

Internationalisation des applications Rails



Yann Lugrin

CTO @ Liquid Concept

Membre de l'équipe de développement de Globalize

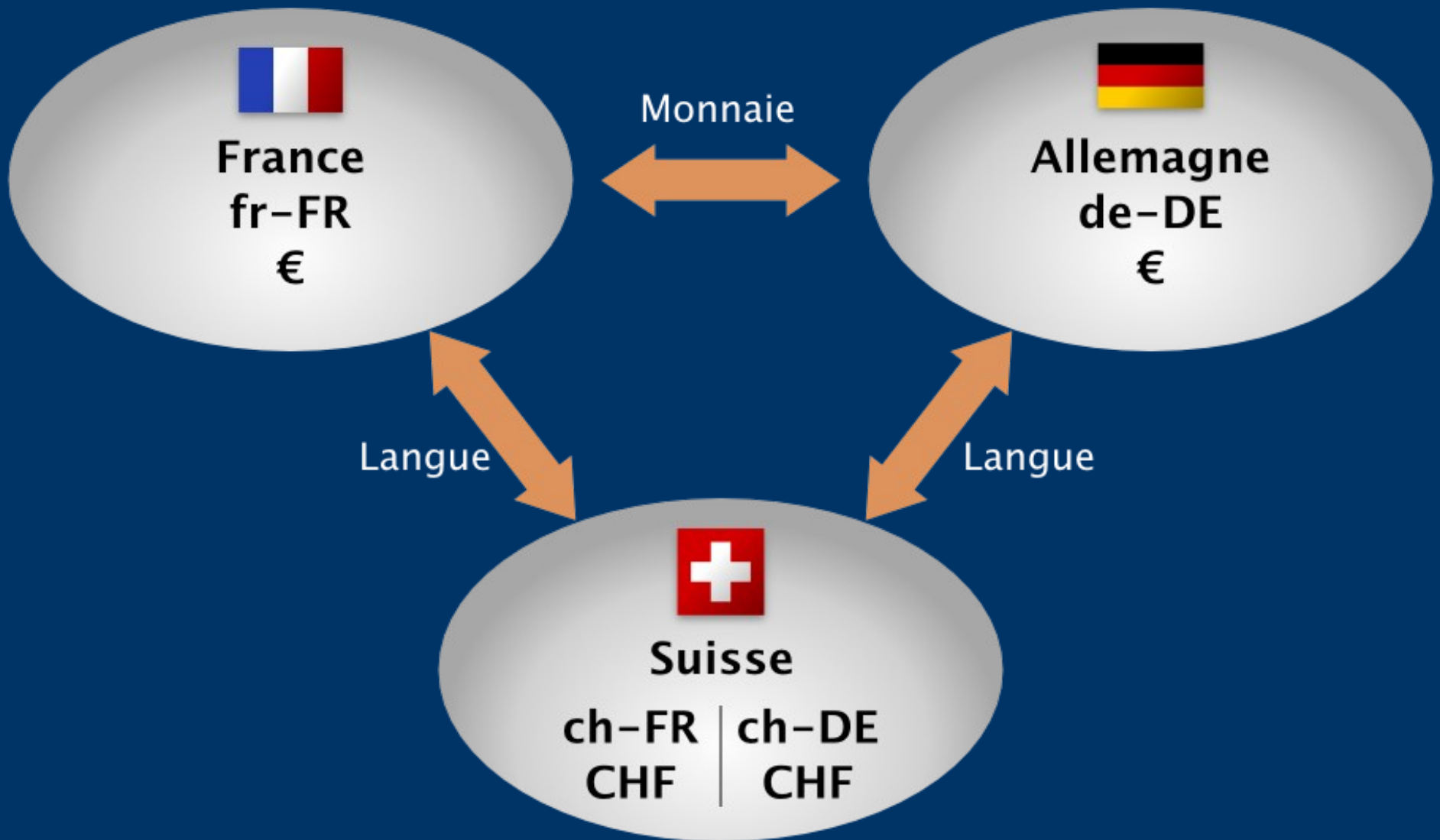
Paris on Rails – 10 décembre 2007

Internationalisation

- Définition
- Les besoins
- Les problèmes
- Les solutions
- Résumé

Internationaliser ?

- Adapter l'interface / Supporter les données
 - Différences régionales
 - Différences linguistiques



Quels besoins ?

- Interface dans plusieurs langues
- Contenu ou documents différents selon les régions
- Contenu ou documents dans plusieurs langues
- Révision de l'interface

Interface ?

- Mise en forme
- Grammaire
- Contenu « statique » de la page
- Nombres / Dates
- Monnaie / Devises
- Validation / Messages

Mise en forme

- Longueur du texte
- Orientation du texte

Grammaire

- Genre
- Nombre

Contenu « statique »

- Textes
- Images
- Documents à télécharger
- Liens

Nombres / Dates

- Nombres :
 - en français : 128 000,50 ou 128000,50
 - en anglais : 128,000.50 ou 128000.50

- Dates (10 décembre 2007)
 - en français : 10 décembre 2007 ou 10/12/07
 - en anglais : December 10, 2007 ou 12/10/07

Monnaie / Devises

- Euro :
 - en Français : 1 234,00 EUR
 - en Anglais : 1,234.00 EUR
- Dollars :
 - en Français : \$ 1 234,00
 - en Anglais : \$ 1,234.00
- Le format des nombres dépend de la langue, mais pas la devise

Validations / Messages

- Adapter les conditions de validation
 - NPA différent selon le pays
 - Format de la date différent selon la langue ou la région
 - ...
- Traduction/Adaptation des messages

Données ?

- Documents servis
- Contenu textuel
- Tout contenu ne doit pas être traduit :
 - nom
 - adresse

Documents

- Un fichier par langue

paris_on_rails_fr.pdf → Français

paris_on_rails_en.pdf → Anglais

- Un fichier multilingue

paris_on_rails.pdf → Français, Anglais

Contenu

- Un enregistrement par langue

/contents/1 → Content.find(1) → Français

/contents/2 → Content.find(2) → Anglais

- Enregistrement en plusieurs langues

/contents/1 → Français
/contents/1 → Anglais

Quels sont les problèmes

- Ruby et Unicode
- Rails et Unicode
- Accès aux données
- Routage
 - Vers les contrôleurs
 - Vers le contenu statique
- Processus de traduction

Quelles sont les solutions ?

- Unicode
 - jcode
 - ActiveSupport::Multibyte::Chars
- Plugins
 - **Globalize**
 - Globalite
 - SimpleLocalization
 - Gibberish

Unicode

- UTF-8 (RFC 2279)
 - Codage de taille variable
 - 128 premiers caractères identiques à Ascii

Ruby et l'Unicode

- jcode (standard library)
- Limitations de jcode
 - Longueur des chaines
 - Casse
 - ...

```
1  
2 $KCODE = 'u'  
3 require 'jcode'  
4
```

- Ces limitations sont levées dans Rails

Rails et l'Unicode

- XHTML : langue & charset
- Base de données
- ActiveSupport::Multibyte::Chars

XHTML : langue & charset

- Le charset par défaut de Rails est UTF-8
- Langue et encodage de la page

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
3 <head>
4   <title>Paris on Rails 2007</title>
5   <link rel="icon" type="image/png" href="/images/favicon.png" />
6
7   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
8   <meta http-equiv="Content-language" content="fr-FR" />
9
```

- Encodage des fichiers en UTF-8

Bases de données et Unicode

- MySQL, PostgreSQL et sqlite 3
- Configuration de la base pour le support Unicode
- Encodage des fixtures en UTF-8

ActiveSupport::Multibyte::Chars

- Encodage du code en UTF-8
- Chars est le support de l'Unicode de Rails

```
1 $KCODE='u'  
2  
3 string = "Café à un accent"  
4 string[0..3] # => "Caf?"  
5  
6 string.chars[0..3] # => "Café"  
7  
8 string.reverse # => "tnecca nu ?? ??faC"  
9  
10 string.chars.reverse # => "tnecca nu à éfaC"  
11 string.chars.upcase # => "CAFÉ À UN ACCENT"  
12 string.chars.reverse.upcase # => "TNECCA NU À ÉFAC"  
13  
14 string.length # => 18  
15 string.chars.length # => 16
```

Globalize

- Mise en place
- Interface
- Données
 - Model – méthode « table globale »
 - Model – méthode « table model »



globalize-rails.org

Mise en place de Globalize

```
$ script/plugin install http://svn.globalize-rails.org/svn/globalize/branches/for-1.2
```

```
$ rake globalize:setup
```

```
./config/initializers/globalize.rb
```

```
1 Globalize::Locale.set_base_language 'fr-FR'
```

```
./app/controllers/application.rb
```

```
1 class ApplicationController < ActionController::Base
2   before_filter :set_locale
3
4   protected
5   def set_locale
6     begin
7       Globalize::Locale.set params[:language]
8     rescue ArgumentError
9       redirect_to params.merge( :locale => 'fr-FR' )
10    end
11  end
12 end
```

Contenu statique

- String#t, String#l

```
1 'ok'.t
2 '_message'.t('default')
3 'Bienvenu, %s, merci de lire vos messages !' / name
4
5 Globalize::ViewTranslation.find(:all,
6   :conditions => [ 'text IS NULL AND language_id = ?', Globalize::Locale.language.id ],
7   :order => 'tr_key')
```

- Globalize::ViewTranslation
 - Model contenant les traductions du contenu statique
 - Permet de faire une interface pour la traduction

Nombres, Dates, Monnaie & Devises

- Format des nombres

`12345678.localize` → `"12.345,78"`

- Format des dates

`Time.now.localize("%d %B %Y")` → `"10 décembre 2007"`

- Devise avec la classe `Currency`

`c = Currency.new(123456.78)`

`c.localize` → `"12.345,78"`

`c.format(:code => true)`

`c.localize` → `"12.345,78 EUR"`

Model – méthode « table globale »

- ActiveRecord::base#translate

./app/models/content.rb

```
1 # Table 'contents'
2 #
3 # Colonne      | type
4 #-----|-----
5 # id           | integer
6 # title        | string
7 # content      | text
8 # author       | text
9 # created_at   | datetime
10 # modified_at  | datetime
11 #
12 class Content < ActiveRecord::Base
13   translates :title, :content
14 end
```

Utilisation d'une table globale créée lors de l'installation de Globalize

rake globalize:setup

title, content → multilingue

Model – méthode « table model »

./app/models/content.rb

```
1 # Table 'contents'
2 #
3 # Colonne      | type
4 #-----|-----
5 # id           | integer
6 # title        | string
7 # title_en     | string
8 # content      | text
9 # content_en   | text
10 # author       | text
11 # created_at  | datetime
12 # modified_at | datetime
13 #
14 class Content < ActiveRecord::Base
15   self.keep_translations_in_model = true
16   translates :title, :content
17 end
```

./config/initializers/globalize.rb

```
1 Globalize::Locale.set_base_language 'fr-FR'
2 Globalize::DbTranslate.keep_translations_in_model = true
```

Routage vers les contrôleurs

- Ressources
→ /fr/home

```
1 ActionController::Routing::Routes.draw do |map|
2   lang_prefix = ':language'
3
4   map.resource 'home', :path_prefix => lang_prefix
5   map.resource 'home'
6 end
```

- Domaine → fr.domaine.tld
- Query string → /home?language=fr
- accept-language / session / valeur par défaut

Routage vers le contenu statique

- Images

→ /images/image_fr.png

→ /images/image_en.png

- Fichiers

→ /fichiers/paris_on_rails_fr.pdf

→ /fichiers/paris_on_rails_en.pdf

- Apache 2.0

- Négociation de contenu

→ LanguagePriority , prefer-language

Processus de traduction

- Ressources humaines
- Interfaces / outils pour la traduction
- Contenu partiellement traduit
- Qualité

Attention à...

- Attention à bien définir les besoins
- Attention à ne pas confondre région et langue
- Attention à toujours tenir compte de l'Unicode
- Attention aux méthodes de pluralization
- Attention aux validations (NPA, Téléphone, ...)
- Attention à la mise en forme et au tri
- Attention à ne pas négliger le travail de traduction

Pour aller plus loin ?

- ActiveSupport::Multibyte::Chars

<http://api.rubyonrails.org/classes/ActiveSupport/Multibyte/Chars.html>

- Globalize

<http://www.globalize-rails.org>

- Me contacter

yann.lugrin@liquid-concept.ch