

internacionalización rails y gettext

una *buena* internacionalización

habilitar soporte multicultural

lenguajes, monedas, formatos y franjas horarias

localización

traducirlo para una determinada cultura

estándares

utf-8, ISO (639-3166), posix, RFC3066, profiles

interacción con el usuario

¡quiero mi *locale*! (preferences | browser o sistema | default)

gnu gettext

librería i18n C/C++ portada (ruby, php, java...)

gnu gettext: ventajas

extracción automática de cadenas del código fuente

generación de ficheros portables y eficientes en

ejecución

po & mo

traducción y comunidad

rosetta, pootle & poedit

herramientas de gestión

xgettext, msgmerge & msgfmt

cobertura multi-idioma real

plurales, polisemias

gnu gettext: ventajas

extracción automática de cadenas del código fuente

generación de ficheros portables y eficientes en

eje

po

```
Plural-Forms: nplurals=3; \
```

traduc

```
    plural=n%10==1 && n%100!=11 ? 0 : \
```

ros

```
        n%10>=2 && n%10<=4 && (n%100<10 || n%100>=20) ? 1 : 2;
```

herran

xg

cobertura multi-idioma real

plurales, polisemias

gnu gettext: el proceso

10 configurar()

20 marcar_código()

30 generar_o_combinar(.PO)

40 traducir(.PO)

50 generar binarios(.PO => .MO)

60 goto *randomly* 20, 30, 40, 50

i18n en rails: las opciones

Localization Simplified

Globalize

Localize

Ri18n

Gloc

Ruby-GetText

ruby-gettext: puesta en marcha

1. instalar la gema

`gem install ruby-gettext`

`ruby setup.rb` para generar sus propios mo

2. configurar el entorno rails

directorios `RAILS_ROOT/po/<<locales>>`

Rakefile

`init_gettext 'dominio'` en `application.rb`

Soporte utf-8: `$KCODE = 'u'` `require 'jcode'` `require 'gettext/rails'`

3. marcar código preferentemente con cabeza

`_('Hola % user soy % app') % { :user => 'Belén', :app => 'Tractis' }`

`n_('% d developer', '% d developers', number_of_fools)`

4. gestionar ficheros

`rake updatepo` (generar `.pot` y combinar con `.po` existentes)

`rake makemo` (on deploy)

ruby-gettext y rails: vistas /controladores

controladores, rutas, filtros

```
before_filter bindtextdomain('myApp', {:locale => 'es'})  
map.connect ':controller/:lang/:action/:id'
```

vistas

plantillas localizadas: `/views/controller/hola_ja.rhtml`

```
<% set_locale "fr" %>  
<%= _("Hello world") %>
```

ruby-gettext: modelos

untranslate, untranslateall

mensajes personalizados

```
N_("User|Passconfirmation")
```

```
validates_presence_of :password
```

```
errors.add("password", _("%{fn} is wrong!")) unless password ==  
  pass_confirmation
```

```
validates_presence_of :title, :message => N_("%{fn} está vacío")
```

ruby-gettext: hacks

```
alias distance_of_time_in_words_without_locale  
distance_of_time_in_words
```

```
bindtextdomain("rails")
```

```
MINUTES = [/^(\d+) minutes? $/, Nn_('1 minute', '%{num} minutes')]
```

```
DAYS     = [/^(\d+) days? $/, Nn_('1 day', '%{num} days')]
```

ruby-gettext: próximos pasos

<http://www.yotabanana.com/>

/config/environment.rb

http://docs.google.com/Doc?id=ajdm2vfc349k_21ccmddz

/app/controllers/application.rb

http://docs.google.com/Doc?id=ajdm2vfc349k_17fccffc

/app/controllers/test_controller.rb

http://docs.google.com/Doc?id=ajdm2vfc349k_18cmq58m

/app/views/test/index.rhtml

http://docs.google.com/Doc?id=ajdm2vfc349k_19hdpvwx

/app/models/test_mailer.rb

http://docs.google.com/Doc?id=ajdm2vfc349k_20dtrmwb

mi experiencia en tractis

entorno colaborativo

pootle + svn + gettext

es posible innovar

db oriented

svn integrated

real translators

proyecto especial: f.i.t. – found in translation

integrado en el desarrollo, transparente, multiproyecto

se liberará bajo licencia GPL

gracias

Juan S. Pérez Herrero % juanseph@gmail.com

<http://www.tractis.com> – <http://blog.negotiation.com>