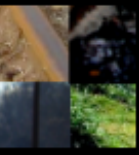




Ruby on Rails

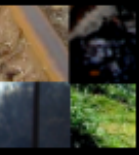
Octobre 2007, Julien Landuré



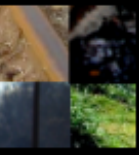


Sommaire

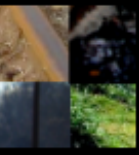
- Introduction
- Présentation du framework
- Technique
 - ActiveRecord
 - ActionController
 - ActionView
 - Tests
- Conclusion
- Ressources et liens
- Démonstration



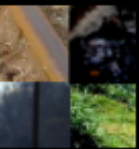
- Retour en force de la plateforme web : ajax, web 2.0
- Avènements de nouveaux outils et de frameworks web
- But commun
 - rendre les développeurs opérationnels
 - ne pas redévelopper sans cesse les fonctionnalités similaires
 - imposer des normes de codage et des bonnes pratiques



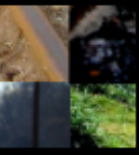
- Créé par **37signals** & David Heinemeier Hansson (aka **DHH**)
- Framework **extrait** de l'application "**Basecamp**"
- Version 1.0 sorti en décembre 2005, version 1.2 sortie le 14 mars 2007... version 2.0 en cours de développement



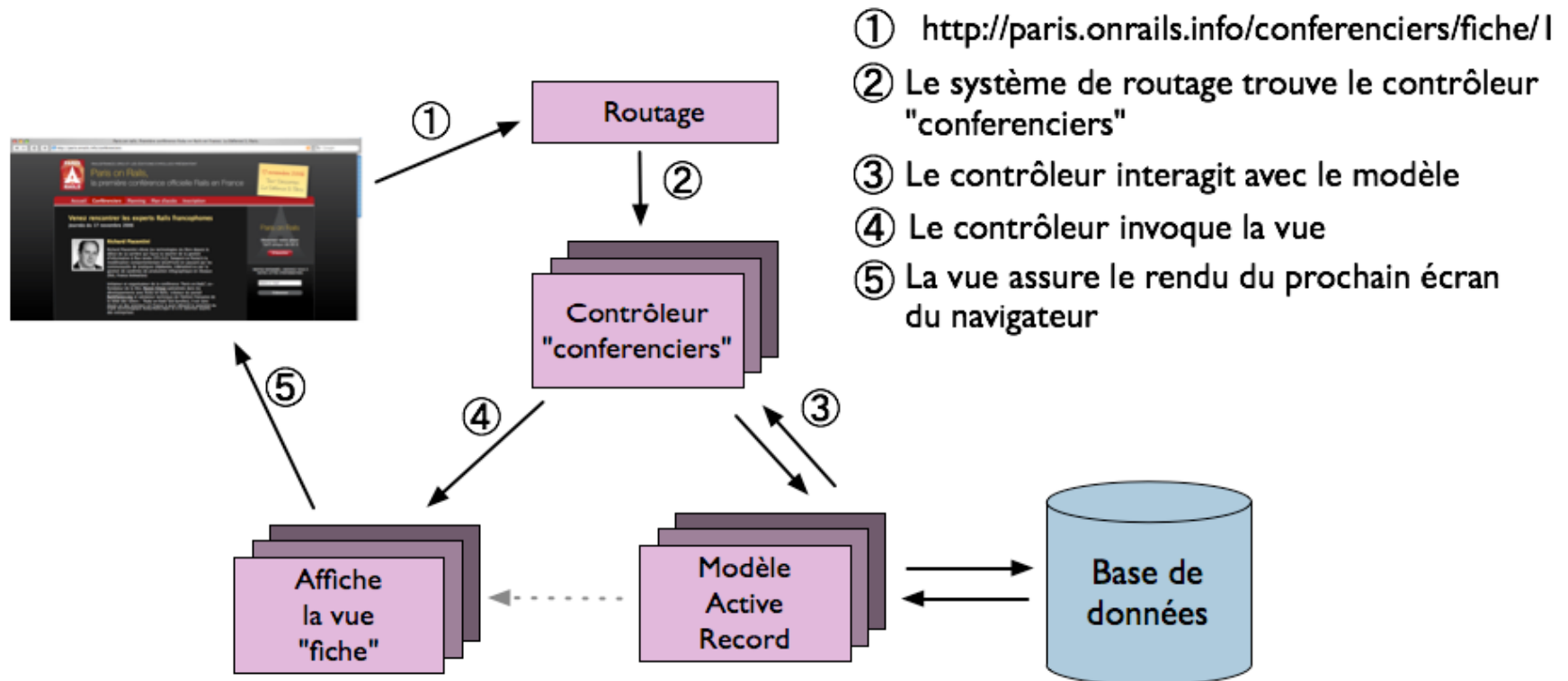
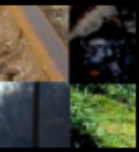
- Framework web
 - Socle technique pour l'interaction de pages
- Design Pattern: MVC (Model View Controller), ORM (Object Relational Mapping)
- "Don't Repeat Yourself" (DRY)
- Ensemble de composants: Active Record, Action Pack, Action Mailer, Action Web Service, Prototype/Scriptaculous
- Populaire
 - 1,300,000 téléchargements
 - Nombreuses manifestations : ParisOnRails etc.

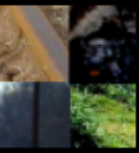


- Basé sur le langage Ruby
 - langage de script (interprété)
 - multiplateforme
 - orienté objet
 - dynamique
- Exemple de codes
 - `5.times { print "Odelay!" }`
 - `exit unless "restaurant".include? "aura"`
 - `['toast', 'cheese', 'wine'].each { |food| print food.capitalize }`



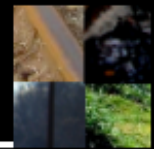
- **Convention** plutôt que **Configuration**
 - Adoption de règles d'usage **implicites**
 - Ne configurer que les **cas particuliers**
 - Gain de **productivité** pour le **développeur** !
- Intégration des tests unitaires et fonctionnels
- Intégration d'**AJAX** (prototype & script.aculo.us)



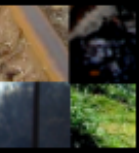


- **Mappage** Objet-Relationnel
- Gestion des **relations** entre les tables
- Plus de SQL à écrire dans la **grande majorité** des cas
- Fonctionne avec de **nombreux** SGBD
 - (MySQL, PostgreSQL, Oracle, ...)
- Support des relations, des callbacks (before_save etc.)

- Classiquement :
 - Les **tables** de la base de données correspondent aux **classes** Ruby
 - Les **enregistrements** aux **objets**
 - Les **colonnes** aux **attributs (propriétés)** des objets



- Exemple de conventions
 - **id** comme nom de **clef primaire**
 - Nom des **tables** au **pluriel** et en **minuscules**, tiret bas entre les mots
 - Nom des **classes** au **singulier** et en **casse mixte**
 - Les fichiers de modèle au **singulier**, en **minuscules**, tiret bas entre les mots, stockés dans **app/models**
- Exemple
 - Nom de la table : commande_clients
 - Fichier modèle : app/models/commande_client.rb
 - Classe Ruby : CommandeClient



- Ceci est une classe de modèle “conventionnelle”

```
class CommandeClient < ActiveRecord::Base  
end
```

- Et elle permet de :

- créer
- supprimer
- éditer
- rechercher (paginer)

```
commandes = CommandeClient.find :all,  
:offset => 10, :limit => 5
```

commande_clients
id
prix
numero
date_achat

- Finders dynamiques

```
CommandeClient.find_by_prix_and_numero(100, 'CC-01')
```

Technique

La couche ORM

■ Les validateurs

- Liés au modèle :
 - **incontournables !**

- Garants de **l'intégrité** des données

```
validates_presence_of :nom, :prenom
```

```
validates_size_of :nom, :prenom, :within => 3..15
```

```
validates_uniqueness_of :nom
```

```
validates_numericality_of :age
```

```
validates_format_of :ville, :with => /^\\w+$/i,
```

```
  :message => "can only contain letters and numbers"
```

```
validates_inclusion_of :commentaire, :in => %w(1 2 3)
```

■ Affichage des **erreurs** en **une ligne** dans la vue

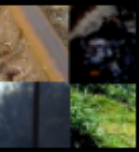
```
<%= error_messages_for :lecteur %>
```

3 errors prohibited this user from being saved

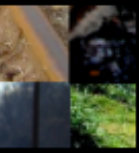
There were problems with the following fields:

- Ville can only contain letters and numbers
- Nom is too short (minimum is 3 characters)
- Nom can't be blank

Nom

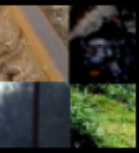


- Et bien plus...
 - **Isolation** des environnements
 - Développement
 - Test
 - Production
 - Fichier de migrations : définition **portable**, **indépendante** du SGBDR
 - Schéma en Ruby : db/schema.rb
 - Gestion d'observateurs
 - Gestion des transactions



- **Coordonne les interactions** entre les vues et le modèle
 - Une **méthode publique** = une **action**
 - Gestion du **routing**
 - Gestion des **sessions**
 - Gestion de la **pagination**
 - Gestion des **caches** (page, action, fragment)
 - **Pré-filtres**, post-filtres et péri-filtres

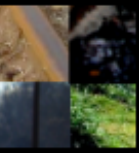
```
URL: /lecteurs/bienvenue?nom=piacentini
class LecteursController < ApplicationController
  def bienvenue
    @lecteur = Lecteur.find_by_nom params[:nom]
    session[:lecteur_id] = @lecteur.id
    flash[:notice] = "hé, déjà #{@lecteur.age} ans !"
  end
end
```



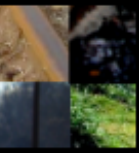
- Gestion du routage

- `http://mon_site/articles/2006/08/31`

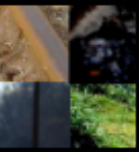
```
map.articles 'articles/:year/:month/:day',  
  :controller => 'articles', :action => 'find_by_date',  
  :year => /\d{4}/, :month => /\d{1,2}/, :day =>  
    /\d{1,2}/
```



- Différents templates
 - ERB (.rhtml),
 - XML(.rxml)
 - RJS(.rjs)
- Présence de **Layouts** (gabarits) : **factorisation** de la mise en page
- **Fichiers partiels** (Partial): **factorisation** du rendu HTML
- **Helpers**: Ils vous simplifient la vue

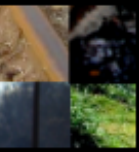


- Templates
 - ERB « Embedded Ruby »
 - HTML classique avec du code ruby
 - Les variables d'instance du contrôleur sont disponibles dans la vue
 - RXML
 - Utilise la bibliothèque **Builder** pour construire des réponses au format **XML**
 - RJS
 - Permet de générer du Javascript...sans écrire une ligne de Javascript
 - C'était **LA** grande nouveauté de Rails 1.1
 - Combiné à **Prototype** et **Script.aculo.us**

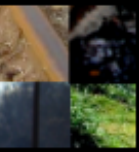


- Helpers
 - Modules contenant des **méthodes utilisables** par les différentes **vues** d'un contrôleur

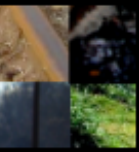
```
ActionView::Helpers::ActiveRecordHelper  
ActionView::Helpers::AssetTagHelper  
ActionView::Helpers::BenchmarkHelper  
ActionView::Helpers::CacheHelper  
ActionView::Helpers::CaptureHelper  
ActionView::Helpers::DateHelper  
ActionView::Helpers::DebugHelper  
ActionView::Helpers::FormHelper  
ActionView::Helpers::FormOptionsHelper  
ActionView::Helpers::FormTagHelper  
ActionView::Helpers::JavaScriptHelper  
ActionView::Helpers::JavaScriptMacrosHelper  
ActionView::Helpers::NumberHelper  
ActionView::Helpers::PaginationHelper  
ActionView::Helpers::PrototypeHelper  
ActionView::Helpers::PrototypeHelper::JavaScriptGenerator::GeneratorMethods  
ActionView::Helpers::ScriptaculousHelper  
ActionView::Helpers::TagHelper  
ActionView::Helpers::TextHelper  
ActionView::Helpers::UrlHelper
```



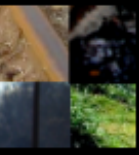
- Rails met à votre disposition **3 types** de tests:
 - Tests **unitaires**
 - Tests **fonctionnels**
 - Tests **d'intégration**
- **Génération automatique** des squelettes
 - Plus aucune raison de ne pas suivre les bonnes pratiques ;-)
 - Fixtures : fichier contenant des données pour les tests
test/fixtures/articles.yml
one:
 id: 1
 title : article 1
two:
 id: 2
 title : article 2



- Tests fonctionnels
 - valider les méthodes d'un contrôleur
- Tests d'intégration
 - simuler une utilisation réelle de l'application
 - « Selenium On Rails »
- Tests sur changements récents
rake test:recent
- Tests sur changements non committés
rake test:uncommitted

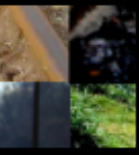


- **ActionMailer et ActionWebservice**
- Les **plugins** (des centaines sont disponibles !)
- **AJAX avec Prototype & Script.aculo.us**
- Le **déploiement** avec **Capistrano**
- **L'intégration continue** avec **CIA**
- Et bien d'autres choses...



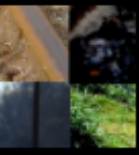
Conclusion

- Courbe d'apprentissage aisée
- Concentration sur le code vraiment utile (la valeur ajoutée)
- Gain de **productivité** sur les tâches de base
- **Réutilisation** du code "générique" : DRY



Ressources et liens

- <http://www.rubyonrails.com/>
- http://en.wikipedia.org/wiki/Ruby_on_Rails
- <http://www.railslodge.com/> - Plugins
- <http://www.oreilly.com/catalog/rubyrails/>
- <https://crite.emn.fr/video/conferences/rubyonrails/> - EMN



- Merci de votre attention... des questions ?
- Démonstration