Création d'un fichier deb pour un projet et déploiement sur APT

Retour à toute les documentations

Règles

- "`Saisie utilisateur`"
- '`Elément cliquable/sélectionnable`'
- `Nom de fichier, dossier ou autre`
- <Élément à remplacer>

> lien, raccourci clavier et phrase de demande de saisie

```
commande, extrait code et extrait de fichier
```

Table des matières

Préparation du projet

Vous devez tous d'abord créer un fichier deb pour votre projet. Pour cela vous devez suivre les étapes de création d'un fichier deb.

Création d'un fichier deb

- Tutoriel source

  > https://www.iodigital.com/nl/history/intracto/creating-debianubuntu-deb-packages

- Créer un dossier pour le projet

```
mkdir <nom_du_projet>
```

- Créer un dossier `DEBIAN` dans le dossier du projet

```
mkdir <nom_du_projet>/DEBIAN
```

- Créer un fichier `control` dans le dossier `DEBIAN`

```
touch <nom_du_projet>/DEBIAN/control
```

- Ouvrir le fichier `control` avec un éditeur de texte

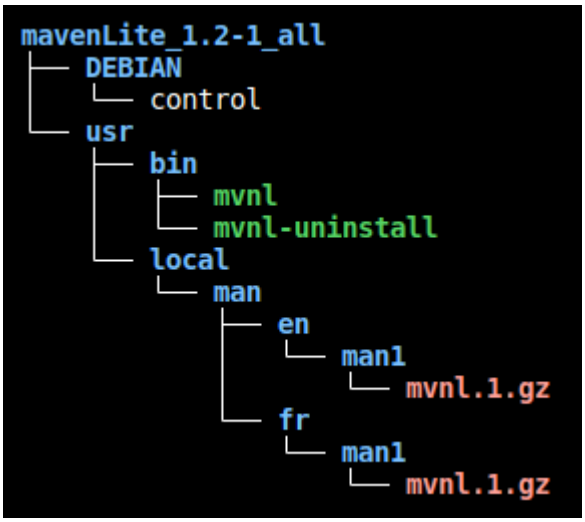
```
open <nom_du_projet>/DEBIAN/control
```

- Ajouter les informations suivantes dans le fichier `control`

```
Package: <nom_du_projet>
Version: <version_du_projet>
Architecture: all <(ou amd64, arm64, i386, ...)>
Depends: <dépendance_1>, <dépendance_2>, ...
Maintainer: <nom_de_l\'auteur>
Description: <description_du_projet>
```

- Copier les fichiers du projet dans le dossier du projet.
  - Veillez à bien copier les fichiers dans le dossier du projet et non dans le dossier `DEBIAN`
  - Créer les dossiers nécessaires dans le dossier du projet, par exemple si vous voulez créer une commande `mvnl` il faut que le fichier de commande est pour nom `mvnl` et qu'il soit dans le dossier `/usr/bin/`. Donc il faut créer le dossier `usr/bin/` dans le dossier du projet et y mettre le fichier `mvnl`. Ceci est du au fait que quand le fichier deb sera installer, les fichiers seront copier dans le système de fichier de l'ordinateur en respectant l'arborescence du dossier du projet.

- Exemple d'arborescence de dossier du projet



- Créer le fichier deb

```
dpkg-deb --build <nom_du_projet>
```

Déploiement du projet

Création de pages de manuel - Linux

- Tutoriel source

> https://www.cyberciti.biz/faq/linux-unix-creating-a-manpage/

- Installer le paquet `txt2man`
- Créer un fichier texte ou markdown avec les informations de la page de manuel
    - Exemple de fichier markdown avec Maven Lite
        - Les titres de niveau 1 sont obligatoires. Ce sont des conventions des pages de manuel.

```
---
date: Janvier 2024
section: 1
title: Maven Lite
---

# NOM

mvnl - Gestionnaire de projet Java simple et léger inspiré du gestionnaire de
projet `Maven`.

# SYNOPSIS

mvnl \[OPTION\]\... \[ARGUMENT\]\...

# DESCRIPTION

description du programme

# EXEMPLES

-   `mvnl -s src -o bin -c -e UTF-8` \--\> compile le projet Java avec
    l\'encodage UTF-8 qui se trouve dans le dossier src et met les
    fichiers compilés dans le dossier bin.

# COMPORTEMENT PAR DÉFAUT
```

```
Par défaut, si aucune option n\'est spécifiée, la commande `mvnl`
affiche la page d\'aide qui est affichée avec l\'option `-h` ou
`--help`. Cette page d\'aide est différente et plus simple que la page
de manuel qui est affichée avec la commande `man mvnl`.

# OPTIONS

## Toutes les options

-v , \--version Affiche la version et quitter.

-h , \--help Affiche l\'aide et quitter.

## Les options obligatoires pour la compilation sont :

-s , \--source Dossier racine du projet à compiler.

-o , \--output Dossier de sortie des fichiers compilés.

-c , \--compilation Compile le projet.

## Les options obligatoires pour le lancement sont :

-m , \--main Classe principale à lancer.

-l , \--launch Lance le projet.

-cp , \--classpath Voir l\'option -cp dans la liste des options
ci-dessus.

## Les options obligatoires pour la compilation et le lancement sont :

-s , \--source Dossier racine du projet à compiler.

-o , \--output Dossier de sortie des fichiers compilés.

-m , \--main Classe principale à lancer.

-cl , \--compile-launch Compile et lance le projet. (équivalent à -c -l)

# CODES DE RETOUR

0 : Tout s\'est bien passé.

1 : Une erreur est survenue.

# FICHIERS

Maven Lite est constitué uniquement de 3 fichiers.

-   `'mvnl'`, le fichier principal qui se situe dans le dossier
    \'/usr/bin/\'.

-   `'mvnl.1.gz'`, le fichier d\'aide contenant la page de manuel française
    affichée avec la commande `man mvnl` qui se situe dans le dossier
    `'/usr/local/man/fr/man1/'`.

-   `'mvnl.1.gz'`, le fichier d\'aide contenant la page de manuel anglaise
    affichée avec la commande `man mvnl` qui se situe dans le dossier
    `'/usr/local/man/en/man1/'`.

# BOGUES

Bogues connu

# AUTEUR

Écrit par Robart Floris.
```

```
# RAPPORT DE BOGUES

Reporter les bogues par mail à l\'adresse \<email@gmail.com\>
```

- Convertissez votre fichier en fichier Roff

```
txt2man -t <nom_du_fichier> > <nom_du_fichier>.1
```

```
pandoc --from markdown --to roff <nom_fichier_source> -o <nom_fichier_destination>.1
```

- Il peut être nécessaire d'ajouter la ligne suivante au début votre fichier `.1`

```
.TH "<Nom>" "<section>" "<Date>" "<Nom> Version <version format X.X.X>" "<type de
l'application>"
```

  - Exemple avec Maven Lite

```
.TH "Maven Lite" "1" "Janvier 2024" "Maven Lite Version 1.2.0" "Commandes
Utilisateur"
```

- Convertissez votre fichier `Roff` en fichier conpresser `.1.gz`.

  - Attention, le nom du fichier `.1.gz` doit obligatoirement être le nom de la commande que vous avez créer. Dans mon cas le nom de la commande est `mvnl` donc le nom du fichier `.1.gz` doit être `mvnl.1.gz`

```
gzip <nom_du_fichier>.1 > <nom_du_fichier>.1.gz
```

- Pour que la commande `man` fonctionne il faut placer le fichier `.1.gz` dans le dossier `/usr/local/man/<langue>/man1/`.

```
sudo cp mvnl.1.gz /usr/local/man/fr/man1/mvnl.1.gz
```

Retour à toute les documentations