USER ALERTS IN RAILS WITH MESSAGE BLOCK

& Testing Gems with Cucumber/Cukigem

Ben Hughes
@rubiety

WHAT USER ALERTS?

Flash Messages:

```
flash[:notice] = "User Created"
```

```
<% if flash[:notice] %>
  <p><%= h flash[:notice] %></p>
<% end %>
```

ActiveRecord Errors:

```
class User < ActiveRecord::Base
  validates_presence_of :login
end
```

```
<% form_for(@user) do |f| %>
  <%= error_messages_for :user %>
<% end %>
```

MESSAGE BLOCK FEATURES

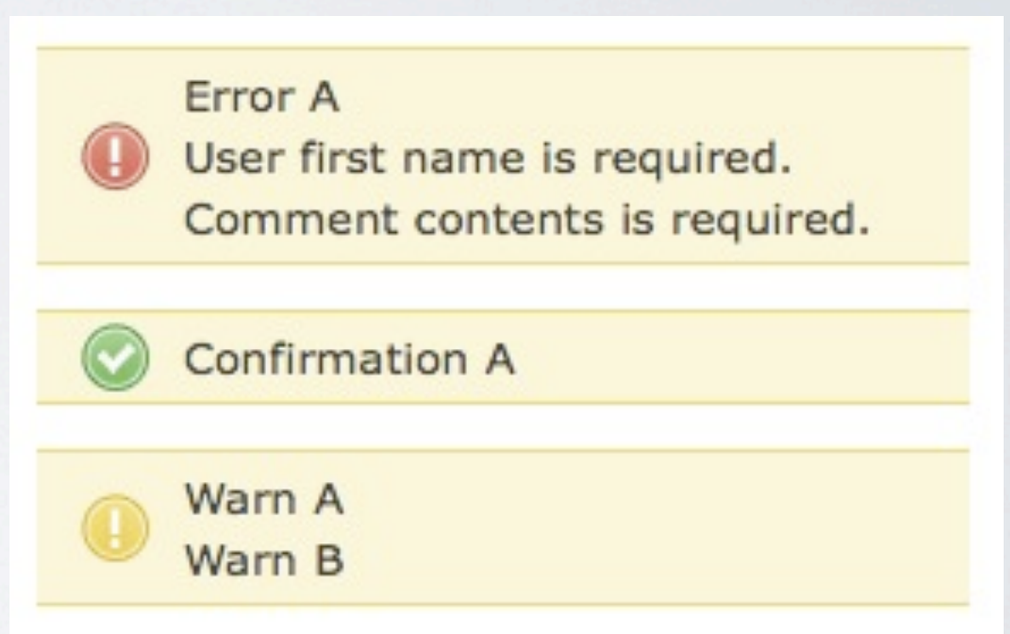
- Merges AR errors and flash messages together
- Groups by message type (confirm, error, warn, etc.)
- Accepts flashes with array of strings
- Pull AR errors from multiple models
- Update via JavaScript for easy Ajax

```ruby
class CommentsController
  def create
    @new = User.create(:first_name => "")
    @comment = Comment.create(:contents => "")

    flash.now[:error] = "Error A"
    flash.now[:confirm] = "Confirmation A"  # Note you can use different types
    flash.now[:warn] = ["Warn A", "Warn B"]  # Can set to an array for multiple messages
  end
end
```

```
<%= message_block :on => [:user, :comment] %>
```

```
<div id="message_block" class="message_block">
  <ul class="error">
    <li>Error A</li>
    <li>User first name is required.</li>
    <li>Comment contents is required.</li>
  </ul>
  <ul class="confirm">
    <li>Confirmation A</li>
  </ul>
  <ul class="warn">
    <li>Warn A</li>
    <li>Warn B</li>
  </ul>
</div>
```



```
<div id="something">
  <%= message_block :on => :job %>
</div>

<script type="text/javascript">
  document.observe('dom:loaded', function() {
    message_block = new MessageBlock('message_block');
    message_block.update({
      error: ['Error One', 'Error Two'],
      confirm: ['Confirmation Message']
    });

    // Clears the message block
    message_block.clear();
  });
</script>
```

```
render :status => :unprocessable_entity, :json => { 'error' => @job.errors.full_messages }
```

INSTALLATION & ASSETS

```
# In Gemfile:
gem "message_block"
```

```
$ rake message_block:install
```

```
<%= stylesheet_include_tag "message_block" %>
```

TESTING GEMS W/CUCUMBER

Fully testing a Rails gem plugin from the outside in involves a "host application", how do we do this with Cucumber?

SIMPLE PROBLEM? NO

- Class Caching in test environment is enabled by default
- Need to reset database connections after migrations
- Making webrat/capybara "attach" to the host application is difficult
- Don't want to regenerate rails app on every story run, so need to "reset" application to blank state
- Making the host app use your gem with Bundler is challenging

CUKIGEM

A series of step definitions for cucumber to help testing your gems within a host (test) Rails application.

```
Given "I generate a rails application"
Given "I have a rails application"
Given "I ensure a rails application is generated"
  Given "I turn off class caching in the rails application"
  Given "I ensure a rails application is generated"
  Given "I reset the rails application"
  Given "I reset the routes file"
  Given "I define an application controller"
  Given "I reset the Gemfile"
  When "the Gemfile is configured for testing"
  When "the Gemfile contains this gem"
When "I setup the database"
When "I start the rails application"
When "I save the following as 'app/controllers/tasks_controller.rb'"
When "I append the following to 'Gemfile'"
When "I run 'rake message_block:install'"
Then "the file 'public/stylesheets/message_block.css' should exist"
```

QUESTIONS?

Ben Hughes

@rubiety

http://github.com/rubiety