

Magento Developers Paradise

March 20th, 2013

Changes Overview of View Layer in Magento 2

Anton Makarenko

Sr. Software Engineer, Magento 2



Contents

1. Problem definition for Magento 2 view layer
2. File system changes: modules, themes
3. Logical changes: theme and view configuration
4. Logical changes: blocks and containers
5. Plans for further evolution

Goals for developer and user

1. PROBLEM DEFINITION FOR MAGENTO 2 VIEW LAYER

Goals: Developer Experience

- Freedom in creating and structuring themes
- Flexibility in implementing layout, more granularity
- Less incentive to duplicate code

“True Modularity” Requirements

- All module files must be in one directory (of the module)
- A module must guarantee its operability, rather than depend on a theme
- Dependency on a module must be clearly visible

Goals: User Experience

- Simpler configuration of design in admin
- Less concepts to learn (fallback, skin...)
- Failsafe mechanisms for layout updates entered by the user
- Visual design editor as a standard feature (comes from Magento Go)

Improvements in modules and themes

2. FILE SYSTEM CHANGES

Directories Restructuring

Magento 1

```
___/  
|___/app/design/...  
|___/errors  
|___/js  
|___/media  
|___/skin  
|-- index.php
```

Magento 2

```
___/  
|___/app/design/...  
|___/pub  
| |___/errors  
| |___/lib  
| |___/media  
| |___/static  
| |-- index.php  
|-- index.php
```


View Files in Modules

```
___/app/code/Mage/Catalog/view
  |___/adminhtml/...
  |___/frontend
    |___/category/...
    |___/product
      |  |-- price.phtml
      |  |-- view.phtml
    |-- layout.xml
    |-- widgets.css
```

Templates and Layouts Relocation

```
__/_/app/design/frontend/base/default  
|__/_/layout/catalog.xml  
|__/_/template/catalog/product/view.phtml
```

```
__/_/app/code/Mage/Catalog/view  
|__/_/frontend  
|__/_/product/view.phtml  
|-- layout.xml
```

Static Files Relocation

```
___/skin/frontend/base/default  
|___/js/opcheckout.js
```

```
___/app/code/Mage/Checkout/view  
|___/frontend  
|___/js/opcheckout.js
```

Theme File Structure

```
__/  
|__/_/Mage_Catalog  
| |__/_/images/slider_btn_zoom_in.gif  
| |__/_/product/view.phtml  
|__/_/Mage_Page/...  
|-- theme.xml  
|-- view.xml
```

Fallback Scheme: Templates and Layouts

```
___/app/design/<area>/<current_theme>  
|___/Mage_Catalog/product/view.phtml
```

```
___/app/design/<area>/<parent_theme>  
|___/Mage_Catalog/product/view.phtml
```

```
___/app/code/Mage/Catalog/view  
|___/<area>/product/view.phtml
```

Fallback Scheme: Static Content

app/design/<area>/<theme>/Mage_Page/locale/de_DE/logo.gif

app/design/<area>/<theme>/Mage_Page/logo.gif

app/design/<area>/<parent>/Mage_Page/locale/de_DE/logo.gif

app/design/<area>/<parent>/Mage_Page/logo.gif

app/code/Mage/Page/view/<area>/locale/de_DE/logo.gif

app/code/Mage/Page/view/<area>/logo.gif

pub/lib/logo.gif

How Static Content Gets Into a Public Directory

- Developer mode: execute fallback every time and copy to `pub/static/...`
- Default mode: execute fallback and copy to `pub/static/...` only first time
- Production mode: build a fully qualified path without checking anything

3. LOGICAL CHANGES: THEME AND VIEW CONFIGURATION

Themes that Make More Sense

Magento 1

- Skins:
 - base: default
 - default: blank, blue, default, french, german, iphone, blue
 - enterprise: default, iphone
 - pro: default
- Themes:
 - base/default
 - default: blank, default, iphone, modern

Magento 2

- Themes:
 - blank
 - demo
 - demo blue
 - iphone
 - modern
 - fixed design
 - fluid design

Theme Management System

- All themes are registered in DB
- “Virtual” themes:
 - Are created by the user, they don’t exist “out of the box”
 - Have user-uploaded assets: CSS and JavaScript
 - The user assets do not participate in theme inheritance

theme.xml

- A user-friendly theme name
- Version
- Parent theme
- Preview images, etc...

view.xml

- `view.xml` in modules and themes:
 - `<module>/etc/view.xml`
 - `<theme>/view.xml`
- Declares variables, available in blocks/templates
- First, the `view.xml` of all modules are merged; then, `view.xml` of current theme is added

4. LOGICAL CHANGES: BLOCKS AND CONTAINERS

Layout Improvement Goals

- Increase flexibility to fit needs of visual design editor – display and manipulate elements on a page in WYSIWYG-style
- Allow more granular layouts without necessity of instantiating more objects in process of layout generation

Illustration: Block Instances

Zolof The Rock And Roll Destroyer: LOL Cat T-shirt

[Email to a Friend](#)

★★★★☆ 3 Review(s) | [Add Your Review](#)

Availability: In stock

\$13.50

Quick Overview

Printed on American Apparel Classic style 5495 California t-shirts.

*Size * Required Fields

Choose an Option...

*Color

Choose an Option...

\$13.50

Qty: 1 **Add to Cart**

[Add to Wishlist](#) | [Add to Compare](#)

Double click on above image to view full picture

MORE VIEWS

Three small thumbnail images showing different views of the t-shirt.

Illustration: Child Block

```
<div class="account-login">
  <div class="page-title">
    <h1><?php echo $this->__( 'Login
or Create an Account' ) ?></h1>
  </div>
  <?php echo
$this->getChildHtml( 'form.login.extra' )
  ?>
```

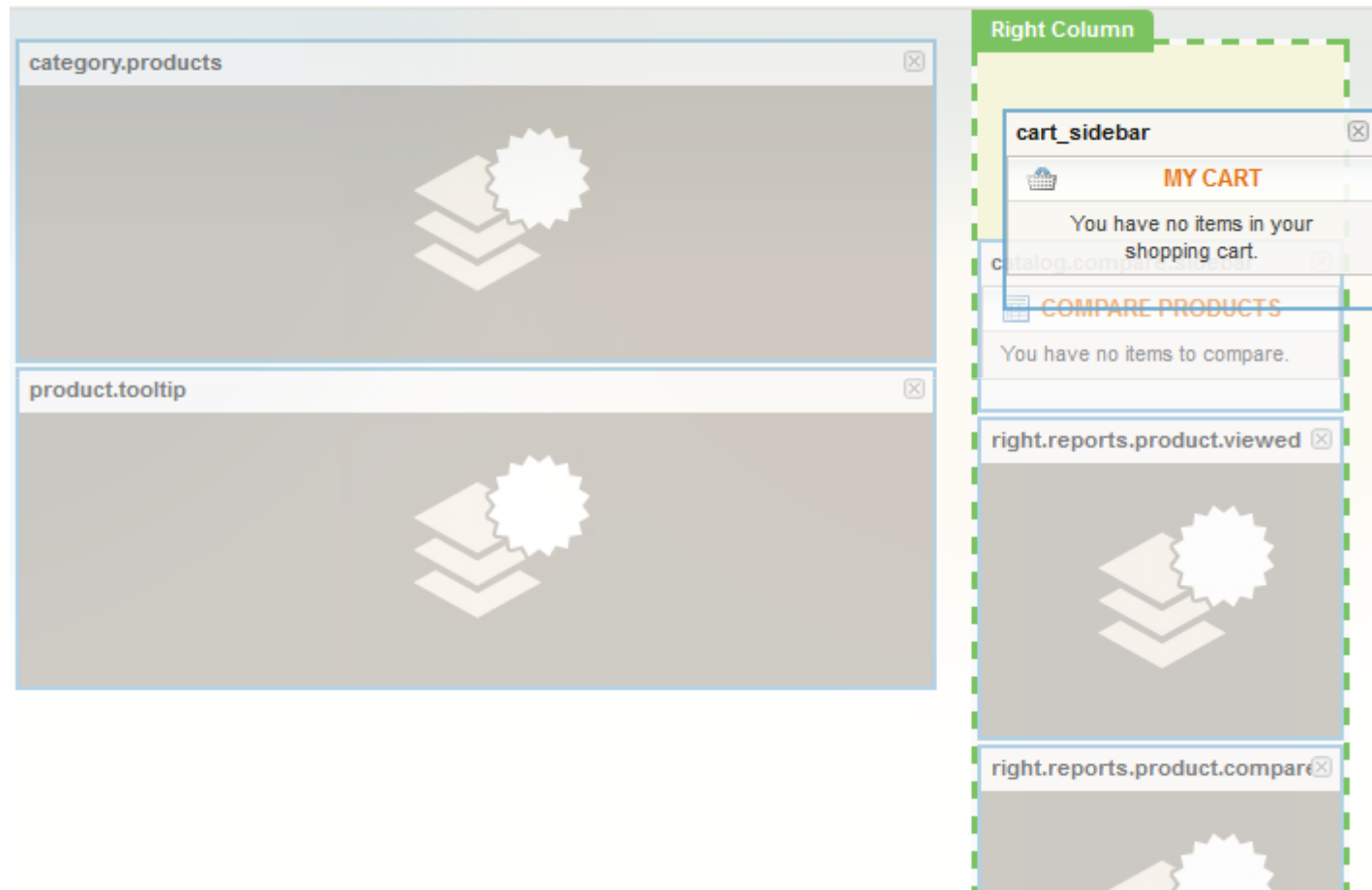

Containers

```
<container name="cms.wrapper"  
  label="CMS Content Wrapper"  
  htmlTag="div"  
  htmlClass="std">  
  <block type="Mage_Cms_Block_Page"  
    name="cms_page" />  
  <container name="..." label="..." />  
</container>
```

What Becomes Possible with Containers and Blocks

- Hierarchical structure (skeleton) of entire page can be implemented using containers
- Behavior of container is predictable, which allows moving elements between containers
- Container can be broken down into pieces by the user, not only developer
- Block obtains new meaning of an indivisible unit of page. It can be broken down only by the developer

Containers in VDE



Moving Elements

```
<move element="name.in.layout"  
destination="new.parent.name"  
as="new_alias"  
after="name.of.element.after"  
before="name.of.element.before" />
```

Page Types

```
<cms_page_view
  type="page"
  parent="default"
  translate="label">
  <label>CMS Pages (All)</label>
```

- Some of layout handles are formalized as page types
- List of page types is used in widgets and visual design editor

5. FURTHER PLANS FOR MAGENTO 2 VIEW LAYER

Formalized Layout

- XML-schema for layout files
- No code execution from layout
- Refined definition of page types and their algorithm
- Definition of “layouts” (1-, 2-, 3-column, etc) as set of containers, rather than templates

Further Evolution of Containers

- Ability for developer to provide limitations for container:
 - Lock manipulation with child elements
 - Lock parent (important for form elements)
- Additional container attributes for visual editing features

Refactoring of Blocks

- Ability to reuse an instance of block for multiple structural elements (“flyweight” pattern)
- Considering approach of “view models” (aka “data sources”)
- Different types of templates (alternatives to PHTML)

Further Improvements of Themes

- Optimization to reduce code duplication
- Removal of Magento branding from themes intended for extending
- Extended metadata in themes (theme.xml)

Summary

- Modular and more logical file structure
- Simpler and more flexible themes
- Better granularity in layout
- Visual design editor as a core feature

Thank you

anton@magento.com

<http://github.com/magento/magento2>

<http://wiki.magento.com/display/MAGE2DOC/>