Rails Internationalization

Slides at:

http://graysky.org/public/rails_i18n_talk.pdf

Can Rails Scale?

Can Rails Scale?

... to Multiple Languages?

Agenda

- Who is Mike?
- Internationalization (i18n) Overview
- Rails Support for i18n
- Pitfalls
- Translator plugin
- Further Reading

Mike Champion

- Speaks English and Spanish very poorly
- Heading to RailsConf - see you in Las Vegas?
- The Usual URIs
  - graysky.org
  - twitter.com/graysky
  - github.com/graysky
- Developer at SnapMyLife
  - Mobile photo sharing
  - Global user base

Iñtërnâtiônàlizætiøn

- Level 0: App handles utf-8 data
  - Database & `database.yml` configuration
  - Multi-byte string handling *(since Rails 1.2 / Ruby 1.9)*
    - Ruby 1.8: `“café”.length => 5`
    - Rails MB: `“café”.mb_chars.length => 4`
- Level 1: App is internationalized
  - Strings extracted, dates, etc. *(support in Rails 2.2)*
- Level 2: App is localized
  - Strings translated / tested *(on your own - good luck!)*

I18n Concerns

- Strings
  - Concatenation is evil when used to form sentences
- Sorting & Searching
  - Swedish: z < ö while in German: ö < z
- Dates/Times & Timezones
  - 12/4/09 vs. 4/12/09
- Units, Addresses & Currency
  - Miles vs. kilometers, 5-digit zip codes
- Cultural
  - Surname & given name
  - Icons, colors, etc.

¡Dios mío!

(Lo Siento)

I18n Module

- I18n library in ActiveSupport for 2.2
  - Locale files in `config/locale`
    - `I18n.load_path += File.join(“more.yml”)`
- Simple backend uses YAML (and Ruby) files:

```
en:
  say_hello: 'Hello World'
  date:
    short: “%Y-%m-%d”

I18n.translate('say_hello')
I18n.localize(Time.now,
              :format => :short)
```

I18n Module, cont.

- Variable interpolation

```
say_hello: 'Hello {{name}} World'

I18n.t('say_hello', :name =>'Ruby')
```

- Pluralization *(only supports some languages)*

```
minutes_ago:
  one: '1 minute ago'
  other: '{{count}} minutes ago'

I18n.t('minutes_ago', :count => 2)
```

Rails I18n

- Internationalized ActiveSupport for:
    - Numbers - `number_with_delimiter, etc.`
    - Datetime - `distance_of_time_in_words, etc.`
    - Currency - `number_to_currency`
- Localized views - `index.de.erb.html` *(Rails 2.3)*
    - Useful if translating to one other language

International Models*

- Map model name & attributes to human-readable strings using I18n lookup
  - `User.human_name() => “Customer”`
  - `User.human_attribute_name(‘pin’) => “Password”`
  - Useful to easily change user-visible column names
- Translate ActiveRecord validation errors
  - `validates_presence_of :pin`
  - `activerecord.errors.messages.models.user.attributes.pin.blank: “Password can’t be blank, yo”`

** Not the Exciting Kind...*

Sorting (aka Collation)

- Example: Germans, French and Swedes sort the same characters differently
- MySQL supports different collation algorithms
  - `utf8_general_ci` - faster, less correct
  - `utf8_unicode_ci` - slightly slower, better
    - implements Unicode Collation Algorithm
      - http://www.unicode.org/reports/tr10/
- Haven’t seen a Ruby implementation

Pitfalls

- Gems & plugins may not be i18n’d
- Embedded links/markup are painful
  - `“You <span>must</span> out my <%= link_to(“awesome blog”, ‘http://foo’) %> today!”`
  - Prefer label : value when possible
  - Text verbosity will change layouts.
- Duplicate keys shadow each other in yml
- `MY_ERROR_MSG = t(‘bad_password’)`
- Careful with fragment/action caching

Translator

- I18n integration could be more Rails-y
- Automatically determines scoping using a convention
  - Adds "t" method to views, models, controllers, mailers
  - `t('title')` in blog_posts/show.erb will fetch `t('blog_posts.show.title')`
- Test helper to any catch missing translations
- Pseudo-translation (ex. "[My Blog]") to find missing extractions
  - Useful for testing layouts when text expands in other language
- Locale fallback to `default_locale` if cannot find string
- Plugin at: http://github.com/graysky/translator

Translator Key Convention

```
en:
  # controller
  blog_posts:
    # typical actions
    index:
      title: "My Blog Posts"

    # footer partial (non-shared)
    footer:
      copyright: "Copyright 2009"

  # Layouts
  layouts:
    blog_layout:
      blog_title: "The Blog of Ricky"

  # ActionMailers
  blog_comment_mailer:
    comment_notification:
      subject: "New Comment Notification"

  # ActiveRecord (note singular)
  blog_posts:
    byline: "Written by {{author}}"
```

Determining Locale

- Locale more than just Language
- User Preference
- Domain name - *foo.com vs. foo.de*
- Path prefix - *foo.com/de/blog_posts*
- HTTP Accept-Language Header
    - `zh-Hans (simplified Chinese) => “zh”`
- IP Geolocation
    - GeoIP Lite Country database

Further Reading

- Rails I18n Guide
  - http://guides.rubyonrails.org/i18n.html
- i18n_demo_app
  - http://github.com/clemens/i18n_demo_app
- Rails i18n Group
  - http://groups.google.com/group/rails-i18n
- Translate - web ui for yml translatations
  - http://github.com/newsdesk/translate
- Unicode Common Locale Data Repository (CLDR)
  - http://www.unicode.org/cldr/
- "Survival Guide to i18n" (not Rails specific)
  - http://www.intertwingly.net/stories/2004/04/14/i18n.html

Questions?