Pro Magento

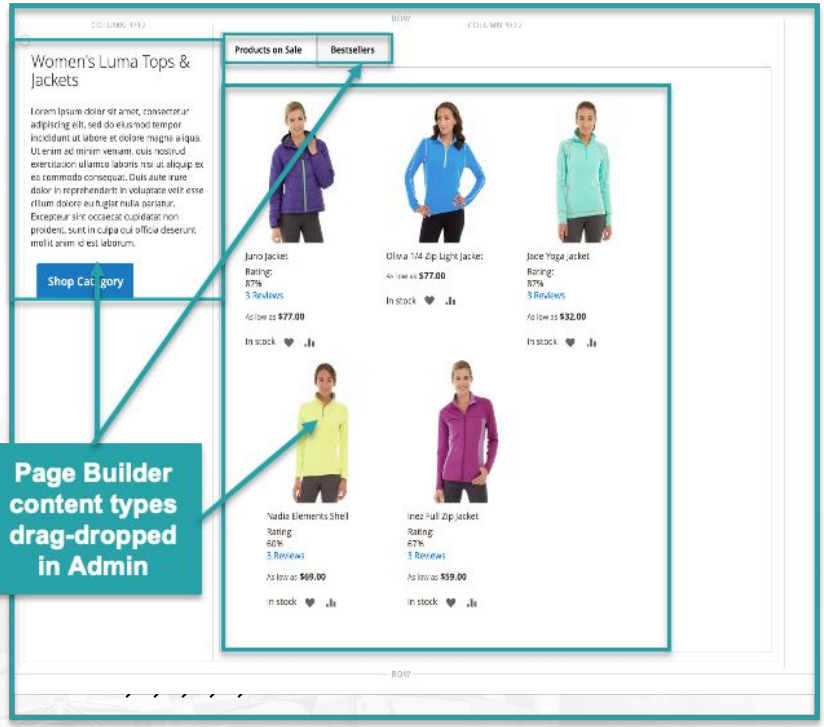
PRO Magento
Meetup #3

How does Magento Page Builder work?

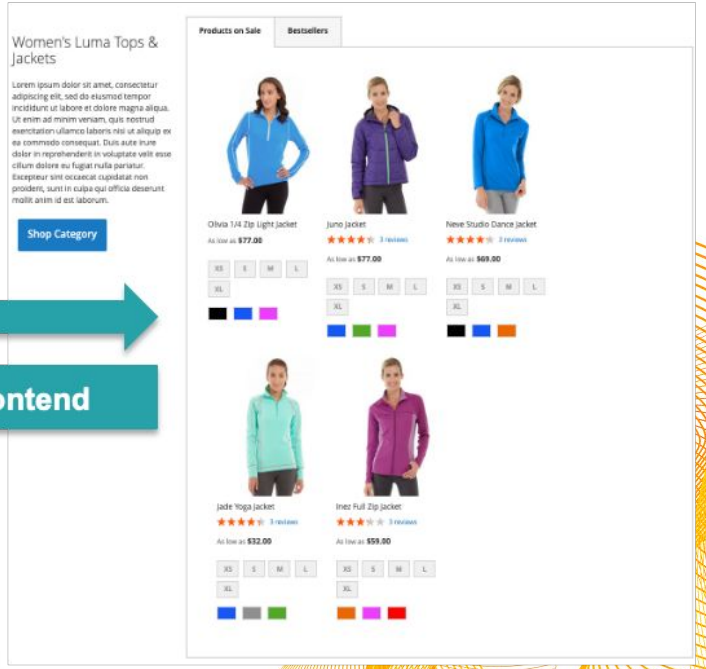
Paras Sood

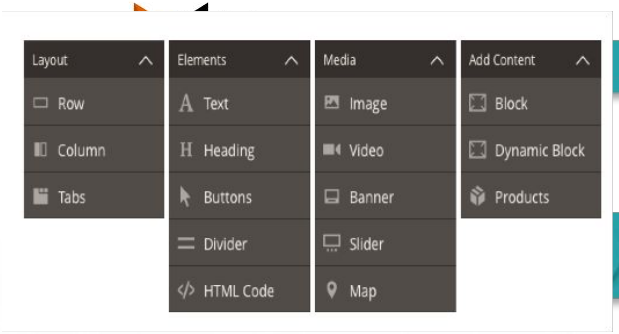
Senior Software Engineer, Adobe

What is Page Builder?

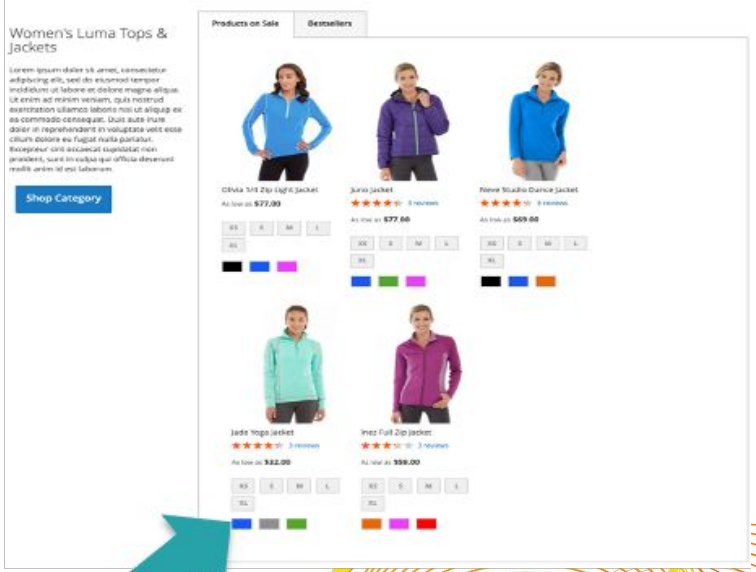


Frontend





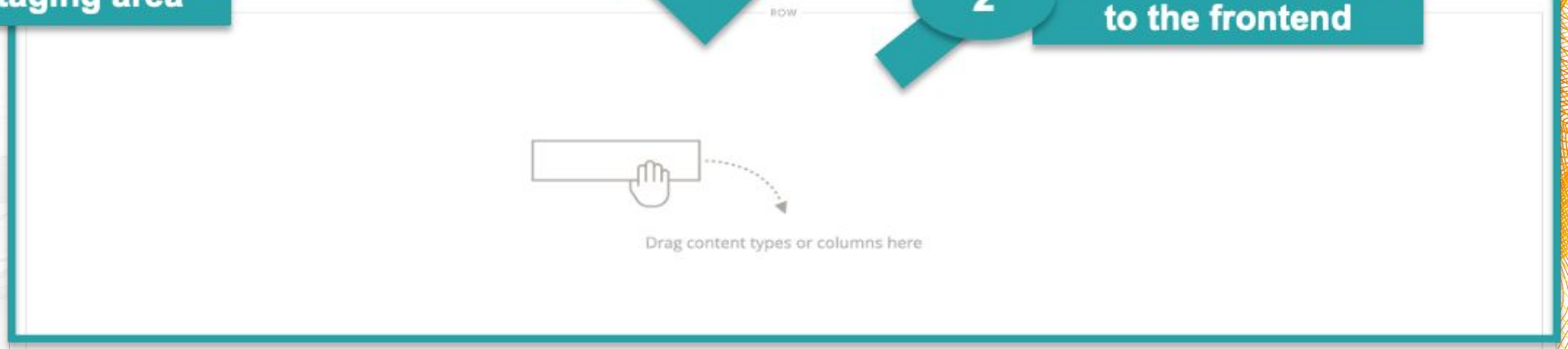
Drag and Drop



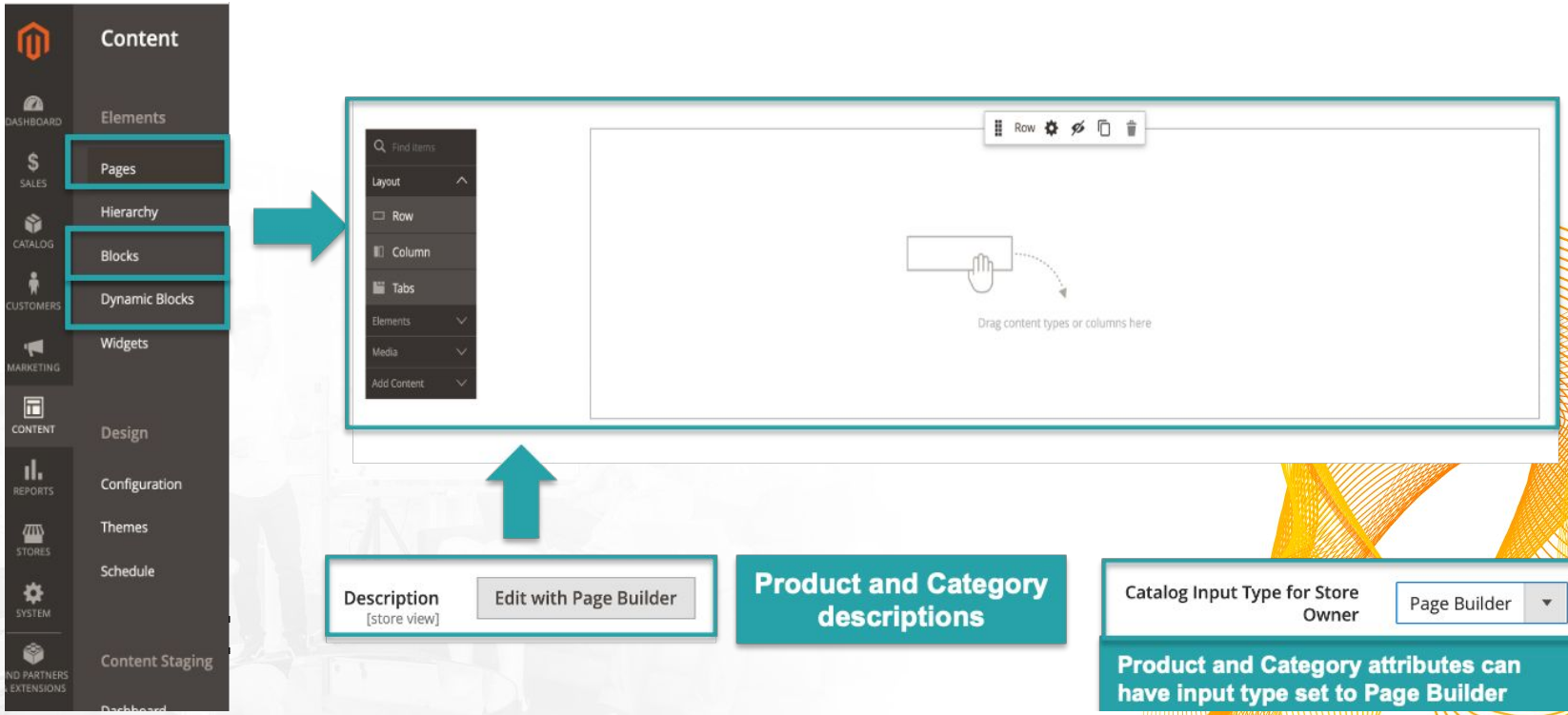
Staging area

2

Preview and Publish to the frontend



Areas Page Builder is available to be used in

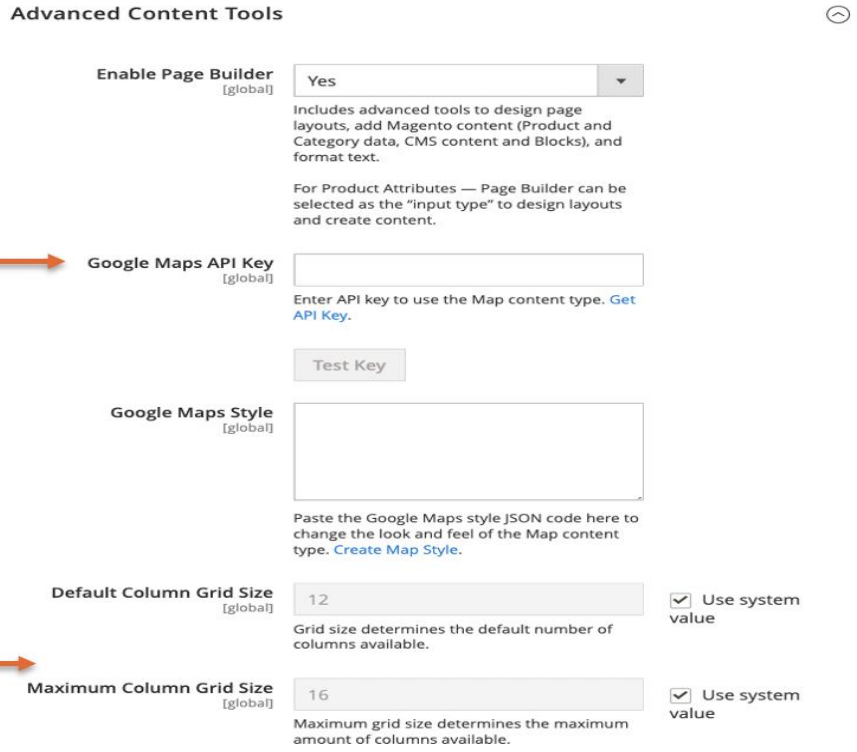


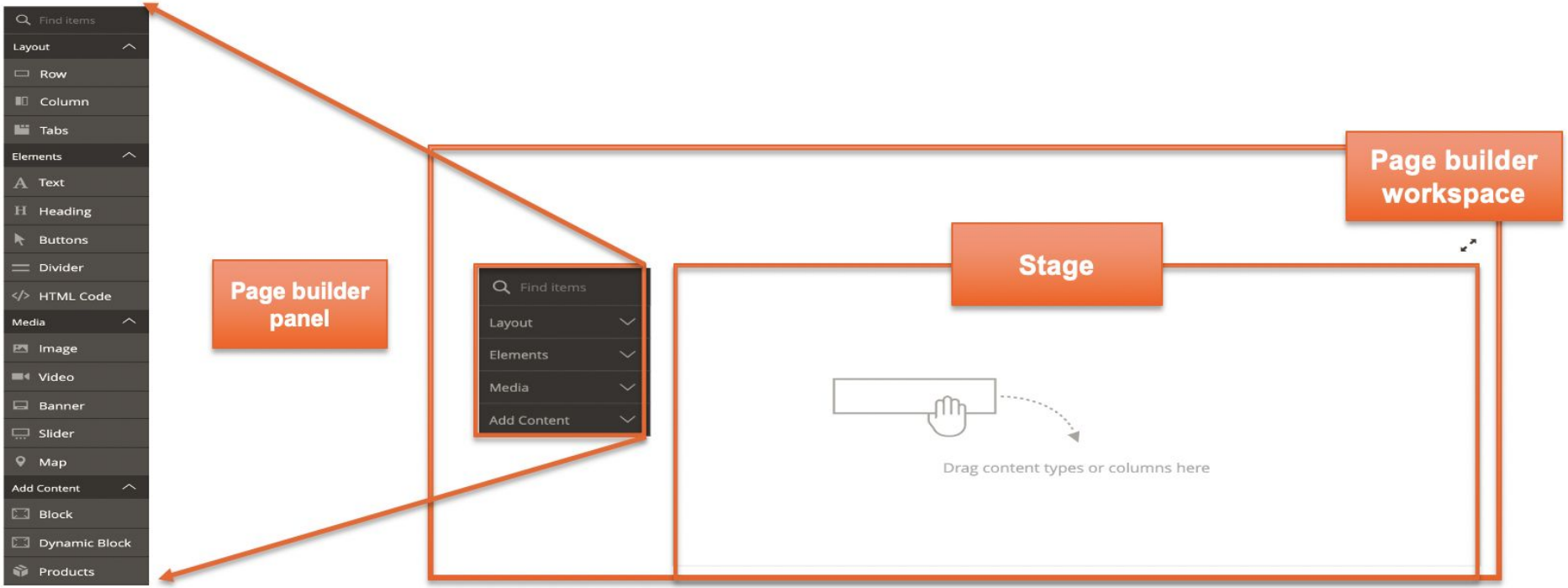
Settings at: Stores > Configuration > General > Content Management > Advanced Content Tools

Google Maps API key and style – to add google maps widget to content via the Page Builder

Grid guideline configuration:

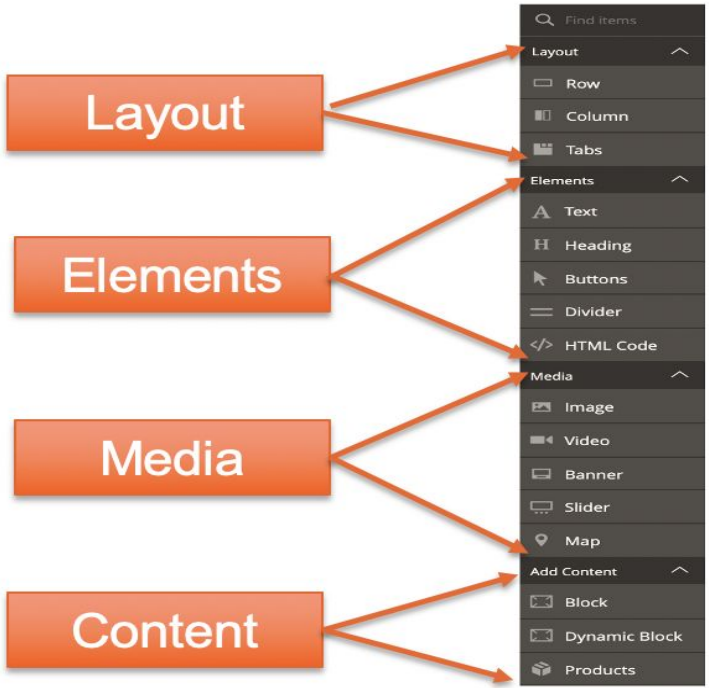
- **Default Column Grid Size** - default number of columns in grid
- In the **Maximum Column Grid Size** - largest number of columns in the grid



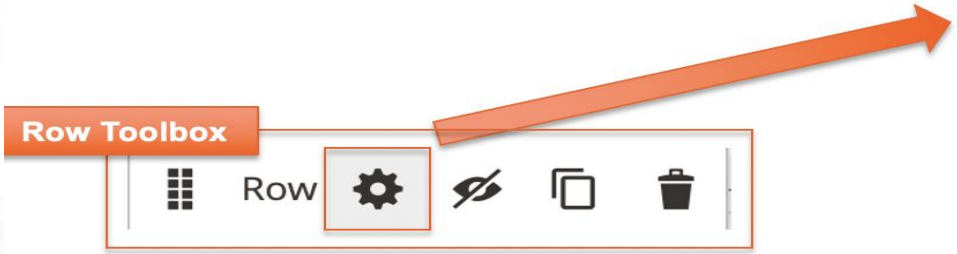

Find items
Layout
Row
Column
Tabs
Elements
Text
Heading
Buttons
Divider
HTML Code
Media
Image
Video
Banner
Slider
Map
Add Content
Block
Dynamic Block
Products
Page builder panel
Find items
Layout
Elements
Media
Add Content
Stage
Page builder workspace
Drag content types or columns here

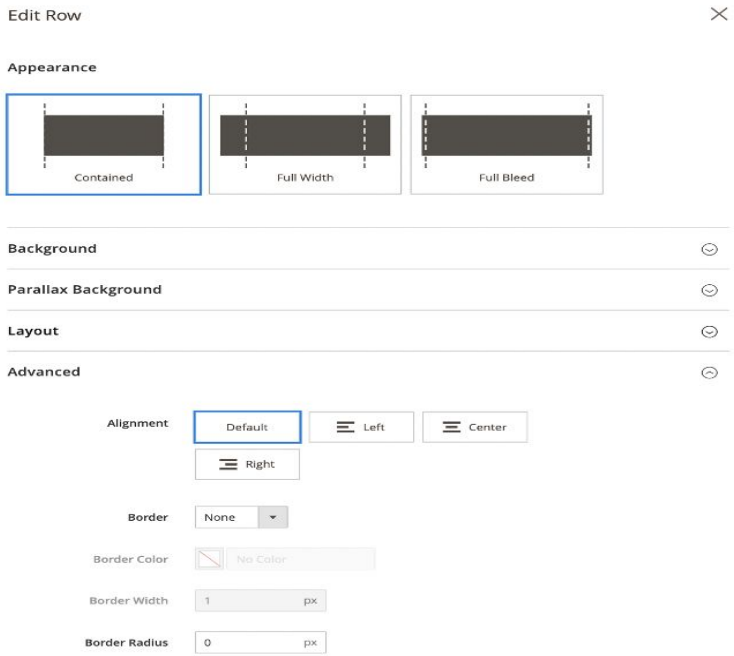
ProMagento

The Page Builder panel is organized into 4 main content types that can be dragged to the Stage



Content Types

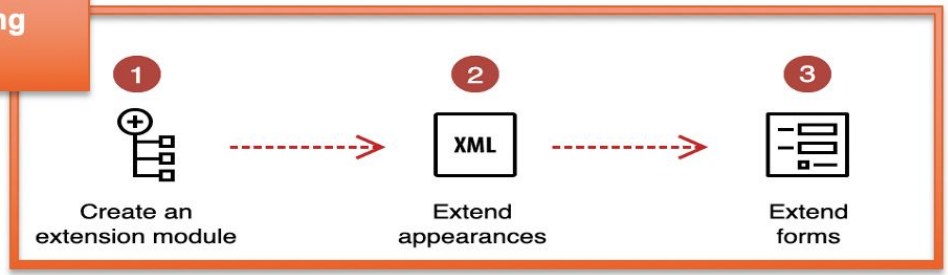




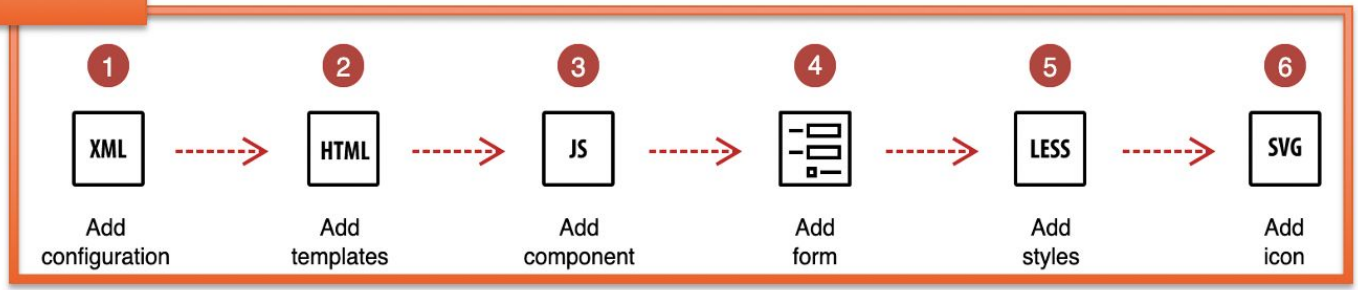
ProMagento

Customizing Page Builder

Extend an existing content type



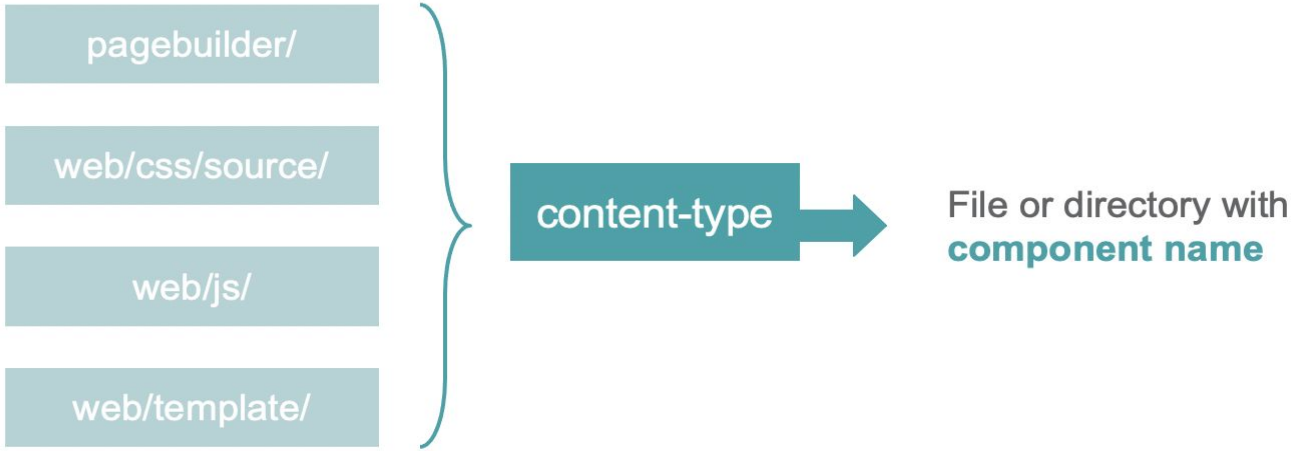
Adding a new content type



Pro Magento

How the Design System impacts the theme?

Page Builder theme has always the same organization:



ProMagento

What about styles organization?

Page Builder is a module, its main style file is **web/css/source/_module.less**

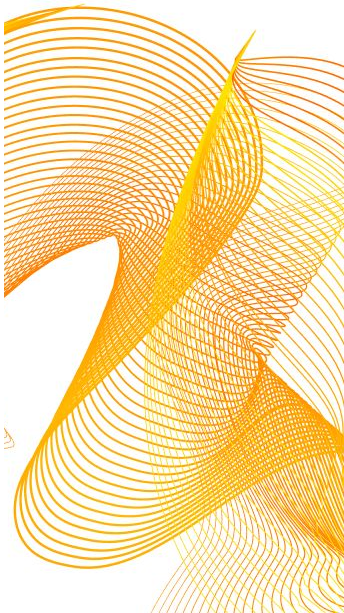
Use this file to merge your content type styles into the pub/static output with all the other module styles

```
// _module.less

[...]

@import '_icons.less';
@import '_layout.less';
@import 'slick/_slick.less';
@import 'content-type/_import.less';

[...]
```

_import.less

Orchestrate the merge of all your content type styles

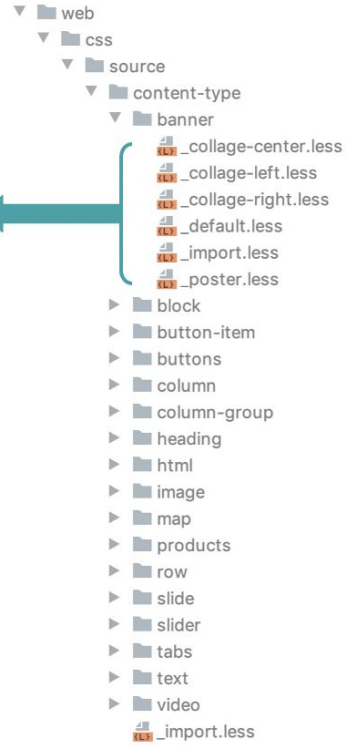
Each content type has its own _import.less to merge each appearance styles



Example: Banner

```
// content-type/_import.less

@import 'banner/_import.less';
@import 'block/_import.less';
@import 'buttons/_import.less';
@import 'button-item/_import.less';
@import 'column/_import.less';
@import 'column-group/_import.less';
@import 'heading/_import.less';
@import 'html/_import.less';
@import 'image/_import.less';
@import 'map/_import.less';
@import 'products/_import.less';
@import 'row/_import.less';
@import 'slide/_import.less';
@import 'slider/_import.less';
@import 'tabs/_import.less';
@import 'text/_import.less';
@import 'video/_import.less';
```

```
// content-type/banner/_import.less

@import './_default.less';
@import './_poster.less';
@import './_collage-center.less';
@import './_collage-left.less';
@import './_collage-right.less';
```



ProMagento

Where can you find a part of the Page Builder theme?

module-page-builder (core)

view/adminhtml
→ all default admin theme
→ all default storefront render templates

view/frontend
→ all default storefront styles

<Vendor>/<theme>

web/css/source
→ override of default storefront styles

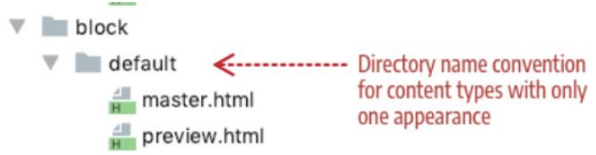
<Vendor>/<module>

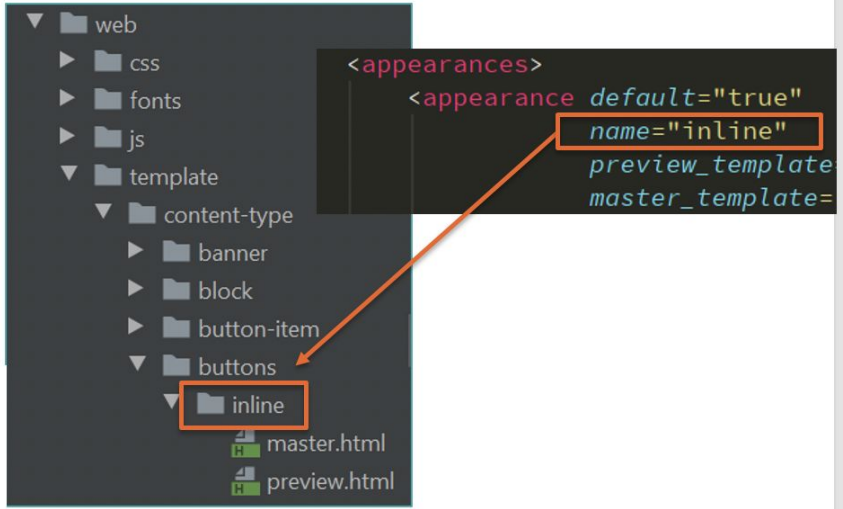
view/adminhtml
→ customization of default admin theme
→ customization of default storefront render templates
→ styles of admin new extension or content type

view/frontend
→ extend of all default storefront styles
→ styles of storefront new extension or content type

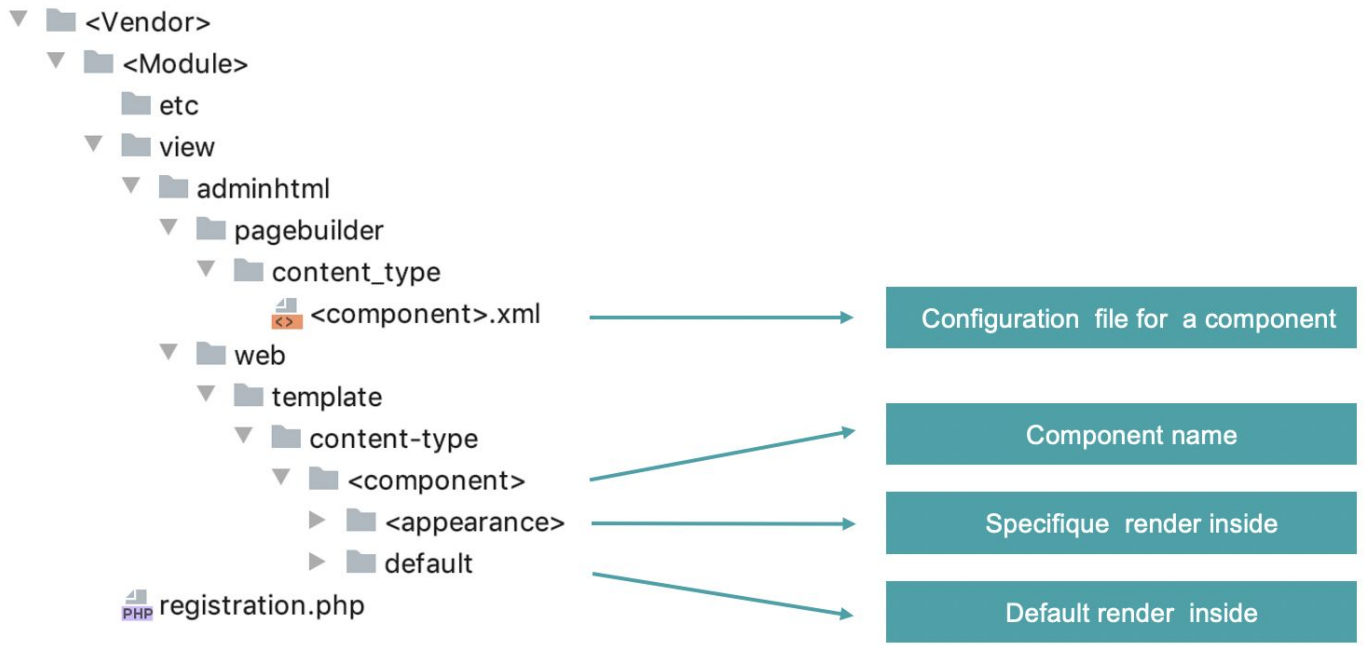
Creating admin templates

- `preview.html` vs `master.html`
- Directory name must match appearance name
- Use `default` for content types with only one appearance





How to Override



Pro Magento

Example: banner.xml

```
// view/adminhtml/pagebuilder/content_type/banner.xml

<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_PageBuilder:etc/content_type.xsd">
  <type name="banner">
    <appearances>
      <appearance name="collage-right"
            master_template="Module_PageBuilder/content-type/banner/collage-right/master">
      </appearance>
    </appearances>
  </type>
</config>
```

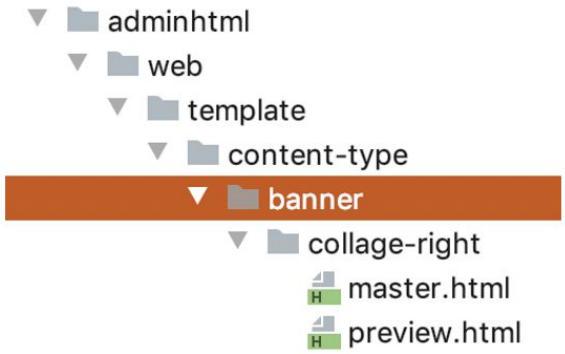
Component Name

Appearance Name

Template path

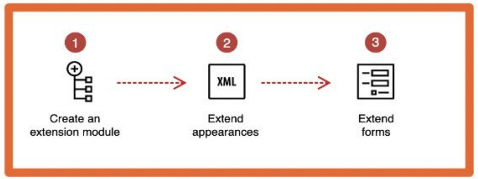
ProMagento

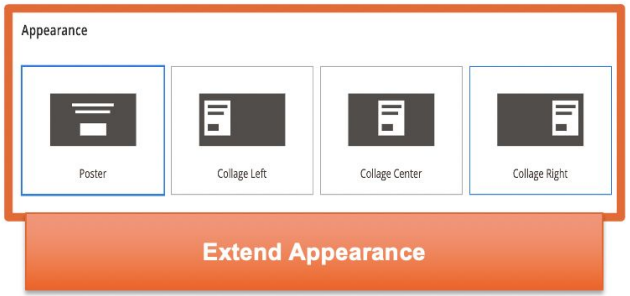
Example: template for banner on your storefront

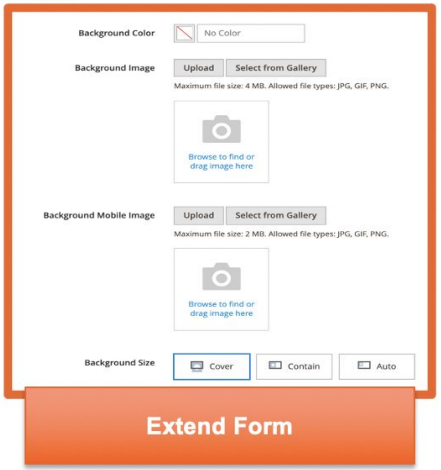


By convention
the storefront render
is always
master.html

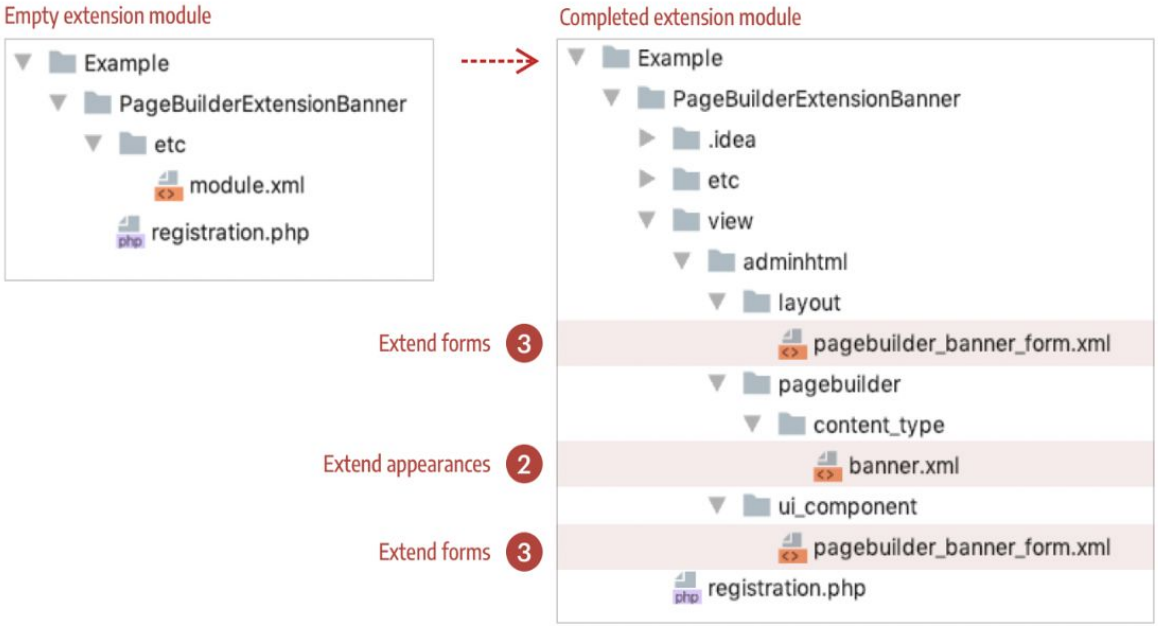
Extending a content type







Step 1 – Create an extension module



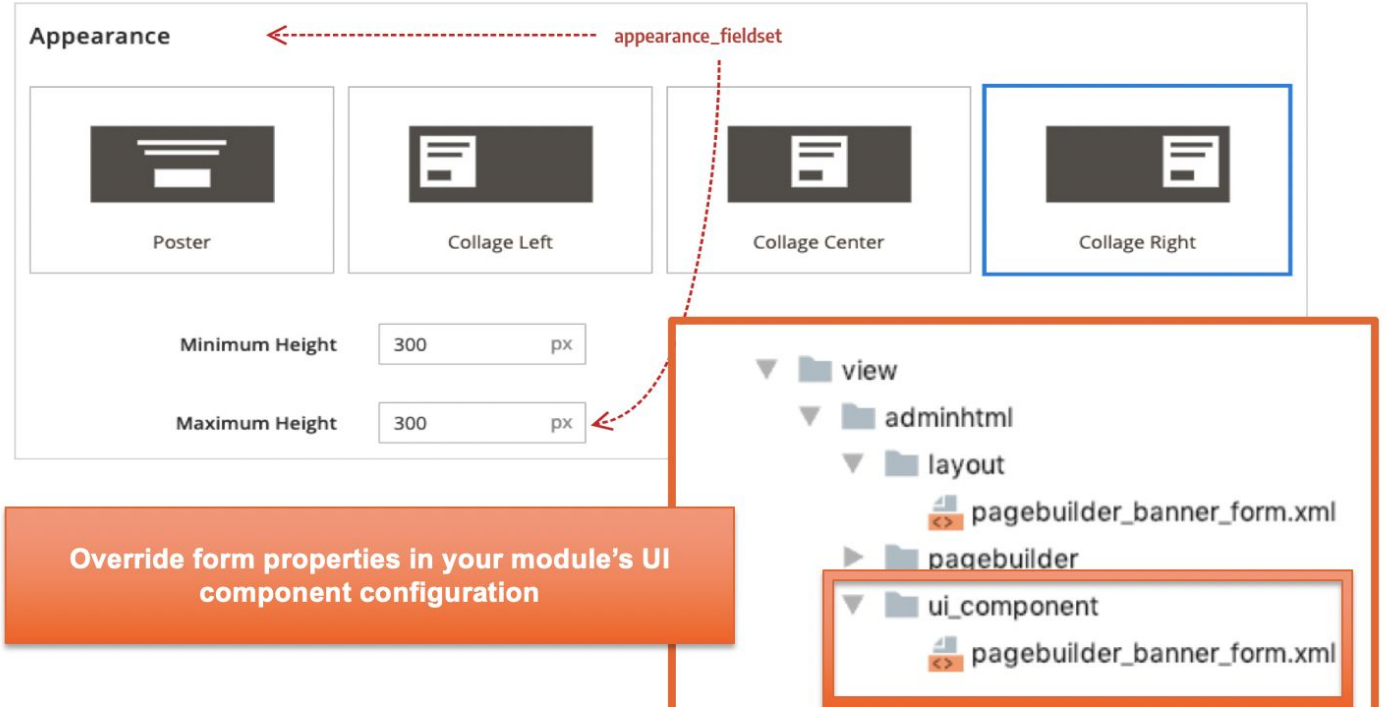
Step 2 – Extend Appearances

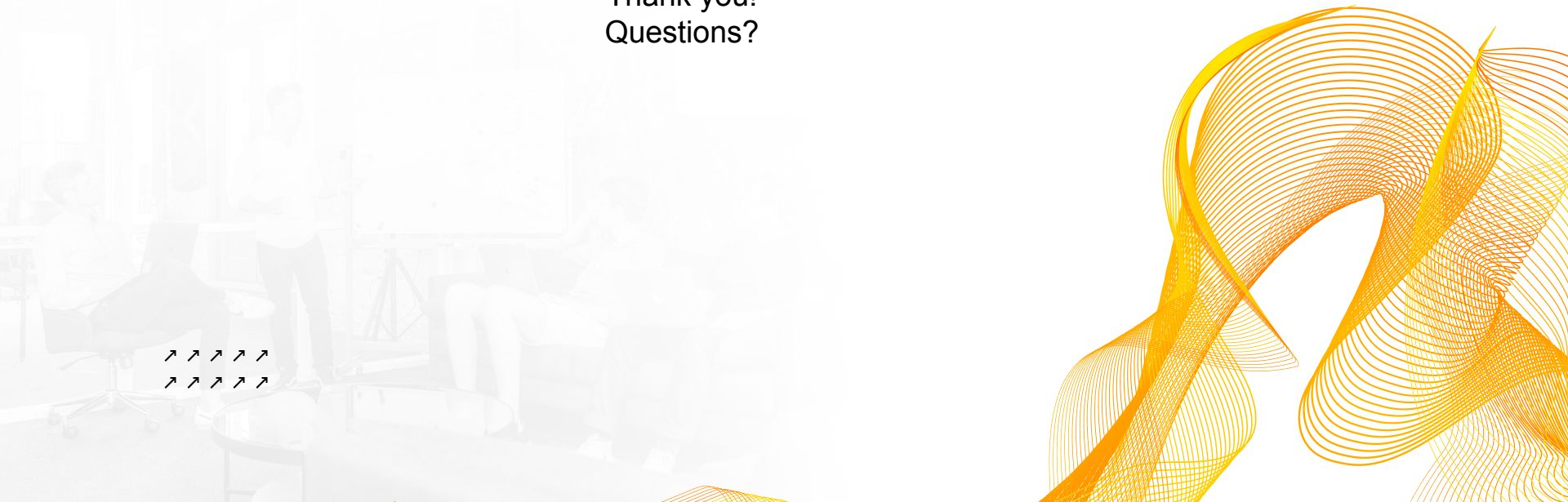
```xml
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_PageBuilder:etc/content
    <type name="banner">
        <appearances>
            <appearance name="collage-left">
                <elements>
                    <element name="wrapper">
                        <style name="max_height" source="max_height" converter="Magento_PageBuilder/js/converter/style/remove-px"/>
                    </element>
                </elements>
            </appearance>
            <appearance name="collage-right">
                <elements>
                    <element name="wrapper">
                        <style name="max_height" source="max_height" converter="Magento_PageBuilder/js/converter/style/remove-px"/>
                    </element>
                </elements>
            </appearance>
            <appearance name="poster">
                <elements>
                    <element name="wrapper">
                        <style name="max_height" source="max_height" converter="Magento_PageBuilder/js/converter/style/remove-px"/>
                    </element>
                </elements>
            </appearance>
            <appearance name="collage-centered">
                <elements>
                    <element name="wrapper">
                        <style name="max_height" source="max_height" converter="Magento_PageBuilder/js/converter/style/remove-px"/>
                    </element>
                </elements>
            </appearance>
        </appearances>
    </type>
</config>
```

Like layout files, Page Builder configuration files are merged

Override appearance properties in your module's page builder config files

Step 3 – Extend Forms



Override form properties in your module's UI component configuration

Thank you!
Questions?