Zeste de savoir

Zest Writer 2.0 est disponible

3 août 2020

Table des matières

Zest Writer est un éditeur de texte en markdown dont le but est d'apporter aux auteurs de Zeste de Savoir une batterie d'outils, afin que ce dernier puisse se concentrer sur ce qu'il sait faire de mieux: la rédaction!

Zest Writer vous accompagne même pendant les vacances

Déplacement des extraits/chapitres de votre contenu via drag & drop, conversion d'un billet en article et vice-versa, raccourci claviers, aperçu temps réel du markdown, rédaction hors ligne et autres fonctionnalités qui feront la joie de l'auteur que vous êtes et/ou vous donnera envie de créer votre premier contenu même au fin fond de la creuse sans aucune connexion internet.

Trois ans après la sortie de la version précédente (oui je sais ça fait beaucoup), voici l'arrivée de la version 2.0 de Zest Writer.

La grosse nouveauté de cette version est qu'elle est compatible avec zmd, le moteur markdown de Zeste de Savoir qui a été ré-écrit en JavaScript.

Je vous propose dans la suite de ce billet de vous montrer comment mettre à jour votre version de Zest Writer, ainsi que les apports de cette nouvelle version.

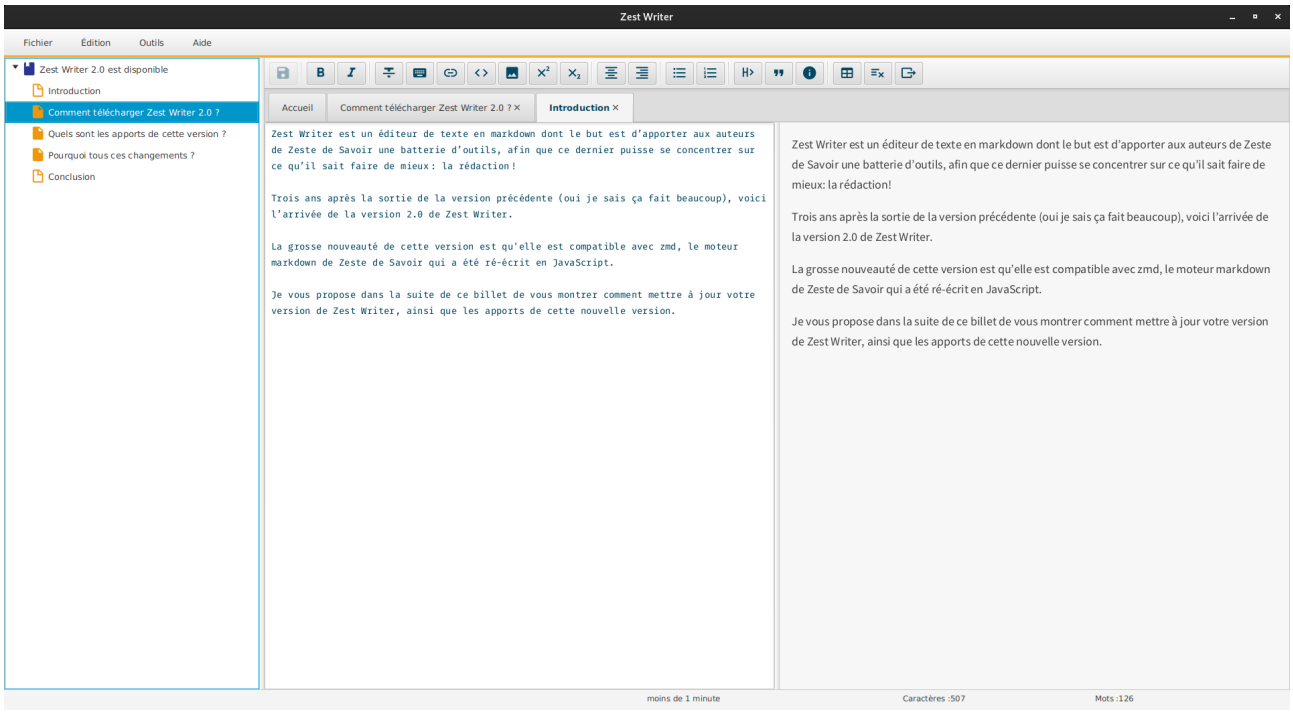

FIGURE 0.1. – Zest Writer 2.0 sous Ubuntu

1. Comment télécharger Zest Writer 2.0 ?

Zest Writer est un logiciel multiplateforme, ce qui signifie que vous pouvez l'avoir aussi sur votre système d'exploitation préféré.

> Avec la version 2.0 de Zest Writer, nous ne distribuons plus les exécutables `jar` pour la simple et bonne raison que certaines JVM ne sont pas compatibles avec Zest Writer. Pour vous éviter des problèmes de JVM et par souci de simplicité, on préfère vous livrer directement le paquet pour votre version qui embarque la bonne JVM.

Pour l'installer, il vous suffit de suivre les instructions ci-dessous.

1.1. Windows

Téléchargez le fichier d'installation et lancez-le: Lien de téléchargement

1.2. OsX

Téléchargez le fichier d'installation et lancez-le: Lien de téléchargement

1.3. Linux

1.3.1. Debian, Ubuntu, Mint, etc.

Dans un terminal:

```
# importez la clé GPG de bintray
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80
    --recv-keys 379CE192D401AB61

# ajoutez une nouvelle source
echo "deb https://dl.bintray.com/zest-writer/deb wheezy main" |
    sudo tee -a /etc/apt/sources.list.d/zestwriter.list

# mettez à jour vos dépôts
sudo apt-get update

# installez Zest Writer
sudo apt-get install zestwriter
```

1.3.2. Fedora, CentOs, etc.

1. Créez le fichier `/etc/yum.repos.d/zestwriter.repo` et copiez le contenu suivant à l'intérieur:

```
[zestwriter]
name=zestwriter
baseurl=http://dl.bintray.com/zest-writer/rpm
gpgcheck=0
enabled=1
```

1. Installez Zest Writer en saisissant dans votre terminal `yum install zestwriter`.

2. Quels sont les apports de cette version ?

Cette version est une grosse refonte de l'application, et même si elle apporte pas mal de correctifs, certaines fonctionnalités ont été retranchées (la plupart reviendront probablement dans les prochaines versions).

2.1. Les nouvelles fonctionnalités

— L'intégration du nouveau moteur zmarkdown: la précédente version fonctionnait encore avec l'ancien moteur markdown de ZdS écrit en Python, un boulot colossal a été fait pour utiliser le nouveau moteur écrit en JavaScript.
— Zest Writer supporte désormais la nouvelle version du format d'import/export de ZdS (version `2.1` du `manifest.json`) avec notamment la gestion de la publication partielle.

2.2. Les bugs corrigés

— #318 : La zone d'édition qui avait tendance à disparaître lors de l'externalisation du rendu est désormais fonctionnelle.
— #120 : Les caractères accentués que l'on ne pouvait pas taper dans la zone de rédaction (ce qui rendait l'écriture laborieuse) sont désormais fonctionnels.

2.3. Les fonctionnalités retranchées

Ces fonctionnalités ont été retranchées temporairement le temps de la livraison d'une première version fonctionnelle. Nous espérons qu'elles reviendront rapidement par la suite.

— La disparition du module de correction orthographique/grammatical
— La disparition de l'export `PDF` (on garde cependant l'export au format `markdown`)
— La zone de rédaction n'affiche plus de numéro de ligne pour se repérer
— La fonction de recherche/remplacer dans la zone de rédaction ne permet plus d'afficher les sélections multiples

Voilà un état des lieux du point de vue utilisateur, de ce qui change pour vous.

> *i* Toutes les versions `< 2.0` deviennent donc obsolètes et non maintenues.

3. Pourquoi tous ces changements ?

On ne se lance pas dans une refonte aussi énorme sans que celle-ci ne soit justifiée. La refonte de Zest Writer a été motivée par deux choses:

1. Intégrer le nouveau moteur zMarkdown de Zeste de Savoir
2. Permettre enfin de pouvoir saisir certains caractères accentués (`ê`, `ï`, etc.)

Ces deux objectifs ont entraîné tout un tas de changement au sein de l'application.

3.1. De Python-Zmarkdown à zMarkdown (JS)

Python-Zmarkdown est l'ancien moteur utilisé par Zeste de Savoir qui permet de convertir un texte en markdown vers du html. Ce dernier souffrait de pas mal de bugs, et commençait à être difficilement maintenable. L'équipe technique du site à donc décidé de crée un nouveau moteur , cette fois-ci le moteur est écrit en JavaScript.

Zest Writer, étant un éditeur hors ligne, il faut savoir qu'avant il embarquait un interpréteur python pour pouvoir traiter le markdown de la même façon que le site sans avoir besoin de se connecter au site (sinon on ne pourrait plus parler d'éditeur hors ligne).

Le défi imposé par le passage au nouveau moteur zMarkdown à Zest Writer était d'arriver à intégrer un interpréteur/moteur JavaScript au sein de Zest Writer qui permette à ce dernier de continuer à parser le markdown de la même façon que le site, tout en restant un éditeur hors ligne.

Après de nombreuses recherches et tests, mon choix s'est porté sur GraalVM. Il s'agit d'une machine virtuelle universelle pour exécuter des applications écrites en JavaScript, Python, Ruby, R, les langages de JVM comme Java, Scala, Kotlin, et les langages LLVM comme C et C++.

Plusieurs choses m'ont séduite avec GraalVM:

— La machine virtuelle est pilotable en Java (le langage dans lequel est écrit Zest Writer) ;
— Le support de la norme ECMA Script 2020 et le fait que sa dernière version se base sur Node.js v12, qui est un gage de modernité et me permet d'être serein quant au futures ~~folies~~ évolutions de zmarkdown ;
— Le coté polyglotte de GraalVM me permet d'entrevoir un module de correction basé sur Grammalecte (qui est écrit en python) ;
— Le fait que ça "juste marche" et reste vraiment multiplate-forme.

Voilà dans l'ensemble comment on arrive, avec les bon outils à passer sur le nouveau moteur de zMarkdown dans sa version 8.3.0.

3.2. De RichTextFX au Textarea

RichTextFX est une bibliothèque écrite en Java qui permet en gros d'offrir une zone de texte enrichie dans son application. Elle se base sur *JavaFX* (la bibliothèque d'interface graphique pour Java). Jusqu'ici, Zest Writer utilisait cette bibliothèque pour offrir des choses sympa comme l'affichage des numéros de lignes dans la zone de rédaction, ou encore la sélection multiple.

Vu comme ça on pourrait penser que cette bibliothèque est idéale, mais non. Elle avait un gros défaut (qui est resté non corrigé depuis maintenant quatre ans): il était impossible de taper des caractères circonflexes ou tréma. Après avoir attendu (en espérant que le problème se corrige), j'ai fini par baisser les bras et revenir à une zone de texte simple mais fonctionnelle.

3.3. De Java 8 à Java 11

C'est peut-être cette migration qui m'a le plus coûté. La raison est simple, depuis Java 9, le projet Jigsaw qui permet de rendre une application Java modulaire a changé beaucoup de choses dans la gestion des dépendances.

Il y a trois ans, la grosse majorité des dépendances de Zest Writer n'était pas compatible avec `Java > 8`, il a donc fallu attendre que ce soit le cas (à la bonne discrétion des mainteneurs donc). Jusqu'à aujourd'hui il y a certaines dépendances qui n'ont pas encore passé le cap (oui langagetools je parle bien de toi). À un moment il a fallu acter le divorce et passer à autre chose.

Si la nouvelle version à mise autant de temps à arriver (en dehors de ma motivation variable), c'est en partie à cause de cette migration.

> *i* Avec le recul, si j'avais su que ça prendrait autant de temps pour que les dépendances migrent, j'aurai certainement sorti une version intermédiaire uniquement compatible Java 8.

Cette migration a permis de faire le tri dans les dépendances qui sont devenues inutiles comme toutes les bibliothèques d'apache client http et okhttp. Maintenant que dans le JDK 11 on a ce qu'il faut comme client http, il y a beaucoup de dépendances qui n'étaient plus justifiables.

3.4. De JavaFX 8 à JavaFX 13

JavaFX est la bibliothèque moderne recommandée pour faire des interfaces pour des applications Java. Zest Writer l'utilise toujours, car elle continue de répondre au besoin.

Pour être compatible Java 11, il fallait migrer de version de Java FX, et (c'est la que ça se corse) changer de plugin de déploiement de l'application.

Zest Writer utilise gradle pour construire ses binaires pour toutes les plateformes. Pour les applications JavaFX, il existe des plugins qui permettent de packager chacun des binaires (deb, rpm, dmg, msi). Le plugin utilisé pour Java FX 8 n'étant plus compatible, il a fallu changer de plugin de revoir complètement la phase de packaging. C'est ainsi que nous avons découvert l'excellent Badass JLink Plugin qui fait le café et dont les développeurs sont très réactifs.

> *i* En gros, beaucoup de migrations de technologies, c'était osé, surtout que le gain pour l'utilisateur ne reflète pas finalement pas le temps de travail passé en coulisse (mais je suppose que c'est ça aussi la vie du développeur).

Je ne peux terminer ce billet sans remercier @Arius qui m'a motivé à travailler d'arrache-pied sur cette version pour la mettre à disposition de la communauté. Je vais peut-être enfoncer une porte ouverte, mais il n'y a rien de plus motivant que les retours/attentes d'utilisateurs.

Merci aussi à tout ceux qui ont pris le temps de reporter les bugs afin qu'ils soient corrigés.

4. Liens et ressources

— Zest Writer, Qu'est-ce que c'est?
— Toutes les manières de contribuer à Zest Writer
— Le guide du développeur de Zest Writer

Merci à ceux qui ont pris la peine de me lire jusqu'ici.

Comme à chaque fois, je ne peux que vous souhaiter bonne rédaction!