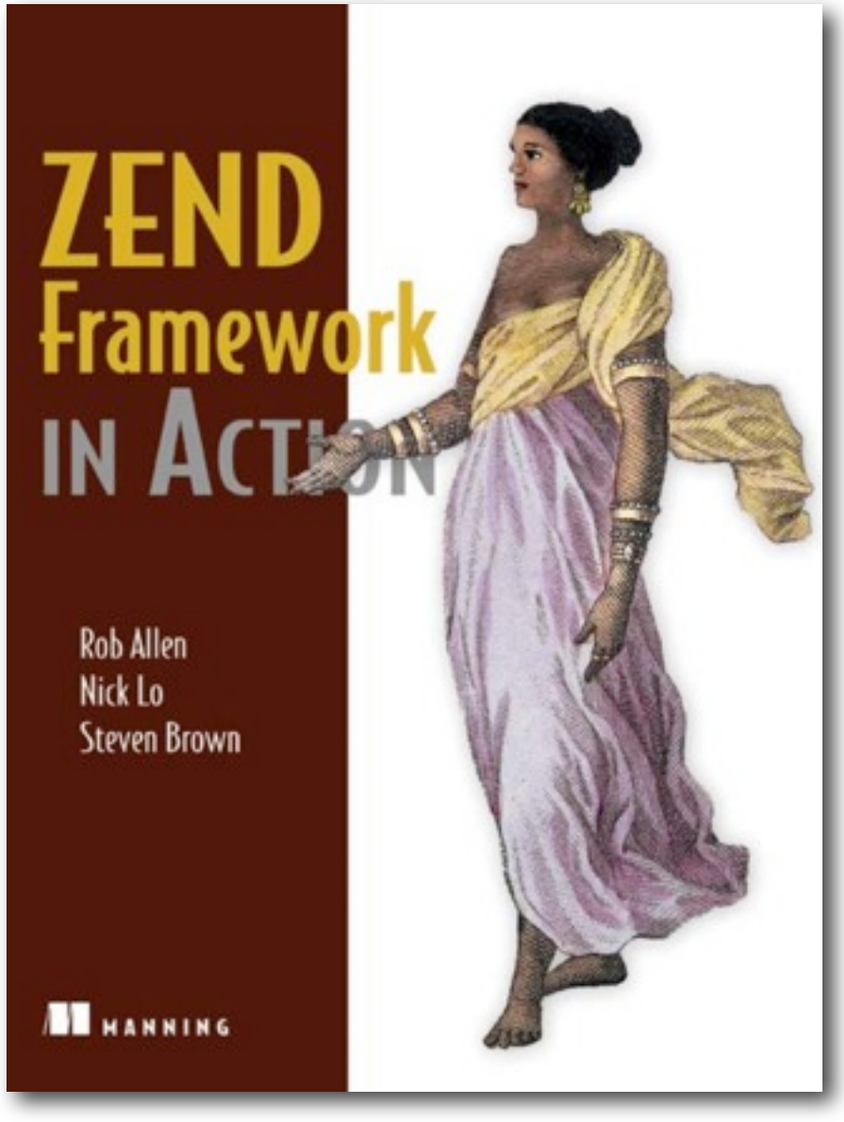
Working with Zend_Form

Rob Allen

Rob Allen?


php


ZEND
Framework
IN ACTION
Rob Allen
Nick Lo
Steven Brown
MANNING

What is Zend_Form?

Architecture

- Forms: the centre piece
- Forms are made up of:
  - Elements, Display groups, Sub-forms
- Each element is made up of:
  - Filters, Validators, Decorators

Zend_Form_Element

- Lynchpin of the system
- Holds
  - element meta data
  - filter chains
  - validator chains
  - decorators for rendering
- Examples:
  - Text, Textarea, Select, etc

Grouping of elements

- Zend_Form_DisplayGroup
  - group forms visually
  - usually rendered as a Fieldset
- Zend_Form_SubForm
  - group forms logically
  - allows partial validation

Filters

- Normalise input prior to validation
- Changes the data
- Uses Zend_Filter classes
- Examples:
  - StripTags, StringTrim, Digits, StringToLower

Validators

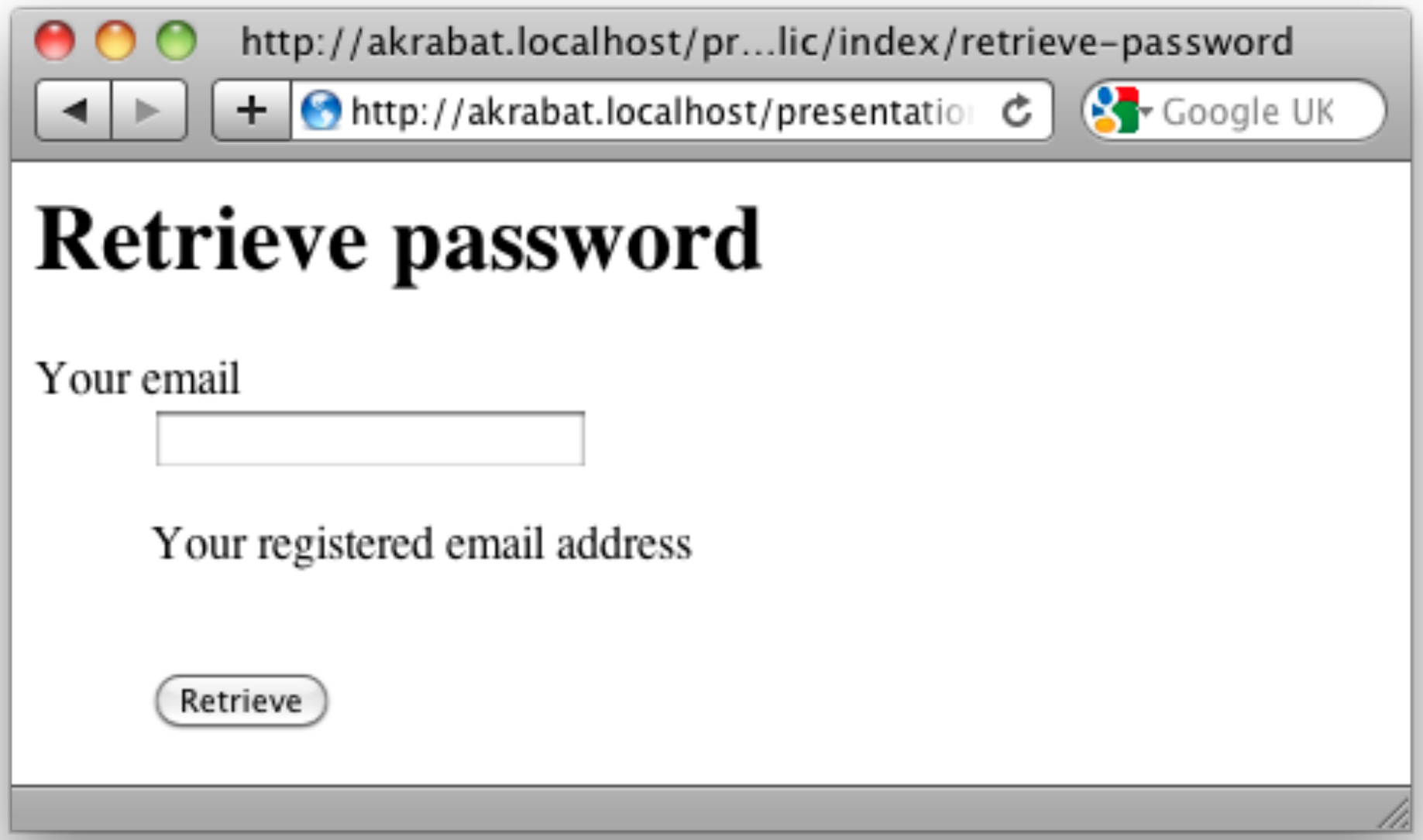
- Yes / No decision on acceptability of data
- Uses Zend_Validate classes
- Examples:
  - StringLength, GreaterThan, NotEmpty, Int
  - CreditCard / Hostname / Iban / Isbn

Decorators

- Render the form and elements
- Combination of Decorator & Strategy patterns
  - Each one decorates the content passed to it
  - Typically inside -> out
  - Return value replaces previous value
- Examples:
  - ViewHelper, Label, HtmlTag

Zend_Form in use

Retrieve password



HTML

```
<h1>Retrieve password</h1>
<form action="" method="post">
<dl class="zend_form">

<dt><label for="email">Your email</label></dt>
<dd><input type="text" name="email" value=""></dd>

<dt></dt>
<dd><input type="submit" name="retrieve" value="Retrieve"></dd>
</dl>
</form>
```

Creation in code

```
class Application_Form_RetrievePassword extends Zend_Form
{
    public function init()
    {
        /* Form Elements & Other Definitions Here ... */
    }
}
```

Email element

```
$email = new Zend_Form_Element_Text('email');
$email->setRequired(true);
$email->setLabel('Your email');
$email->setDescription('Your registered email address');
$email->addValidators(array(
    'EmailAddress',
    new Zend_Validate_StringLength(array('min'=>5))
    ));
$this->addElement($email);
```

Submit element

```
$submit = new Zend_Form_Element_Submit('retrieve');
$submit->setLabel('Retrieve');
$submit->setIgnore(true);
$this->addElement($submit);
```

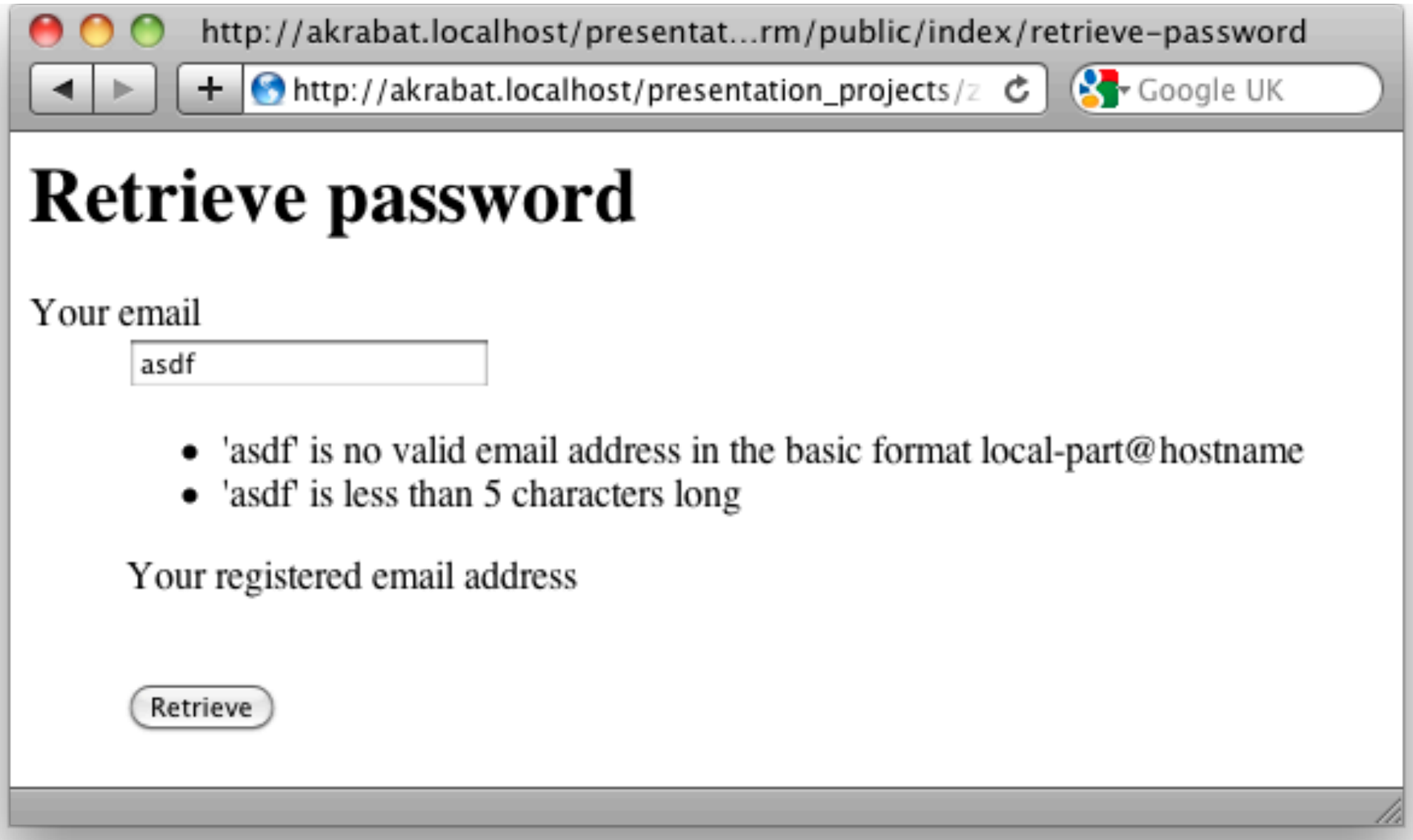
Action code

```
public function retrievePasswordAction()
{
    $request = $this->getRequest();

    $form = new Application_Form_RetrievePassword();
    if ($request->isPost()) {
        if ($form->isValid($request->getPost())) {
            // success
            $formData = $form->getValues();
        }
    }

    $this->view->form = $form;
}
```

View code

```
<?php echo $this->form; ?>
```

Errors!


http://akrabat.localhost/presentat...rm/public/index/retrieve-password
http://akrabat.localhost/presentation_projects/z
Google UK
Retrieve password
Your email
asdf
'asdf' is no valid email address in the basic format local-part@hostname
'asdf' is less than 5 characters long
Your registered email address
Retrieve

“Translate” the errors

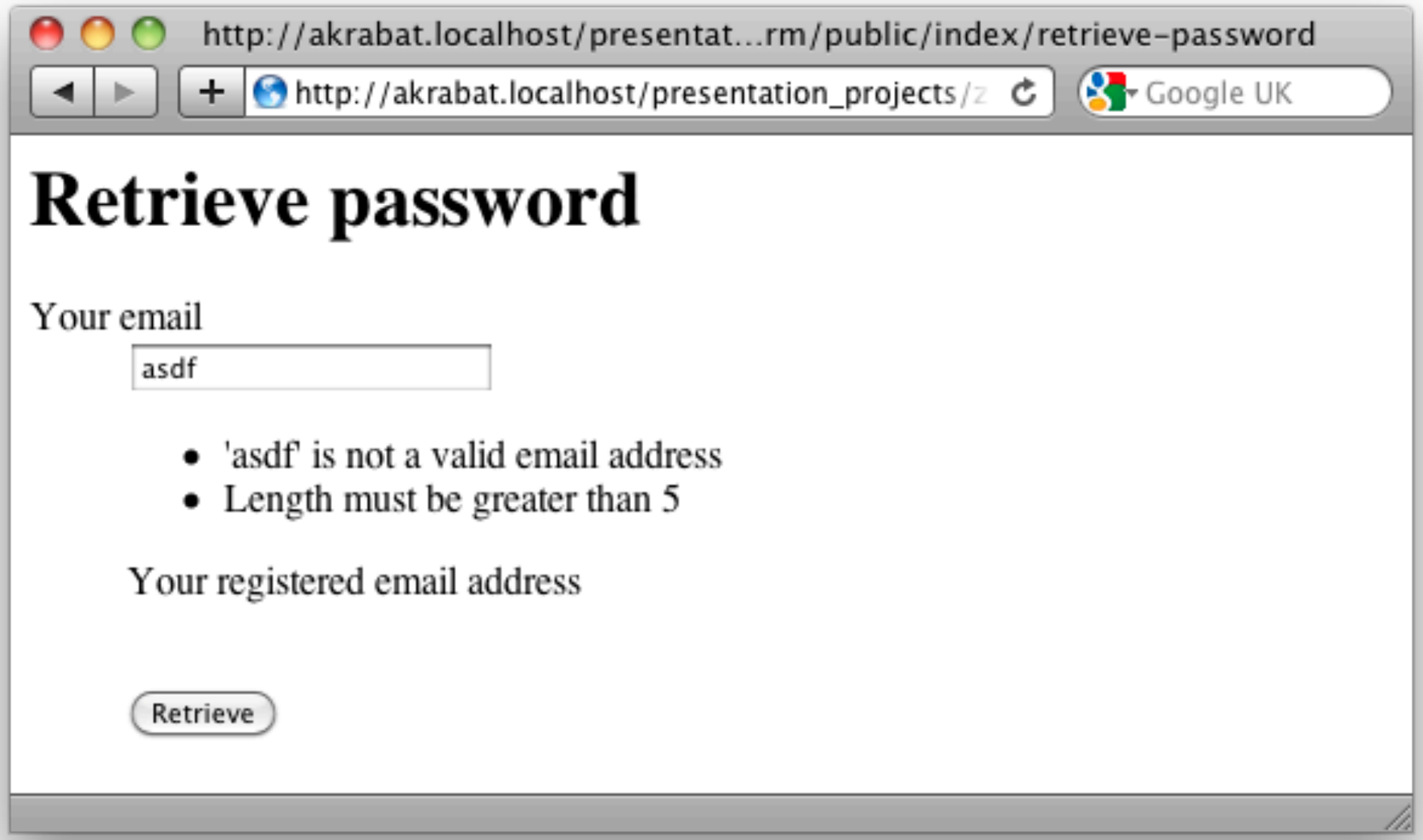
```
// configs/translations/forms.php
return array(
    Zend_Validate_NotEmpty::IS_EMPTY =>
        '%field% is required',
    Zend_Validate_StringLength::TOO_SHORT =>
        'Length must be greater than %min%',
    Zend_Validate_StringLength::TOO_LONG =>
        'Length must be smaller than %max%',
    Zend_Validate_EmailAddress::INVALID =>
        "'%value%' is not a valid email address",
    Zend_Validate_EmailAddress::INVALID_FORMAT =>
        "'%value%' is not a valid email address",
    // ...
);
```

Attach to the form

```
// Application_Form_RetrievePassword::init()
$path = APPLICATION_PATH. '/configs/translations/forms.php';
$translate = new Zend_Translate('array', $path, 'en');
$this->setTranslator($translate);
```

```
// or for all forms, in your Bootstrap
public function _initFormErrorMessages()
{
  $path = APPLICATION_PATH. '/configs/translations/forms.php';
  $translate = new Zend_Translate('array', $path, 'en');
  Zend_Form::setDefaultTranslator($translate);
}
```

Better messages



Default element decorators

1. Zend_Form_Decorator_ViewHelper
2. Zend_Form_Decorator_Errors
3. Zend_Form_Decorator_Description
4. Zend_Form_Decorator_HtmlTag
5. Zend_Form_Decorator_Label

Default element HTML

```
<dt id="email-label">
    <label for="email" class="required">Your email</label>
</dt>
<dd id="email-element">
    <input type="text" name="email" id="email" value="asdf">
    <ul class="errors">
        <li>'asdf' is not a valid email address</li>
        <li>Length must be greater than 5</li>
    </ul>
    <p class="description">Your registered email address</p>
</dd>
```

ViewHelper

```
<dt id="email-label">
    <label for="email" class="required">Your email</label>
</dt>
<dd id="email-element">
    <input type="text" name="email" id="email" value="asdf">
    <ul class="errors">
        <li>'asdf' is not a valid email address</li>
        <li>Length must be greater than 5</li>
    </ul>
    <p class="description">Your registered email address</p>
</dd>
```

Errors

```
<dt id="email-label">
    <label for="email" class="required">Your email</label>
</dt>
<dd id="email-element">
    <input type="text" name="email" id="email" value="asdf">
    <ul class="errors">
        <li>'asdf' is not a valid email address</li>
        <li>Length must be greater than 5</li>
    </ul>
    <p class="description">Your registered email address</p>
</dd>
```

Description

```
<dt id="email-label">
    <label for="email" class="required">Your email</label>
</dt>
<dd id="email-element">
    <input type="text" name="email" id="email" value="asdf">
    <ul class="errors">
        <li>'asdf' is not a valid email address</li>
        <li>Length must be greater than 5</li>
    </ul>
    <p class="description">Your registered email address</p>
</dd>
```

HtmlTag

```
<dt id="email-label">
    <label for="email" class="required">Your email</label>
</dt>
<dd id="email-element">
    <input type="text" name="email" id="email" value="asdf">
    <ul class="errors">
        <li>'asdf' is not a valid email address</li>
        <li>Length must be greater than 5</li>
    </ul>
    <p class="description">Your registered email address</p>
</dd>
```

Label

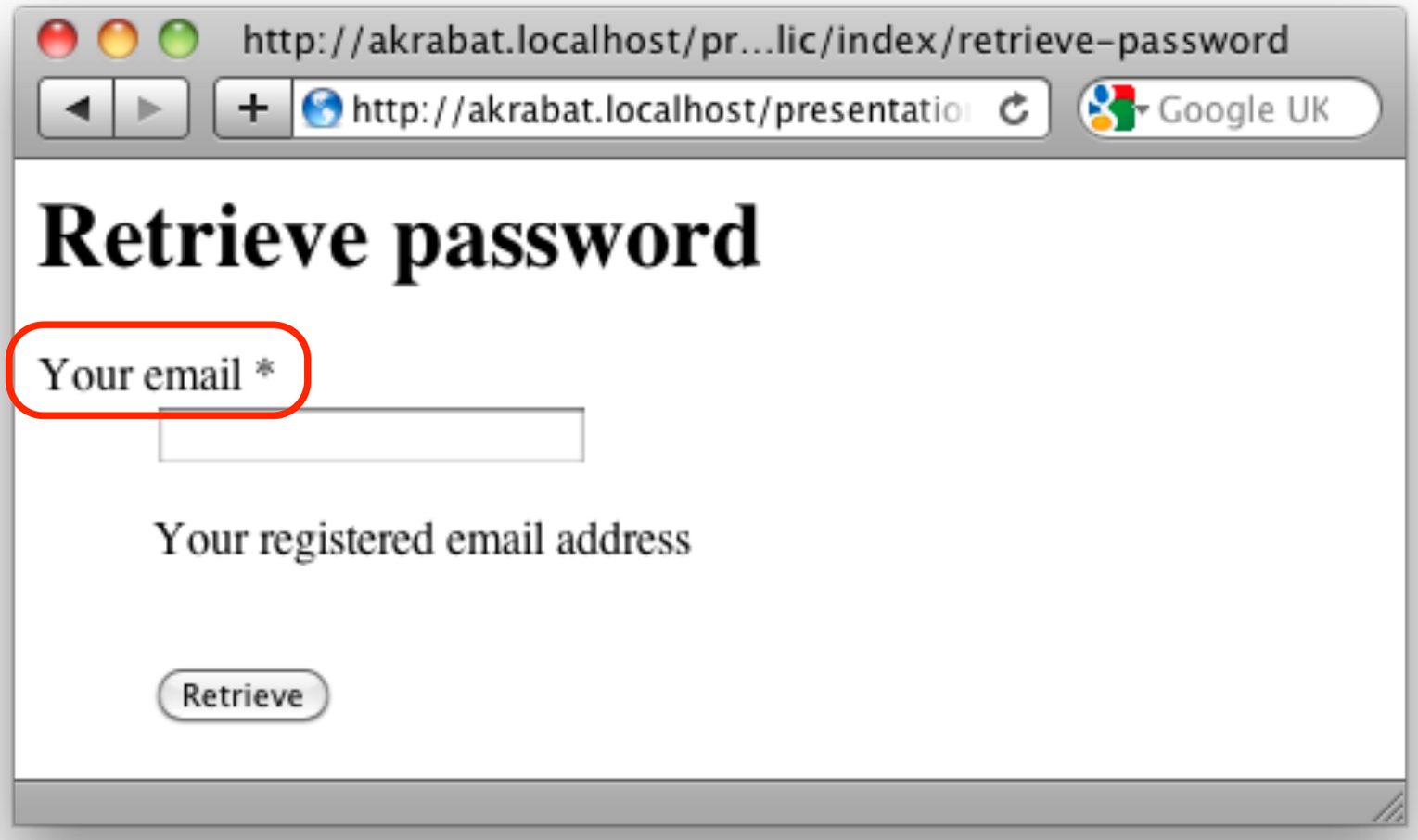
```
<dt id="email-label">
    <label for="email" class="required">Your email</label>
</dt>
<dd id="email-element">
    <input type="text" name="email" id="email" value="asdf">
    <ul class="errors">
        <li>'asdf' is not a valid email address</li>
        <li>Length must be greater than 5</li>
    </ul>
    <p class="description">Your registered email address</p>
</dd>
```

Decoration code

```
// Zend_Form_Element::loadDefaultDecorators()
$this->addDecorator('ViewHelper')
      ->addDecorator('Errors')
      ->addDecorator('Description', array('tag' => 'p',
          'class' => 'description'))
->addDecorator('HtmlTag', array('tag' => 'dd',
     'id' => $this->getName() . '-element'))
->addDecorator('Label', array('tag' => 'dt'));
```

Add required asterisk

```
$label = $email->getDecorator('Label');
$label->setOption('requiredSuffix', ' *');
```



Using LI instead

```
$email->clearDecorators();
$email->addDecorator('ViewHelper')
      ->addDecorator('Errors')
      ->addDecorator('Description', array('tag' => 'p',
          'class' => 'description'))
      ->addDecorator('Label')
      ->addDecorator('HtmlTag', array('tag' => 'li',
          'id' => $email->getName() . '-element'));
```

Element HTML for an LI

```
<li id="email-element">
    <label for="email" class="required">Your email</label>
    <input type="text" name="email" id="email" value="asdf">
    <ul class="errors">
        <li>'asdf' is not a valid email address</li>
        <li>Length must be greater than 5</li>
    </ul>
    <p class="description">Your registered email address</p>
</li>
```

Form-level decorators

```
<form enctype="application/x-www-form-urlencoded"
    action="" method="post">
<dl class="zend_form">
 <!-- elements go here -->
</dl>
</form>
```

1. Zend_Form_Decorator_FormElements
2. Zend_Form_Decorator_HtmlTag
3. Zend_Form_Decorator_Form

FormElements

```
<form enctype="application/x-www-form-urlencoded"
    action="" method="post">
<dl class="zend_form">
 <!-- elements go here -->
</dl>
</form>
```

HtmlTag

```
<form enctype="application/x-www-form-urlencoded"
    action="" method="post">
<dl class="zend_form">
 <!-- elements go here -->
</dl>
</form>
```

Form

```
<form enctype="application/x-www-form-urlencoded"
    action="" method="post">
<dl class="zend_form">
 <!-- elements go here -->
</dl>
</form>
```

Default form decorators

```
// Zend_Form::loadDefaultDecorators()
$this->addDecorator('FormElements')
     ->addDecorator('HtmlTag', array('tag' => 'dl',
         'class' => 'zend_form'))
     ->addDecorator('Form');
```

UL decoration: Form

```
$this->clearDecorators();
$this->addDecorator('FormElements')
     ->addDecorator('HtmlTag', array('tag' => '<ul>'))
     ->addDecorator('Form');
```

Full UL HTML

```
<form action="" method="post">
<ul>
  <li id="email-element">
    <label for="email" class="required">Your email</label>
    <input type="text" name="email" id="email" value="asdf">
    <ul class="errors">
        <li>'asdf' is not a valid email address</li>
    </ul>
    <p class="description">Your registered email address</p>
  </li>
  <li id="retrieve-element">
        <input type="submit" name="retrieve" value="Retrieve">
  </li>
</ul>
</form>
```

Create your own

```
$this->addPrefixPath('App_Form', 'App/Form');
$this->addElementPrefixPath('App_Form', 'App/Form');
$this->addElementPrefixPath('App_Validate', 'App/Validate',
        'VALIDATE');
```

Yes/No element

```
class App_Form_Element_YesNo extends Zend_Form_Element_Radio
{
    public function init()
    {
        $this->addMultiOptions(array('1'=>'Yes', '0'=>'No'));
        $this->setSeparator(' '); // do not want a <br/>
    }
}

// Usage in form's init()
$spam = new App_Form_Element_YesNo('marketing');
$spam->SetLabel('Can we spam you?');
$this->addElement($spam);
```

ThumbsUp decorator

```
class App_Form_Decorator_ThumbsUp extends
    Zend_Form_Decorator_Abstract
{
    public function render($content)
    {
        $errors = $this->getElement()->getMessages();
        $separator = $this->getSeparator();
        if (empty($errors)) {
            return $separator . '<img src="/thumbsUp.gif" />';
        }

        return $content;
    }
}
```

In conclusion

- Zend_Form provides a flexible solution to a tricky problem
- Validation and filtering can ensure you get good data
- Understanding of decorators is key to a great form
- CSS is for presentation!

Questions?

Thank you

feedback: http://joind.in/1591

email: rob@akrabat.com

twitter: @akrabat

QR Code