
| II.1.4 Style..... | 5 |
| II.2 L'affectation des tâches..... | 5 |
| II.3 Les tests..... | 5 |
| III. Licence..... | 6 |
| III.1 Vos droits..... | 6 |
| IV Liens utiles..... | 6 |

I. Comprendre le logiciel

I.1 Objectifs du logiciel

L'objectif du logiciel MooX est de convertir le format des quiz proposés par la plateforme moodle. Ces quiz sont originellement au format XML (avec des règles de mise en forme précises), et MooX est chargé de les convertir dans un format JSON. Cette action doit pouvoir être réversible et MooX doit donc être capable de prendre un fichier JSON ainsi généré pour être ensuite reconverti en un fichier XML (respectant la mise en forme des quiz).

I.2 Documentation

Si vous souhaitez contribuer au développement de MooX, il vous faudra obligatoirement comprendre son fonctionnement. Pour cela, une documentation de conception est disponible avec le code source de l'application¹ dans le dossier 'Doc'. Une Javadoc est également fournie.

I.2.1 Documentation de conception

Cette documentation présente l'architecture globale du logiciel. Il est vivement conseillé de lire ce document et en particulier la partie concernant les interactions entre les modules avant de se lancer dans la compréhension du code en lui même.

En plus de l'architecture globale du logiciel, une explication concernant l'agencement du code source y est incluse.

I.2.2 Javadoc

La javadoc pourra vous être utile dans le cas où vous avez besoin d'interagir avec des modules existants. La javadoc vous fournira les explications nécessaires pour utiliser les méthodes et les classes existantes. Des détails concernant l'implémentation des modules peuvent également être contenu dans la javadoc.

I.2.3 Autres ressources

D'autres ressources complémentaires sont disponibles sur le Google Drive du projet:
<https://drive.google.com/?tab=mo&authuser=0#folders/0B1RGTRXGvDpoeVhyQUM5OEpfNkE>

¹ <https://github.com/Yuyuu/Moox>

II. Développer avec la communauté

Afin de pouvoir développer correctement avec l'ensemble des contributeurs, un certain nombre de règles ont été définies et des logiciels de développement collaboratif sont également nécessaires.

II.1 L'accès au code source

La totalité du code source se situe sur un *repository* GitHub à l'adresse suivante: <https://github.com/Yuyuu/Moox>. Pour pouvoir collaborer sur ce projet, il vous faut avant tout faire une demande d'accès en écriture à l'un des fondateurs du projet (cette demande se fait par courrier électronique). Voici la liste des adresses des fondateurs:

- BERTRAND Nathanaël <nathanael.bertrand@gmail.com>
- SERIN Kévin <lepiaf31@gmail.com>
- SIMAR Jérémie <simar.jeremy@gmail.com>
- TERTRE Vincent <vincent.tertre@gmail.com>
- WARIN Laurent <laurent.warin@sfr.fr>

II.1.1 GIT

Ensuite apports au projet (améliorations, corrections, ...) se font grâce au logiciel GIT². Ce logiciel permet le partage de code source et le développement de plusieurs collaborateurs sur un même projet. Ce document ne détaille pas comment utiliser ce logiciel mais il est primordial de savoir l'utiliser afin de pouvoir contribuer au projet.

II.1.2 Les branches

Afin de séparer les différentes choses qui sont modifiées ou ajoutées au logiciel, des branches ont été définies. Voici un récapitulatif des principales branches:

| Nom | Utilité |
|--------|--|
| Master | C'est la branche principale de la <i>release</i> actuelle du logiciel. Elle contient un code source fonctionnel et stable. |
| Dev | C'est la branche qui contient les nouvelles fonctionnalités du logiciel. Ces nouvelles fonctionnalités formeront la prochaine <i>release</i> du logiciel. Le logiciel présent dans cette branche ne peut être qualifié comme stable. |
| Bug | C'est la branche qui sert à résoudre les bugs rencontrés dans la version actuelle. |

² <http://git-scm.com/>

| | |
|------------|--|
| VersionX.X | Les branches de ce type sont les anciennes versions stables du logiciel. |
|------------|--|

Le respect des rôles de ces branches est primordial pour participer au développement de MooX.

II.1.3 Dépendances

Pour ce projet, un logiciel de gestion des dépendances est utilisé. Le logiciel utilisé est Maven dans sa version 3. Il permettra à l'aide d'un unique fichier pom.xml placé à la racine du projet de gérer les dépendances à des bibliothèques externes. Si vous souhaitez contribuer au projet, il vous faudra donc posséder et maîtriser le logiciel Maven pour pouvoir compiler le projet.

II.1.4 Style

Pour assurer une certaine cohésion entre les codes sources des différents contributeurs, il est demandé de respecter les normes d'Oracle. Pour vérifier cela, des plugins d'analyse statique du code sont fournis dans les dépendances du projet. Il est demandé de vérifier que les rapports générés par ces plugins contiennent un nombre d'erreurs de style raisonnable (de l'ordre de la dizaine).

II.2 L'affectation des tâches

Afin d'éviter que plusieurs contributeurs ne développent la même chose, un système de tâches à faire est mis en place. C'est le gestionnaire de tâches de Github qui est utilisé (ces tâches sont nommées des *issues*). Seuls les fondateurs du projet sont habilités à créer et gérer des tâches. Si développer une tâche vous intéresse, vous devez vous manifester après d'eux et ils vous assigneront la tâche. Vous serez alors en charge de développer la tâche.

Lorsque vous effectuez un *commit* en rapport avec une tâche sur laquelle vous êtes assigné, vous devez préfixer votre message avec un dièse suivi du numéro de la tâche afin que Github puisse faire le lien entre le *commit* et la tâche.

Une fois que le travail aura été validé, l'*issue* sera fermée.

II.3 Les tests

Afin d'assurer la qualité des codes fournis par les contributeurs, des tests passés avec succès doivent également être fournis. Pour chaque classe ajoutée ou modifiée, des tests unitaires doivent être fournis.

Pour réaliser ces tests, le plugin Junit est utilisé. Ce plugin et sa version sont renseignés dans le gestionnaire de dépendances et devrait s'installer automatiquement dans votre environnement de développement.

III. Licence

Le logiciel et le code source sont fournis sous licence GPL³. Par conséquent, toute contribution se verra affectée de la même licence. Les détails de la licence sont disponibles sur le *repository* du projet.

III.1 Vos droits

Toute contribution projet MooX impliquera automatiquement la renonciation de votre part des droits patrimoniaux concernant le code produit ou modifié. Un contrat allant en ce sens vous sera soumis lors de votre demande pour devenir contributeur sur le projet.

IV Liens utiles

- Google Drive su projet:
<https://drive.google.com/?tab=mo&authuser=0#folders/0B1RGTRXGvDpoeVhyQUM5OEpfNkE>
- Repository Github du projet:
<https://github.com/Yuyuu/Moox>
- Adresses mails des fondateurs:
 - BERTRAND Nathanaël <nathanael.bertrand@gmail.com>
 - SERIN Kévin <lepiat31@gmail.com>
 - SIMAR Jérémy <simar.jeremy@gmail.com>
 - TERTRE Vincent <vincent.tertre@gmail.com>
 - WARIN Laurent <laurent.warin@sfr.fr>
- Page officielle de GIT:
<http://git-scm.com/>
- Page officielle de Maven:
<http://maven.apache.org/>
- Normes de codage de Oracle:
<http://www.oracle.com/technetwork/java/codeconv-138413.html>
- Licence GNU GLP:
<http://www.gnu.org/licenses/gpl.html>

3 http://fr.wikipedia.org/wiki/Licence_publicue_g%C3%A9n%C3%A9rale_GNU