

knife-cookbook-doc

Table of Contents

- [knife-cookbook-doc](#)
 - [Présentation](#)
 - [Règles de confection des documentations](#)
 - [Documentation du fichier metadata.rb](#)
 - [Documentation des recettes](#)
 - [Documentation des attributs](#)
 - [Documentation des ressources](#)
 - [Documentation des définitions](#)
 - [Autres](#)
- [Installation](#)
- [Utilisation](#)
 - [EN ligne de commande](#)
 - [Automatiquement depuis rakefile](#)

Présentation

knife-cookbook-doc est un plugin de knife pour aider à créer et maintenir un fichier README.md pour un cookbook. Dans la mesure du possible, le plugin utilise les mêmes métadonnées que celles utilisées par chef lors de la génération de la documentation. Le plugin va également scanner les fichiers sources pour les annotations présentes dans les commentaires. Les utilisateurs peuvent également ajouter des fragments de démarques dans le répertoire doc / pour fusionner dans le fichier README.md généré.

L'objectif est de garder le code comme source d'information faisant autorité. L'espoir est que garder la documentation proche du code aidera à maintenir sa devise.

Règles de confection des documentations

D'une façon générale chaque élément d'un cookbook peut contenir de la documentation

Documentation du fichier metadata.rb

Une attention particulière devrait être accordée à la documentation de la compatibilité de la plateforme et des exigences du livre de recettes (c'est-à-dire dépendances, recommandations, suggestions, etc.) lors de l'écriture du fichier metadata.rb du livre de recettes. La documentation pour les attributs et les recettes peut être dérivée de l'analyse du code source ou d'une déclaration explicite dans metadata.rb.

Documentation des recettes

En haut de chaque recette, ajouter une section de documentation détaillée telle que;

```
=begin
#<
La recette est géniale. Il fait chose 1, chose 2 et chose 3!
#>
=end
```

Si l'utilisateur n'a pas spécifié de documentation pour la recette dans metadata.rb, l'outil prend la première phrase de la documentation détaillée comme résumé à utiliser dans la table des matières.

Documentation des attributs

Si l'utilisateur n'a pas spécifié de documentation pour les attributs dans metadata.rb, l'outil analyse les fichiers d'attributs et utilise le bloc de documentation précédant l'attribut.

```
#<> MyApp Admin Group: groupe autorisé à gérer MyApp.
default['myapp']['group'] = 'myapp-admin'
```

Documentation des ressources

Dans chaque ressource, ajouter une documentation détaillée telle que;

```
=begin
#<
Cela crée et détruit le service mon_service

@action create Crée le service mon_service.
@action destroy Détruit le service mon_service.

@section Exemples

  # Un exemple de service mon_service
  mycookbook_mon_service "mon_service"
    port 80
  end
#>
=end
...
#<> @attribute port Port sur lequel le service HTTP va se lier.
attribute :port, :kind_of => Integer, :default => 8080
```

Il convient de noter que la documentation du LWRP exige que l'utilisateur documente les actions, en utilisant @action <action> <description> et les attributs en utilisant @attribute <attribute> <description>. Cela permet des descriptions significatives pour les actions et les attributs à ajouter

au fichier README.

L'autre texte sera ajouté au début de la documentation LWRP sauf s'il est marqué avec @section <heading>, auquel cas il sera ajouté à la fin de la documentation LWRP.

À partir de Chef> 12.5, les propriétés remplaceront les attributs dans les ressources personnalisées. Pour la compatibilité, @property peut être utilisé de manière interchangeable avec @attribute. Les deux sont traitées complètement identiques par le plugin.

Documentation des définitions

Dans chaque définition, ajouter une documentation comme:

```
=begin
#<
Cette définition fait quelque chose d'utile.

User @param à exécuter en tant que [Utilisateur]
Group @param à exécuter en tant que [Groupe d'utilisateurs]

@section Exemples

  # Un exemple de définition
  mycookbook_definition "ressources de l'application"
    user 'appuser'
    group 'appgroup'
  end
#>
=end
```

Autres

Enfin, l'utilisateur peut ajouter des fragments de documentation dans le doc/. Plus important encore, ajouter doc/overview.md qui remplacera la première section Description du readme. Ajouter un document doc/credit.md qui remplacera la dernière section «License and Maintainer» dans le readme. Sous la section Exigences du fichier readme, la plateforme et les livres de recettes sont produits à partir de metadata.rb mais si on doit placer autre chose comme environnement ou databags, il faut créer doc/requirements.md qui ajoutera le contenu à la section Requirements. Les fragments restants seront inclus à la fin du readme dans l'ordre lexicographique du nom de fichier.

Installation

Vous pouvez installer le plugin via RubyGems:

```
gem install knife-cookbook-doc
```

Utilisation

EN ligne de commande

```
knife cookbook doc COOKBOOK_DIR (options)
```

- **-o, -output-file FILE**: Définit le fichier de sortie à rendre par rapport au répertoire du livre de recettes. Par défaut à README.md
- **-t, -template FILE**: Définit le fichier de modèle utilisé pour rendre README.md

Exemples

```
knife cookbook doc path/to/cookbook
knife cookbook doc path/to/cookbook --template README.md.erb
```

Automatiquement depuis rakefile

Alternativement, on peut exécuter cookbook doc directement à partir de Rakefile:

```
require 'knife_cookbook_doc/rake_task'

# With default options
KnifeCookbookDoc::RakeTask.new(:doc)

# Example with custom options
KnifeCookbookDoc::RakeTask.new(:doc) do |t|
  t.options[:cookbook_dir] = './'
  t.options[:constraints] = true
  t.options[:output_file] = 'README.md'
  t.options[:template_file] =
    "#{File.dirname(__FILE__)}/templates/README.md.erb"
end
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=vagrant:knife-cookbook-doc>

Last update: **2025/02/19 10:59**

