CHOUETTE

Maintenance, accompagnement et recette de logiciels pour les échanges de données multimodales

application Chouette V2

-

Utilisation depuis une console Rails

Auteurs :	Michel ETIENNE, Luc DONNET, Marc Florisson (CityWay)
Relecteurs	Patrick GENDRE (CEREMA), Jean SENG (AFIMB)

Résumé :

CHOUETTE est un logiciel libre développé à l'initiative du ministère français chargé des transports (et du développement durable) dans le but de faciliter l'échange de données d'offre (théorique) de transport collectif (TC), en s'appuyant pour cela sur la norme NFP 99506, dite Neptune, qui spécifie un profil d'échange XML.

Les utilisateurs visés sont les collectivités locales Autorités Organisatrices de Transport (AOT), les exploitants des réseaux TC, ainsi que leurs prestataires (bureaux d'étude ou société de services) et d'autres acteurs tels que services de l'état, éditeurs de logiciels, opérateurs de services d'information, chercheurs...

L'application CHOUETTE est disponible sous la forme d'une application WEB destinée à être déployée sur un serveur WEB, et d'une application en ligne de commande.

Ce document présente l'utilisation de Chouette2 depuis une console Ruby on Rails.

Agence française pour l'information multimodale et la billettique

Informations sur ce document :

<table>
<tr><td colspan="4">Organisme commanditaire : AFIMB</td></tr>
<tr><td colspan="4">Titre : Guide d'utilisation de Chouette v2 depuis une console Ruby on Rails</td></tr>
<tr><td colspan="4">Sous-titre :</td></tr>
<tr><td colspan="2">Organismes auteurs
CITYWAY
CEREMA DT Med.</td><td>Rédacteurs
Marc FLORISSON
Zakaria BOUZIANE
Michel ETIENNE
Luc DONNET</td><td>Participants
Patrick GENDRE
Jean SENG</td></tr>
<tr><td colspan="2">Maitre d'ouvrage
AFIMB</td><td colspan="2"></td></tr>
<tr><td colspan="2">Mots clés :
profil d'échange Neptune, information multi-modale, application Chouette, manuel d'installation, postgreSQL, Ruby, Rails</td><td colspan="2">Diffusion :
 
publique (licence Creative Commons CC-by-nd)</td></tr>
<tr><td>Nombre de pages :
6 pages</td><td>Date
Octobre 2014</td><td>Confidentialité :
Non</td><td>Bibliographie :
Oui</td></tr>
</table>

Historique des versions / révisions :

Version	Date d'application	Description des changements	Auteur
2.5	31/10/2014	Création à partir de l'annexe du doc d'installation (qui datait de 2012 v2.0.2)	Michel Etienne Patrick Gendre

1 UTILISATION DE LA CONSOLE RAILS

Le framework RubyOnRails fournit un certain nombre de facilités en mode commande: http://guides.rubyonrails.org/command_line.html#rails-console

Il est intéressant d'en tirer parti pour manipuler de manière souple les données métier gérées par Chouette.

Pour des détails sur la procédure de mise en place de chouette2 sous Rails, reportez-vous au guide d'installation.

Par défaut, l'environnement (au sens des applications RubyOnRails) de la console est "development". Dans ce cas, la console utilise les paramètres de connexion associés à « `development` » dans le fichier `config/database.yml`

L'environnement peut être précisé en option. Par exemple, la commande ci-dessous permet de se connecter en environnement de « production » :

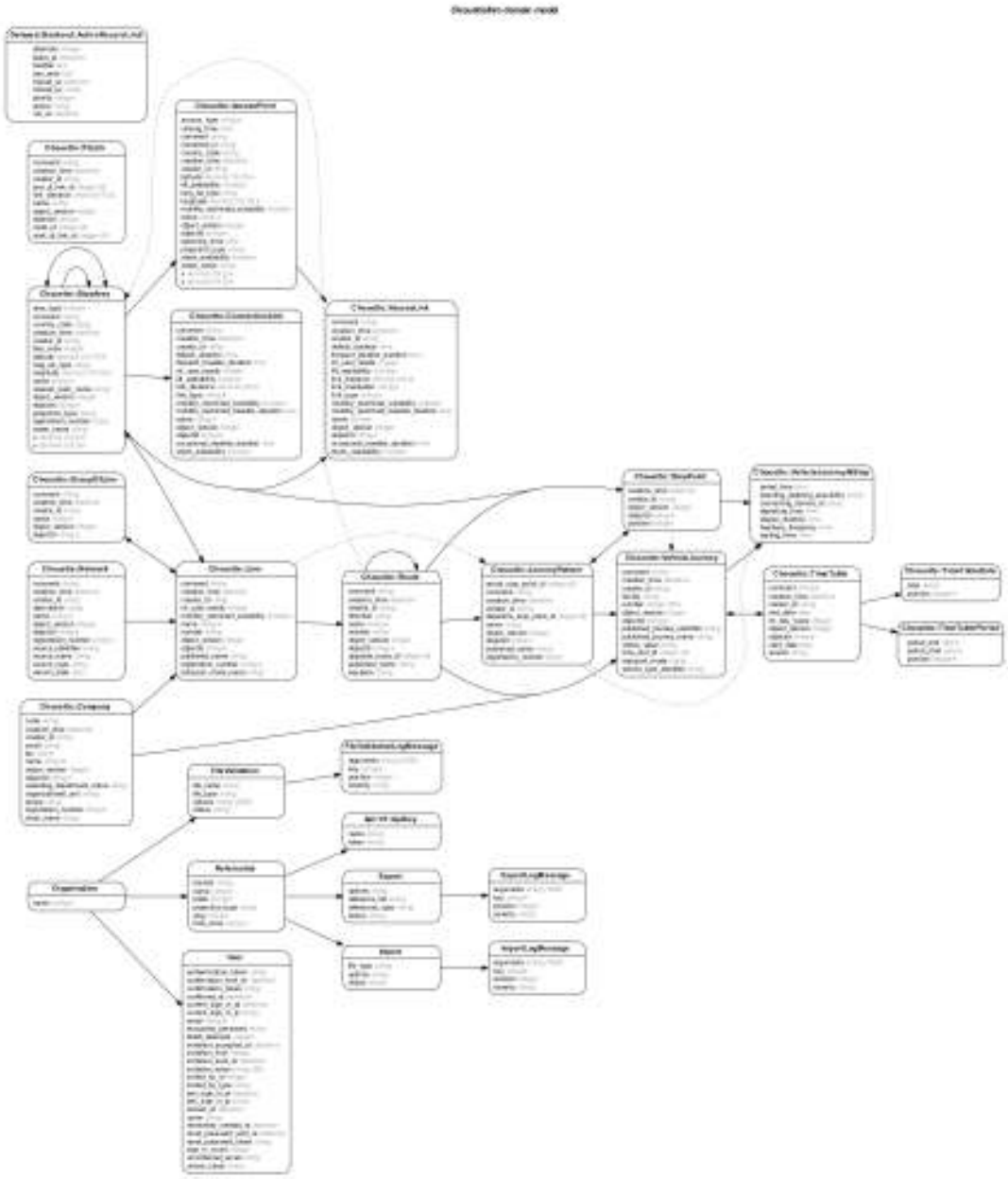
```
bundle exec rails c production
```

Dans la suite de cette annexe, quelques scripts sont détaillés de manière à illustrer quelques cas d'utilisation de la console. Ces scripts ruby peuvent être exécutés dès l'ouverture de la console par copier/coller de chaque ligne de commande.

1.1. Présentation du modèle de données en persistance

Le schéma présenté dans le document « Modele-classes.pdf » introduit les modèles (au sens d'une application Rails) qui sont gérés en persistance.

Le schéma précise les relations entre les modèles (flèches) ainsi que les propriétés de chacun des modèles. Ce schéma est présenté ci-après en petite taille.



1.2. Exemples

Affichage des arrêts commerciaux d'une séquence d'arrêts

```
# Lister les référentiels disponibles

Referential.all.map(&:slug)


# Se placer dans un référentiel de la base

# en supposant que la liste ci-dessus contienne "tatrobus"

Referential.find_by_slug("tatrobus").switch
```

```
# Lister les noms des lignes du réseau
Chouette::Line.all.map(&:name)

# Sélectionner une ligne par son nom
# en supposant que la liste ci-dessus contienne "Ligne 1 Bleue"
line = Chouette::Line.find_by_name( "Ligne 1 Bleue")

# Sélectionner la première séquence d'arrêts
route = line.routes.first

# Lister les noms des arrêts de la séquence
route.stop_areas.map(&:name)

# Lister les noms des arrêts commerciaux relatifs aux arrêts de la séquence
route.stop_areas.map(&:parent).map(&:name)
```

Affichage des horaires de départ d'une séquence d'arrêts

```
# Lister les référentiels disponibles
Referential.all.map(&:slug)

# Se placer dans un référentiel de la base
# en supposant que la liste ci-dessus contienne "tatrobus"
Referential.find_by_slug("tatrobus").switch

# Sélectionner la première séquence d'arrêts
route = Chouette::Line.find_by_name( "Ligne 1 Bleue").routes.first

# Lister les horaires de départ par ordre croissant
route.vehicle_journeys.map {|v| v.vehicle_journey_at_stops.first.departure_time}.sort
```

Consultation des liens d'accès d'un arrêt

```
# Lister les référentiels disponibles
Referential.all.map(&:slug)
```

```
# Se placer dans un référentiel de la base
# en supposant que la liste ci-dessus contienne "tatrobus"
Referential.find_by_slug("tatrobus").switch

access_link = Chouette::AccessLink.all.first

# afficher l'arrêt
access_link.stop_area

# afficher l'accès
access_link.access_point
```

Vérification d'une date pour un calendrier d'application

```
# Lister les référentiels disponibles
Referential.all.map(&:slug)

# Se placer dans un référentiel de la base
# en supposant que la liste ci-dessus contienne "tatrobus"
Referential.find_by_slug("tatrobus").switch

# Sélectionner un calendrier
tm = Chouette::TimeTable.all.first

# Vérifier si la date du jour est comprise dans le calendrier
tm.include_day?( Date.today)
```