

Localisation

~

Internationalisation in Ruby on Rails

Ondrej Jaura

Menu

- localisation
 - operating system, database
 - RoR application
 - Unicode
- internationalisation
 - RoR application
 - solutions
 - examples

Localization

Localisation

L10n

Localisation

- encoding
- operating system
- database
- RoR application
 - texts
 - date, time and numbers formatting
 - HTTP header, META tag
 - database connection

Encoding

- possibilities
 - (Unicode) UTF-8 – `utf-8`
 - Latin2 – `iso-8859-2`
 - MS Windows Central European – `cp1250`
- editor with encoding support
 - UTF-8
- converting encoding
 - `iconv -f latin2 -t utf-8`
`file.latin2.txt > file.utf-8.txt`

Operating System

- Linux
 - define locale
- example
 - Gentoo
 - /etc/locale.gen
 - add line(s)
 - sk_SK.UTF-8 UTF-8
 - cz_CZ.UTF-8 UTF-8
 - execute locale-gen

Database

- correct sorting
- uppercase – lowercase conversion
- example
 - PostgreSQL

```
su - postgres
```

```
initdb -D /var/lib/postgresql/data/ -lc-  
collate sk_SK.UTF-8
```

RoR application

- application text
- date, time and numbers formatting
- HTTP header
- META tag
- database connection
- libraries
- Unicode

Application Text

- view
 - static text
- controller
 - validation messages
- model
 - default values
 - values for select boxes
- keep one encoding

Formatting

- date
 - 23. jún 2007
 - June 23, 2007
- currency
- numbers
 - 1 234,56 Sk
 - \$ 1,234.56
- solution: own helper methods

HTTP Header

- **app/controllers/application.rb**

```
class ApplicationController < ActionController::Base
  before_filter :set_header

  def set_header
    response.headers['Content-Type'] =
      'text/html; charset=utf-8' #cp1250, iso-8859-2
  end
end
```

META Tag

- `app/views/layouts/standard.rhtml`
 - or another layout

```
<head>
```

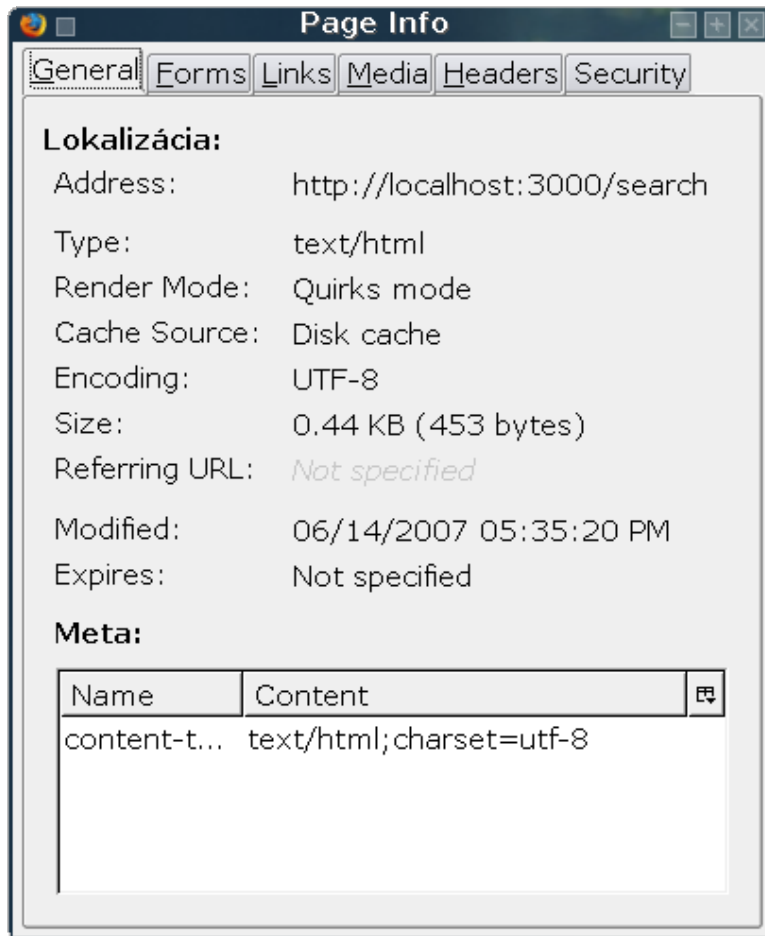
```
  <meta http-equiv="content-type"  
  content="text/html; charset=utf-8" />
```

```
</head>
```

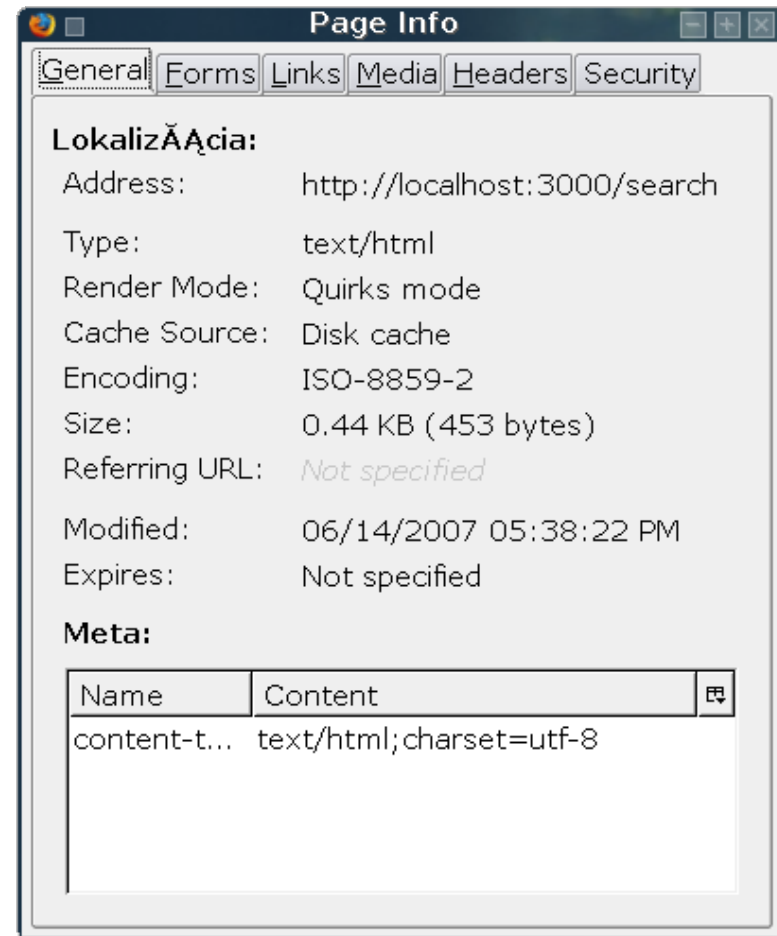
- `app/controllers/application.rb`
 - or corresponding controller

```
  layout 'standard'
```

Encoding Consequences



- HTTP header: utf-8
- META tag: utf-8



- HTTP header: iso-8859-2 (Latin2)
- META tag: utf-8

Database Connection

- **config/database.yml**
 - MySQL
 - `encoding: utf8`
 - PostgreSQL
 - `encoding: unicode`
- **create database with correct encoding**
 - PostgreSQL
 - `createdb -E utf-8 app_development`

Database Connection Configuration

```
development:
```

```
  adapter:  postgresql
```

```
  database: app_development
```

```
  username: ondrej
```

```
  password:
```

```
  host:     localhost
```

```
  encoding: unicode
```

Libraries

- **overwrite**
 - ActiveRecord
- **defined method for translation**
 - `as(text)`
 - ActiveSupport
- **h4ck it**
 - overwriting text messages manually

Ruby and Unicode

- Unicode

- `config/environment.rb`

- `# enable UTF-8 support`

- `$KCODE = 'u'`

- `require 'jcode'`

- limited support

- `"abc".size = 3`

- `"abč".size = 4` `"abč".jsize = 3`

- should change in Ruby 2.0

Internationalization

Internationalisation

I18n

Internationalization

- RoR application
- libraries and solutions for I18n
- examples
 - Gibberish
 - Ruby-GetText

RoR application

- encoding
- database connection
- application text
- date, time and numbers formatting
- libraries

Encoding

- UTF-8 advantage
 - uniform HTTP header
 - uniform META tag
 - uniform encoding of all files
- otherwise use particular encodings
 - example:
 - iso-8859-15 (Germany, UK)
 - iso-8859-2 (Czech, Slovakia, Poland)
 - iso-8859-5 (Bulgaria, Russia)

Database Connection

- optimal (and the only?) solution: UTF-8
- config/database.yml
 - MySQL:

```
encoding: utf8
```
 - PostgreSQL:

```
encoding: unicode
```
- sorting
 - according to database :)

Application Text

- view
 - static text
- controller
 - validation messages
- model
 - default values
 - values for select boxes
 - **multilingual content**

Formatting

- date
 - 23. jún 2007
 - June 23, 2007
- currency
- numbers
 - 1 234,56 Sk
 - \$ 1,234.56

Libraries

- global variables
 - ActiveRecord
- defined method for translation
 - `as(text)`
 - ActiveSupport
- h4ck it
 - connect to i18n library manually

Solutions for I18n

- own simple solution – a few words or sentences
- existing solutions
 - Ruby-GetText
 - Globalize
 - Gloc
 - Localize
 - Gibberish
 - ...

Gibberish

- simple solution
- `"text"[:id]`
- translations in `.yml` files (`sk.yml`, `cz.yml`)
 - `id: text`
- <http://errtheblog.com/post/4396>
- installation
 - `cd vendor/plugins/`
 - `svn co svn://errtheblog.com/svn/plugins/gibberish`

Gibberish \neg ext

- `"Hello!"[:hello]`

→ `Hello!`

- **translation in lang/sk.yml**

```
hello: Ahoj!
```

- **usage**

```
Gibberish.current_language = :sk
```

```
"Hello!"[:hello]
```

→ `Ahoj!`

Gibberish \neg ext and Parameter

- `"Hello {name}!"[:hello, "Susanne"]`
→ Hello Susanne!

- **translation in lang/sk.yml**

```
hello: Ahoj {name}!
```

- **usage**

```
Gibberish.current_language = :sk
```

```
"Hello {name}!"[:hello, "Zuzka"]
```

```
→ Ahoj Zuzka!
```

Gibberish RoR Application

- RoR Application
- login page
- welcome page
- English, Slovak and Czech

Gibberish – Principle

- `"default language text"[:id]`
- translation files in `lang/language_code.yml` directory
 - `id: text translation`
- `Gibberish.current_language = :language_code`

Ruby-GetText

- complex solution
 - pluralization rules
 - ActiveRecord error messages
 - sophisticated tools
- complicated pluralization rules
 - (English) 1 car, 2 cars, 100 cars
 - (Slovak) 1 auto, 2 autá, 100 áut
- <http://www.yotabanana.com/hiki/ruby-gettext-howto-rails.html>
- <http://manuals.rubyonrails.com/read/chapter/105>

Ruby-GetText → translation

- `_ 'Hello'`
- `_ ('Hello')`
- in controllers, views and models (*.rb, *.rhtml)
- translation
 - PO files (.po)
 - produced from source code
 - MO files (.mo)
 - binary
 - http://www.gnu.org/software/gettext/manual/html_node/gettext_5.html

Ruby-GetText + installation

- `gem install gettext`
- create `lib/tasks/gettext.rake`
 - <http://www.yotabanana.com/hiki/ruby-gettext-howto-rails.html>
 - `updatepo`
 - `makemo`
- `mkdir po`
 - `mkdir sk_SK`
 - `mkdir cz_CZ`

Ruby-GetText + installation (cont.)

- **config/environment.rb**

```
$KCODE = 'u'  
require 'jcode'  
require 'gettext/rails'
```

- **app/controller/application.rb**

```
require 'gettext/rails'  
init_gettext "application_name"
```

Ruby-GetText Usage

- rake updatepo
 - first time
 - creates po/application.pot
 - copy po/application.pot to particular language subdirectory as application.po file
 - translation of .po files
 - next times
 - updates .pot file and all .po files
 - translation of .po files

Ruby-GetText – Usage (cont.)

- rake makemo
 - creates directory locale
 - copies .mo files

Ruby-GetText – PO configuration

- plural form

- Slovak, Czech, Polish

```
nplurals=3; plural=(n%10==1 &&  
n%100!=11 ? 0 : n%10>=2 && n%10<=4 &&  
(n%100<10 || n%100>=20) ? 1 : 2);
```

- English

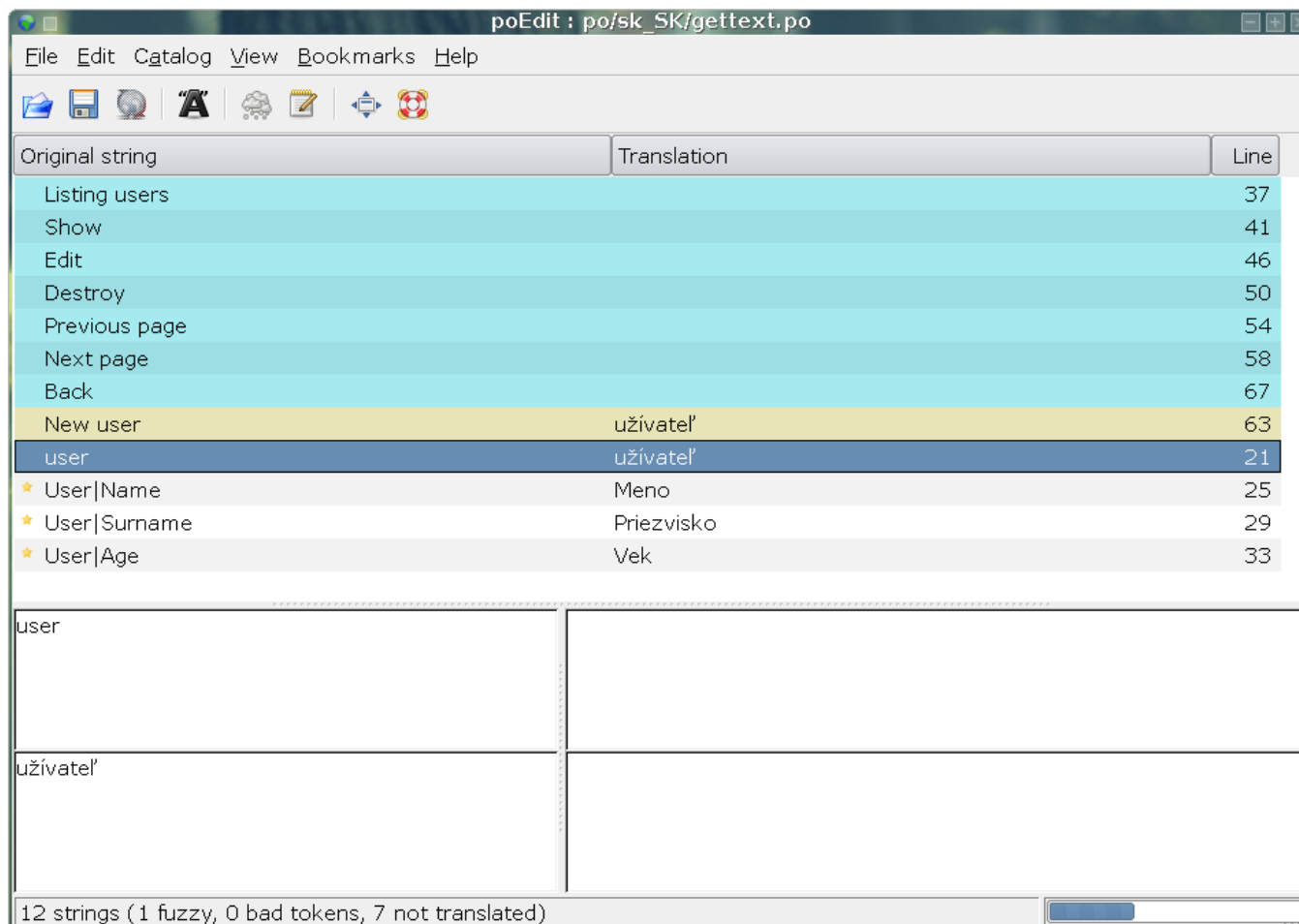
```
nplurals=2; plural=n == 1 ? 0 : 1;
```

- content type

```
text/plain; charset=UTF-8
```

Ruby-GetText – poEdit

- <http://www.poedit.net>



Ruby-GetText in a RoR Application

- all texts (controller, views, models)

```
_('text')
```

```
_('text %{param}') % {:param => value}
```

```
_('text %s') % value
```

- example

```
<%= link_to 'Show', :action => 'show',  
:id => user %>
```

```
<%= link_to _('Show'), :action =>  
'show', :id => user %>
```

- `GetText.locale = "sk_SK"`

Ruby-GetText - RoR Application

- RoR Application
- database of users
- English and Slovak

User Language

- cookies
- user profile
 - stored in database
 - but: login page

User Language – Example

```
class ApplicationController < ActionController::Base

  prepend_before_filter :default_locale

  def default_locale
    if (cookies["lang"].blank?)
      GetText.locale = "sk"
    else
      GetText.locale = cookies["lang"]
    end
  end

  def setLanguage(langCode)
    cookies["lang"] = langCode
    GetText.locale = langCode
  end
end
```

Links

- **internationalization plugins**

- <http://agilewebdevelopment.com/plugins/category/8>
- <http://wiki.rubyonrails.com/rails/pages/InternationalizationComparison>
- <http://www.yotabanana.com/hiki/ruby-gettext-howto-rails.html>
- <http://manuals.rubyonrails.com/read/chapter/105>

- **others**

- <http://wiki.rubyonrails.com/rails/pages/HowToUseUnicodeStrings>
- <http://poocs.net/2006/7/24/revisiting-time-strftime-localization>
- <http://www.fi.muni.cz/~adelton/l10n/postgresql-nls-string/README>

Thank You

- Ondrej Jaura
- ondrej@valibuk.net
- www.valibuk.net
 - my blog
- www.rubyagent.com
 - multi agent systems in Ruby
- www.metasearched.org
 - metasearch theory