

Swiftysite Technique

Extending WebGUI in a maintainable way

Henry Tang
Long Term Results B.V.

Introduction

Swiftysite

- What is Swifysite?
- Demo

Technique

- Development
- Runtime Enviroment

Discussion

- Ideas / Questions?

What is Swiftysite?

- Heavily patched WebGUI version
- Simplified user interface. Allow users to setup website without WebGUI training
- Simplified template system (Swiftysite Designer)
- Limited set of assets
- Site wide version tag
- Easy photo upload (Jumploader)
- SwiftyShop

What is Swiftysite?

Demo

Development

How to patch WebGUI in a maintainable way? (Minimize effort to upgrade to new WebGUI version)

- Avoid editing WebGUI source files directly
- Use the power of Perl
 - Redefine subroutines by assigning new code references to typeglobs.
 - Use the ability to overload locally
- Avoid code duplication
- Use WebGUI API where possible
- Use different namespaces

Example 1

Add a div-tag around the generated HTML of the file upload control `WebGUI::Form::File`.

- Choose custom library directory. For example `/data/swifty/lib`
- Add library directory to `/data/WebGUI/sbin/preload.custom`
- Create package `AMH::Form::File` and redefine `toHtml`
- Only modules names `WebGUI::*` are preloaded. Create package in `WebGUI` namespace to load `AMH::Form::File`.

Example 1

```
package AMH::Form::File;  
use WebGUI::Form::File;  
use strict;
```

```
my $toHtml_org;
```

```
BEGIN {  
    no strict q{refs};  
    no warnings;  
  
    $toHtml_org = {*WebGUI::Form::File::toHtml}{CODE};  
    *WebGUI::Form::File::toHtml = \&toHtml;  
}
```

Example 1

```
sub toHtml {  
  my ($self) = @_;  
  
  return q{<div class="wowFileUpload">  
    . $self->$toHtml_org()  
    . q{</div>};  
}
```

Example 2

Add custom message to payment gateway selection screen.

Code snippet from WebGUI::Shop::Pay

```
sub www_selectPaymentGateway {  
    ...  
    return $session->style->userStyle( $output );  
}
```

Example 2

```
package AMH::Shop::Pay;

use strict;

use WebGUI::Shop::Pay;
use AMH::Wow::Session;

my $www_selectPaymentGateway_org;

BEGIN {
    no strict q{refs}; no warnings;

    $www_selectPaymentGateway_org =
        *{*WebGUI::Shop::Pay::www_selectPaymentGateway}
{CODE};
    *WebGUI::Shop::Pay::www_selectPaymentGateway =
        \&www_selectPaymentGateway;
}
```

Example 2

```
sub www_selectPaymentGateway {
  my ($self) = @_;

  no warnings;
  my $userStyle_org =
    *{*WebGUI::Session::Style::userStyle}{CODE};
  local *WebGUI::Session::Style::userStyle = sub {
    my ($style, $output) = @_;

    $output .= q{<div>}
      . $self->session()->setting()->get(q{Msg})
      . q{</div>};
    return $style->$userStyle_org($output);
  };
  return $self->$www_selectPaymentGateway_org();
}
```

Runtime Environment

- Inspired by WRE
- Ubuntu Linux (8.04)
- mod_perl, mod_proxy setup
- CPAN modules using cpan2dist (<http://search.cpan.org/~kane/CPANPLUS-0.88/bin/cpan2dist>)
- Use /data/WebGUI/sbin/preload.exclude
- Use git (<http://git-scm.com/>)
 - for version control
 - for deployment

Thank you!

Ideas? Questions? Comments?

Henry Tang

<http://www.swiftysite.com>

henry.tang@longtermresults.nl