

Nirvana Templates

Sort of Like QuickTemplates++

Example “Core” output

One giant execute function with:

```
$form .= Xml::openElement( 'form', array( 'method' => 'post', 'action' => $this->getTitle()->getLocalUrl(), 'id' =>
'mw-watchlist-form-namespacesselector' ) );
$form .= '<hr /><p>';
$form .= Xml::label( $this->msg( 'namespace' )->text(), 'namespace' ) . '&#160;';
$form .= Xml::namespaceSelector( $nameSpace, " ) . '&#160;';
$form .= Xml::checkLabel( $this->msg( 'invert' )->text(), 'invert', 'nsinvert', $invert ) . '&#160;';
$form .= Xml::submitButton( $this->msg( 'allpagessubmit' )->text() ) . '</p>';
$form .= Html::hidden( 'days', $values['days'] );
foreach ( $filters as $key => $msg ) {
    if ( $values[$key] ) {
        $form .= Html::hidden( $key, 1 );
    }
}
$form .= Xml::closeElement( 'form' );
$form .= Xml::closeElement( 'fieldset' );
$output->addHTML( $form );
```

Example QuickTemplate

```
class UsercreateTemplate extends QuickTemplate {
```

```
    function execute() {
```

```
        if( $this->data['message'] ) {
```

```
?>
```

```
<div class="<?php $this->text('messagetype') ?>box">
```

```
    <?php if ( $this->data['messagetype'] == 'error' ) { ?>
```

```
        <strong><?php $this->msg( 'loginerror' )?></strong><br />
```

```
    <?php } ?>
```

```
    <?php $this->html('message') ?>
```

```
</div>
```

```
<div class="visualClear"></div>
```

```
<?php } ?>
```

Example Nirvana Template

Controller adds data to itself

```
class HelloWorldController extends WikiaController {  
    // default method=index  
    public function index() {  
        $this->data1 = "Hello";  
        // using normal web request from Global or Context  
        $this->data2 = $this->request->getData("helloparameter", "World");  
        $this->msg1 = 'hello-message';  
    }  
}
```

Calling "render" on a controller automatically uses default Template File: HelloWorld_Index

```
<ul>  
    <li>Hello World</li>  
    <li><?php echo $data1 ?> <?php echo $data2 ?> </li>  
    <li><?php echo wfMessage($msg1)->text(); ?> </li>  
</ul>
```

To render this from another template:

```
<?php echo $this->renderView('HelloWorld', 'Index') ?>
```

Example of real usage in skin

Examples from our main Oasis "Body" template

```
<?= $app->renderView('GlobalHeader', 'Index') ?>
<?= $app->renderView('Notifications', 'Confirmation') ?>
<?= $app->renderView('Ad', 'Top') ?>
...
<div id="WikiaArticleBottomAd" class="noprint">
    <?= $app->renderView('Ad', 'Index', array('slotname' => 'PREFOOTER_LEFT_BOXAD')) ?>
    <?= $app->renderView('Ad', 'Index', array('slotname' => 'PREFOOTER_RIGHT_BOXAD')) ?>
</div>
```

Different styles of optionally including things:

```
<?= empty($wg->WikiaSeasonsPencilUnit) ? " : $app->renderView('WikiaSeasons', 'pencilUnit', array()); ?>

<? if( $wg->EnableWikiaBarExt ): ?>
    <?= $app->renderView('WikiaBar', 'Index'); ?>
<? endif; ?>
```

The “good”

We “inverted” the template rendering path

- Templates include the Controllers which then emit HTML so if you don't include it, no work is done

Does allow HTML in controller (although it is bad style)

- Doesn't use Xml object anywhere (yay)

Can easily (and optionally) drop extension output into skin.

We have hundreds of extensions gated by global vars

All Controller methods can return JSON and become Ajax calls (easy access to data)

Simplify usage of global vars

- Doesn't require duplicate/redundant temp vars or “global” statements, just use them

- `$this->wg->Server; // controller`

- `$wg->Server; // template`

The bad/ugly

PHP Short tags and short array syntax

Mostly PHP only. Allows mustache templates (but those are very limited)

Doesn't really deal with Javascript UI at all, only server side

Output escaping and safety is still up to the developer

Includes a simple "MVC" framework which people might not want

A wrapper for mediawiki global vars? (we also had one for functions but took it out)

So it's basically like, QuickTemplates++

Simple but probably not ambitious enough?