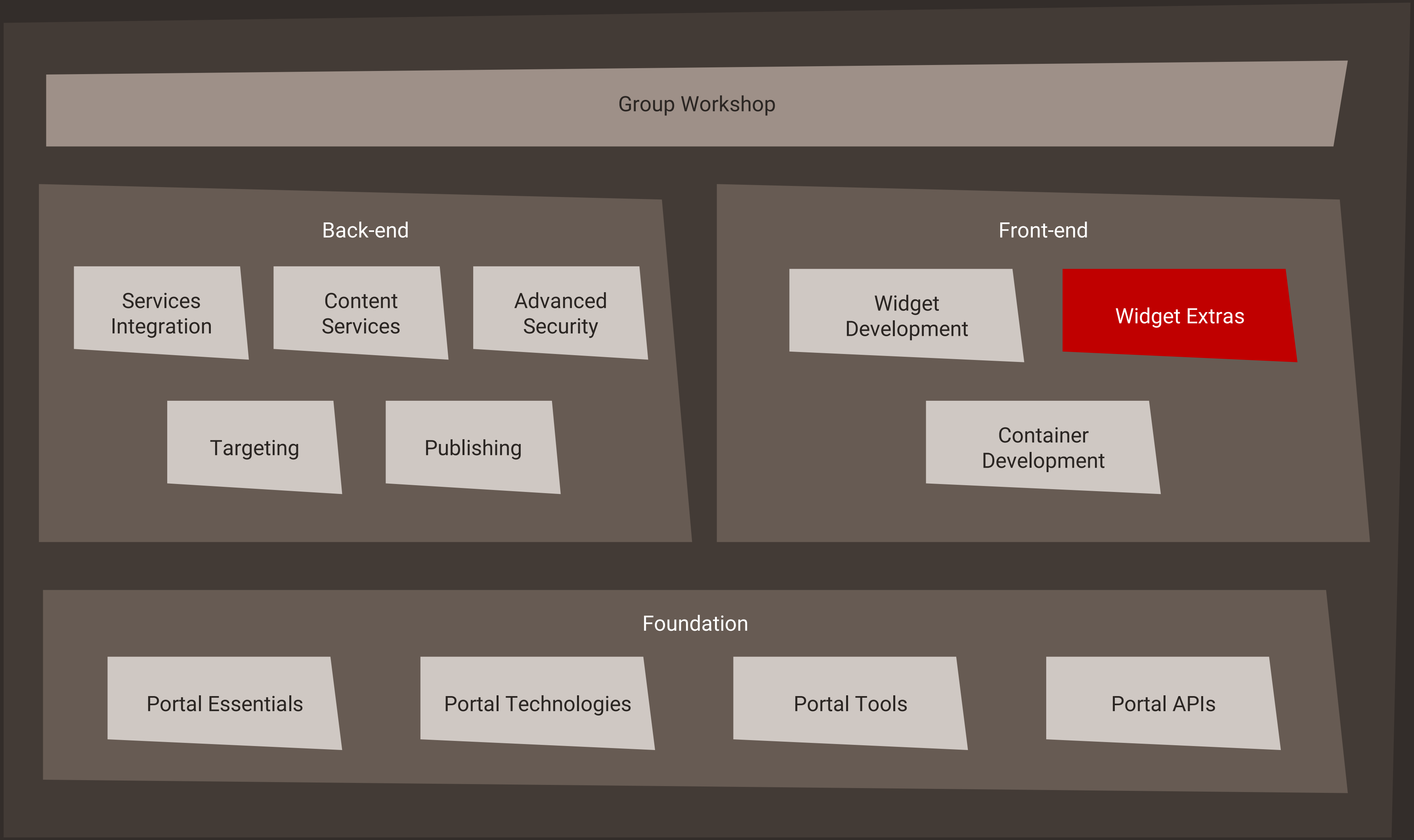
BACKBASE

Widget Extras

Focus Area: Front-end

Training Overview



Group Workshop

Back-end

Services Integration

Content Services

Advanced Security

Targeting

Publishing

Front-end

Widget Development

Widget Extras

Container Development

Foundation

Portal Essentials

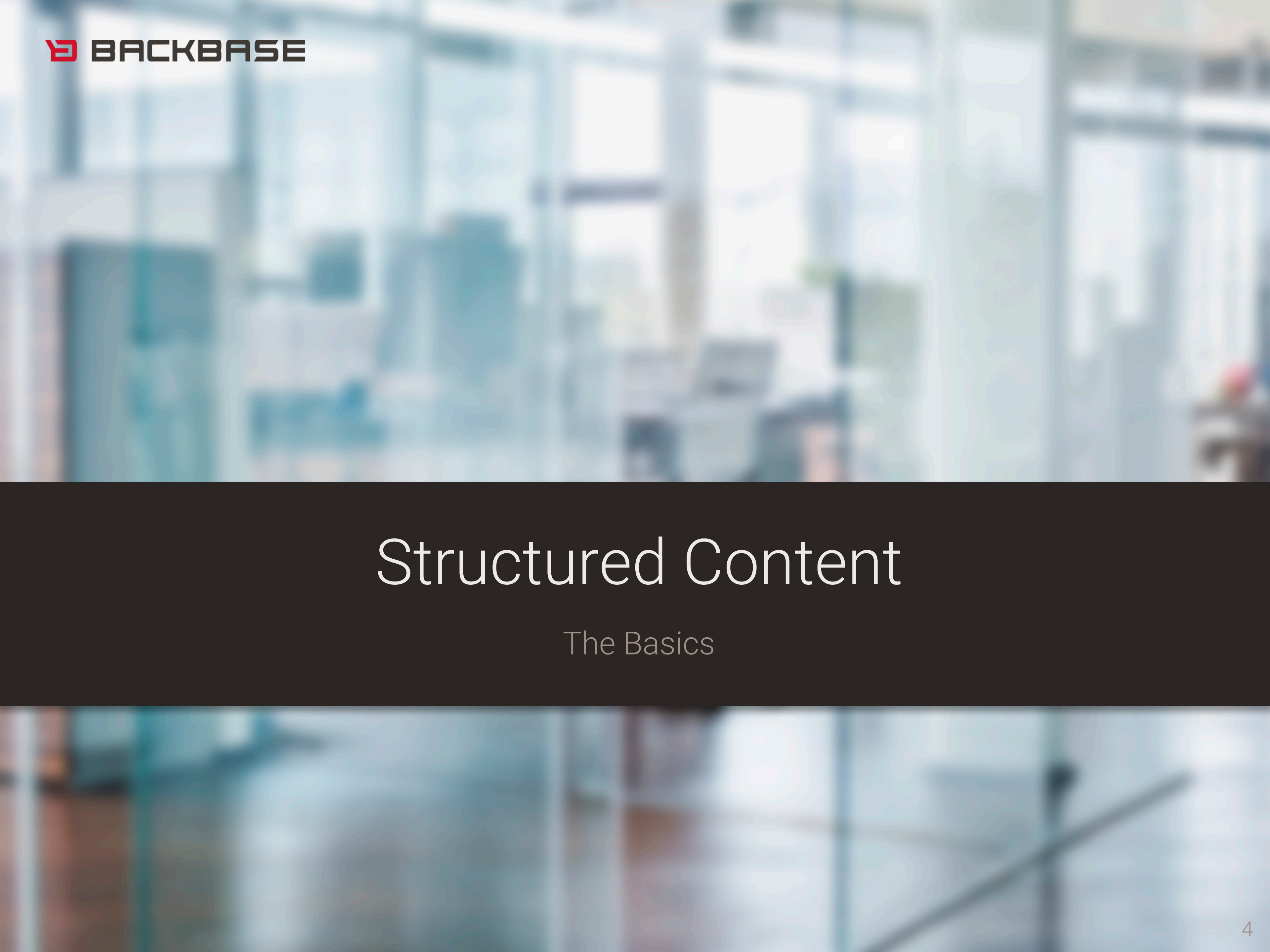
Portal Technologies

Portal Tools

Portal APIs

Objectives

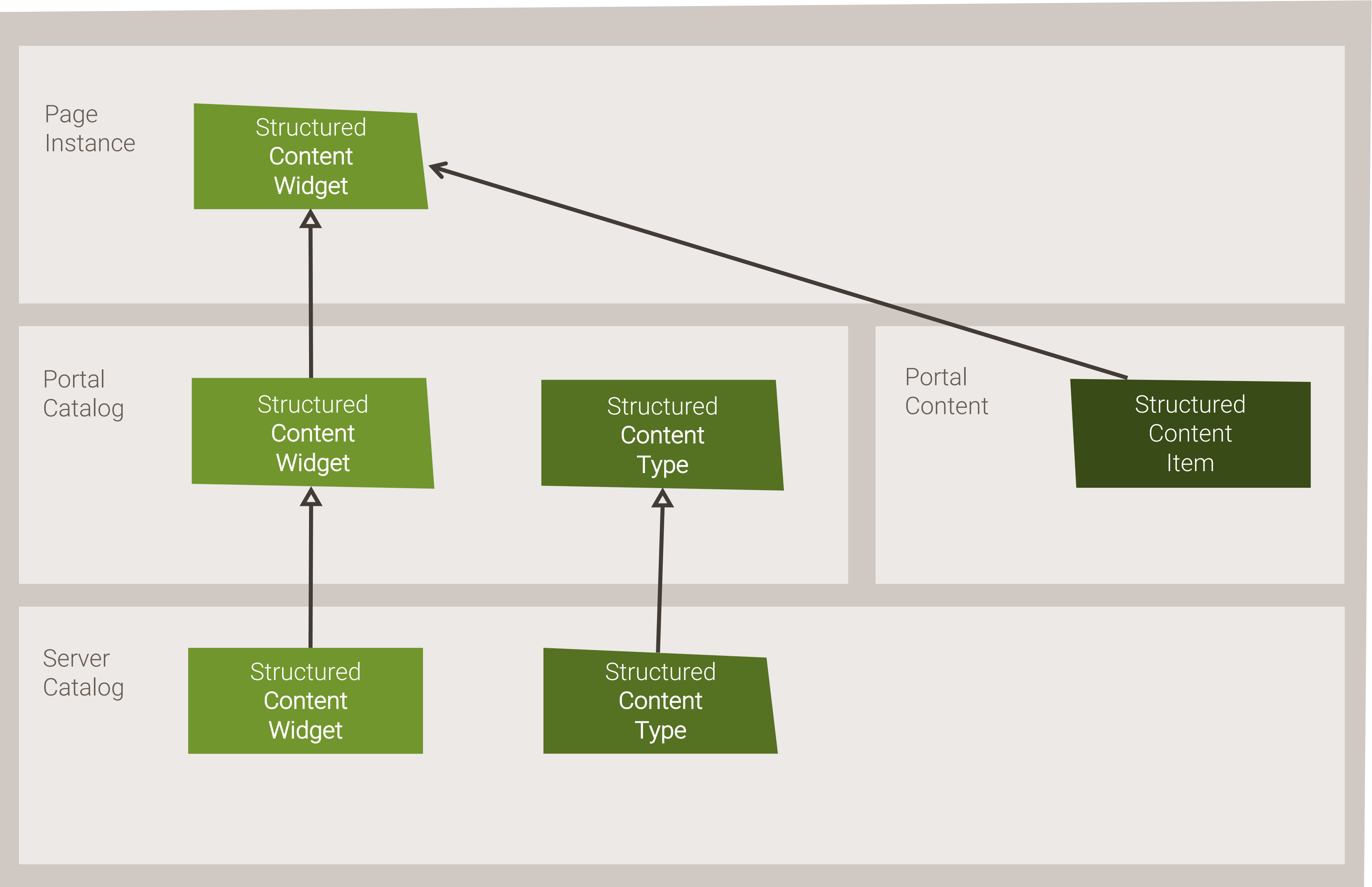
1. Structured Content
   - Structured Content Types
   - Structured Content Items
   - Structured Content Widget
2. ICE Templates
3. Widget Chromes



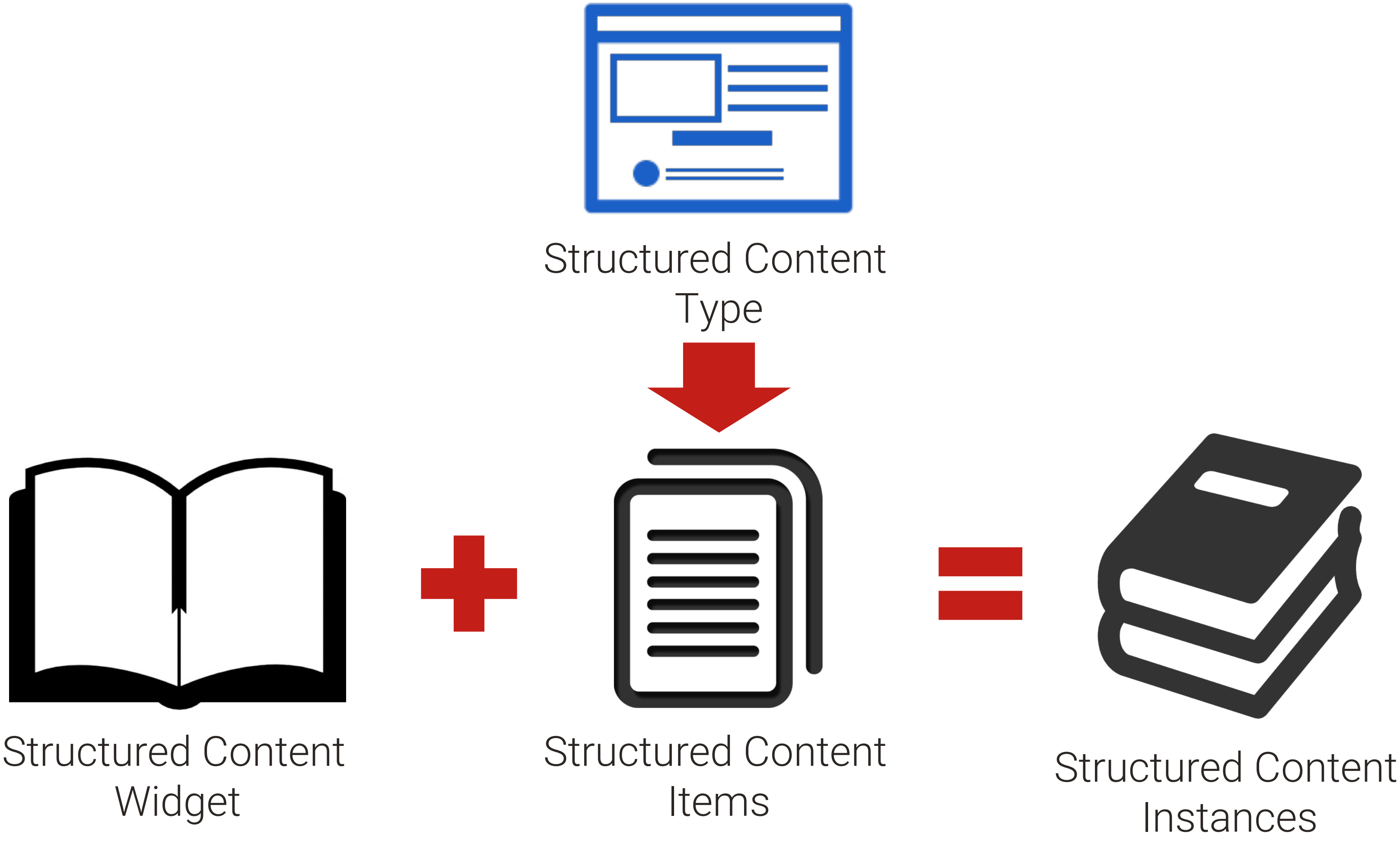
Structured Content

The Basics

Overview


Page Instance
Structured Content Widget
Portal Catalog
Structured Content Widget
Structured Content Type
Portal Content
Structured Content Item
Server Catalog
Structured Content Widget
Structured Content Type

Structured Content



Structured Content Type

Structured Content Widget

Structured Content Items

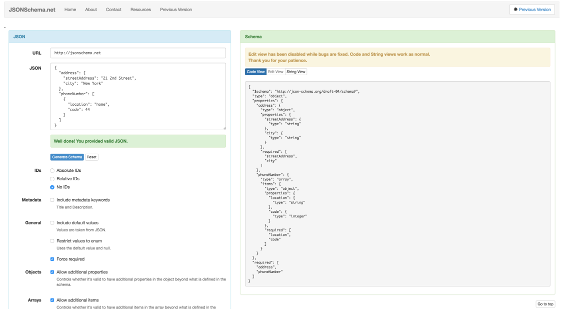
Structured Content Instances

Structured Content Type:

- It's a template for your data.
- Content Type are created using the JSON Schema Standard.
- JSON Schema can be used to:
    - Add validation to data.
    - Set the type of data to be stored.
        - Array.
        - Object
        - Image
        - Another Structured Content Item.
    - Set the type of input
    - Set default values

Structured Content Type

http://jsonschema.net/



URL: http://jsonschema.net

JSON:

```
{
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York"
  },
  "phoneNumber": [
    {
      "location": "home",
      "code": 44
    }
  ]
}
```

Well done! You provided valid JSON.

IDs
○ Absolute IDs
○ Relative IDs
● No IDs

Metadata
□ Include metadata keywords
Title and Description.

General
□ Include default values
Values are taken from JSON.
□ Restrict values to enum
Uses the default value and null.
☑ Force required

Objects
☑ Allow additional properties
Controls whether it's valid to have additional properties in the object beyond what is defined in the schema.

Arrays
☑ Allow additional items
Controls whether it's valid to have additional items in the array beyond what is defined in the

Edit view has been disabled while bugs are fixed. Code and String views work as normal.
Thank you for your patience.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "address": {
      "type": "object",
      "properties": {
        "streetAddress": {
          "type": "string"
        },
        "city": {
          "type": "string"
        }
      },
      "required": [
        "streetAddress",
        "city"
      ]
    },
    "phoneNumber": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "location": {
            "type": "string"
          },
          "code": {
            "type": "integer"
          }
        },
        "required": [
          "location",
          "code"
        ]
      }
    }
  },
  "required": [
    "address",
    "phoneNumber"
  ]
}
```

JSON Schema Example

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "title": "Buying Reason",
    "description": "Buying Reason Structured Content.",
    "type": "object",
    "properties": {
        "icon": {
            "title": "Icon",
            "type": "string",
            "metadata": {
                "infoText": "Drag on an icon image",
                "renderingHint": "image"
            }
        },
        "title": {
            "title": "Title",
            "type": "string",
            "metadata": {
                "placeholderText": "Enter the reason title"
            }
        },
        "description": {
            "title": "Description",
            "type": "string",
            "metadata": {
                "placeholderText": "Describe the essence of this reason"
            }
        }
    }
}
```

Install the tooling

1. Extract **generator-structured-content.zip** file you can find in **BB-Training/tools/generators**
2. Navigate inside **generator-structured-content** folder
3. Install the generator

```
npm link
```

4. Check that the generator has been added

```
bb-generate list
```

5. A new structured-content row should appear

Create a Structured Content Type:

In **BB-Training/exercise** there is file with the expected JSON data that is required to generate the minimum JSON Schema

1. Navigate to **statics/collection-training/prebuilt**
2. Generate a structured content type executing

```
bb-generate structured-content type
```

3. Answers to the generator

```
Name            -   structured-content-type-training-product
JSON path       -   [PATH_TO_BB-Training]/BB-Training/exercise/product.json
```

Set validations and special data:

1. Open **statics/collection-training/prebuilt/structured-content-type-training-product/schema.json**
2. It's time to customize our schema to add validation and meta-data.

Set required fields:

- Set name, title, subtitle and description as required fields

Set Rich Text Editor Fields:

- Set description and content fields to show a WYSIWYG editor

Set placeholders:

- Set a placeholder for name as → "Enter product's name"
- Set a placeholder for title as → "Enter title"
- Set a placeholder for subtitle as → "Enter subtitle"

For more information:

https://my.backbase.com/docs/how-to-guides/article//give-your-custom-content-some-backbone-with-json-schema

Import to portal

1. Navigate to **statics**
2. Use bb-cli to package the whole collection:
    1. Run:

        ```
        bb-package collection-training/src --prebuilt=collection-training/prebuilt -b sass
        ```

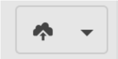
    2. Check a **package.zip** file has been created.
3. Use bb-cli to import the whole collection:

    ```
    bb-import package.zip
    ```

4. Use bb-cli to import the mock collection data:

    ```
    bb-import collection-retail-banking-mocks-2.2.0.zip
    ```

Check it has been added

1. Go to **Shared Content** App.
2. Click in the arrow of this button 
3. Check that you can see a **Product** type on the dropdown.

For more information:

https://my.backbase.com/docs/how-to-guides/article//bring-some-structure-to-your-content/

Create a Product Content Item:

1. Go to **Shared Content** app
2. Select your content type from the Upload dropdown:
3. Input the required content
4. Fill all the fields
5. On array objects add at least two items.
6. Save it
7. You have it available to use in all your portals.

For more information:

https://my.backbase.com/docs/product-documentation/documentation//portal/5.6.2/structcontent_template.html

Create a feature to store templates

1. Navigate to **statics/collection-training/prebuilt**
2. Create a **feature-training-structured-content-templates** folder and navigate inside.
3. Create a **model.xml** file inside with the following code

```
<catalog>
    <feature>
        <name>feature-training-structured-content-templates</name>
        <properties>
            <property label="Title" name="title" viewHint="admin,designModeOnly">
                <value type="string">Training Structured Content Templates</value>
            </property>
            <property label="Version" name="version" viewHint="designModeOnly">
                <value type="string">0.0.1</value>
            </property>
        </properties>
        <tags/>
    </feature>
</catalog>
```

Create a feature to store templates

4. Create a **templates** folder and navigate inside
5. Copy **product-template.html** from **BB-Training/exercise** folder

```
cp -R [PATH_TO_BB-TRAINING]/BB-Training/exercise/product-template.html
./product-template.html
```

Import to portal

1. Navigate to **statics**
2. Use bb-cli to package the whole collection:
    1. Run:

    ```
    bb-package collection-training/src --prebuilt=collection-training/prebuilt -b sass
    ```

    2. Check a **package.zip** file has been created.
3. Use bb-cli to import the whole collection:

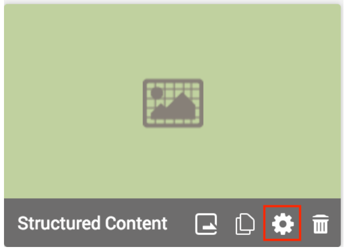
```
bb-import package.zip
```

4. Use bb-cli to import the mock collection data:

```
bb-import collection-retail-banking-mocks-2.2.0.zip
```

Change the template list on server level

1. Open **Enterprise Catalog**
2. Search **Structured Content** in Enterprise Catalog
3. Click on the settings button



4. Click on the **PROPERTIES** tab
5. Replace **Templates** value by

```
Product template,$(contextRoot)/static/features/[BBHOST]/feature-training-
structured-content-templates/templates/product-template.html
```

6. Click on it to add it to your portal

Change the template list on portal level

7. Replace **Template** value by

```
$(contextRoot)/static/features/[BBHOST]/feature-training-structured-
content-templates/templates/product-template.html
```

8. Save it
9. Open Portal Manager
10. Drop Structured Content to one page
11. Click in the widget instance settings icon to show the preferences dialog and you can see that the dropdown now has selected
12. Click in the content tab in the left sidebar.
13. Click in the shared content header to get access to the Structured Content Items.
14. Drop the Product item you filled before on the Structured Content Widget.

ICE Templates

Widget Extras

ICE Widget + Templates

ICE Widget

ICE Templates

BACKBASE

ICE Templates

```
<div class="bt-ice-example">
    <!-- image -->
    <img src="{{simpleContent.image.path}}" />

    <!-- metadata -->
    <h2>{{simpleContent.article.bb:title}}</h2>

    <!-- article.content -->
    <div>{{{simpleContent.article.content}}}</div>

    <!-- link with attributes -->
    <a href="{{simpleContent.link.path}} ”
       title="{{simpleContent.link_desc.bb:title}}">
        {{simpleContent.link_desc.content}}
    </a>

    <!-- Render widget properties -->
    <div title="{{title}}">
        {{title}}
    </div>
</div>
```

Exercise: Create an ICE Template

- Create a custom content template named **media-article.html**, based on the Twitter Bootstrap Media Object: http://getbootstrap.com/components/#media

- Make this template available in the templateList property of the content widget

Via the new extension system in CXP 5.6, it is very easy to extend Content Widgets to add your own custom templates.

Create a feature to store templates

1. Navigate to **statics/collection-training/prebuilt**
2. Create a **feature-training-media-article-template** folder and navigate inside.
3. Create a **model.xml** file inside with the following code

```
<catalog>
    <feature>
        <name>feature-training-media-article-template</name>
        <properties>
            <property label="Title" name="title" viewHint="admin,designModeOnly">
                <value type="string">Training Media Article Template</value>
            </property>
            <property label="Version" name="version" viewHint="designModeOnly">
                <value type="string">0.0.1</value>
            </property>
        </properties>
        <tags/>
    </feature>
</catalog>
```

Create a feature to store templates

1. Create a **templates** folder and navigate inside
2. Inside **templates** create a new **media-article.html** file with the following code:

```
<div class="media be-ex-media-object">
    <h3>Training Media Object Template</h3>
    <div class="media-left">
        <img class="media-object" src="{{article.image.path}}" />
    </div>
    <div class="media-body">
        <h4 class="media-heading">{{article.bb:title}}</h4>
        <div class="media-body">{{{article.contentbody}}}</div>
    </div>
</div>
```

Import to portal

1. Navigate to **statics**
2. Use bb-cli to package the whole collection:
   1. Run:

      ```
      bb-package collection-training/src --prebuilt=collection-training/prebuilt -b sass
      ```

   2. Check a **package.zip** file has been created.
3. Use bb-cli to import the whole collection:

   ```
   bb-import package.zip
   ```

4. Use bb-cli to import the mock collection data:

   ```
   bb-import collection-retail-banking-mocks-2.2.0.zip
   ```

Change the template list on server level

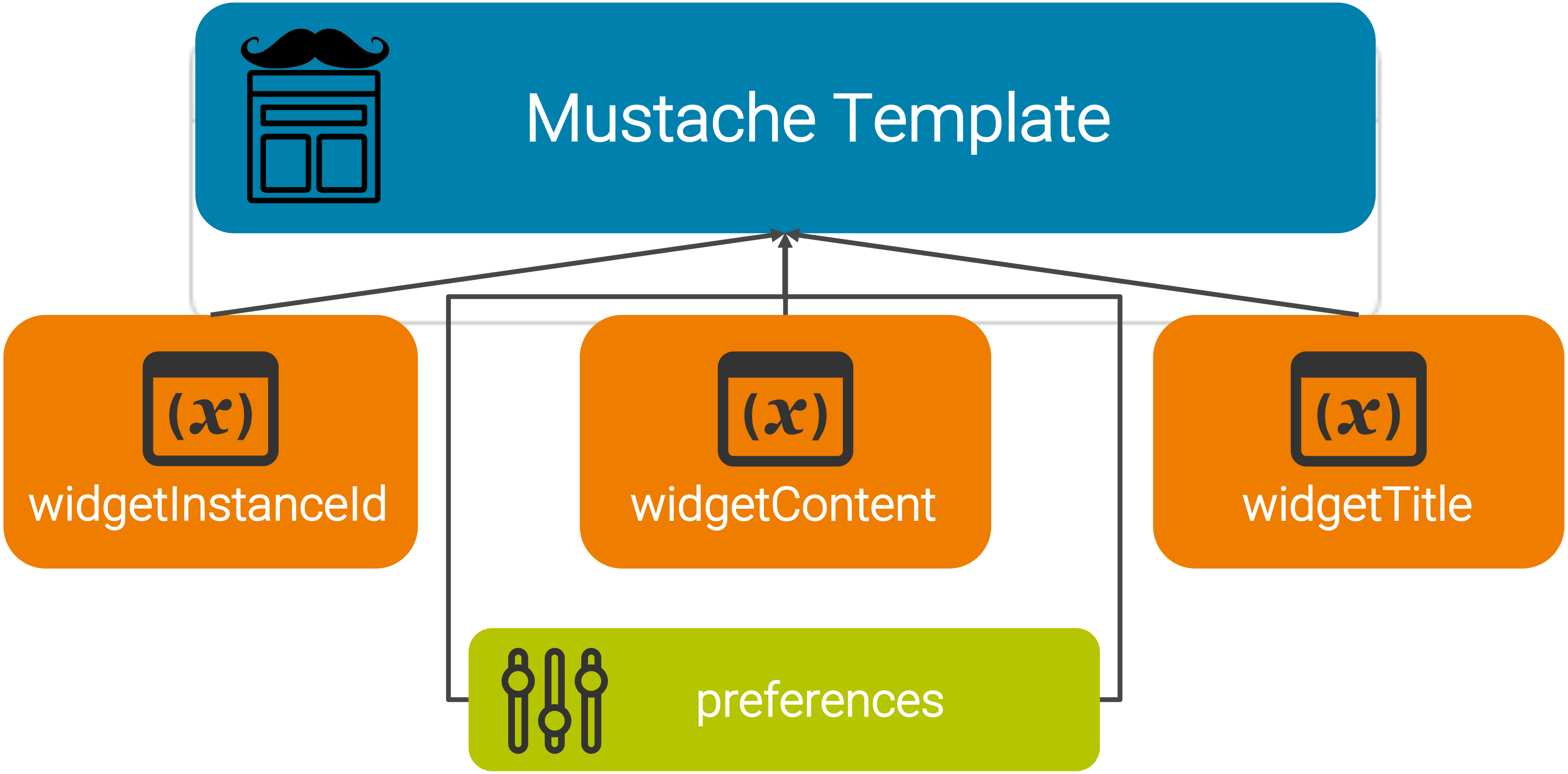
1. Open **Enterprise Catalog**
2. Search **Content** in Enterprise Catalog
3. Click on it to add it to your portal
4. Click on the settings button
5. Click on the **PROPERTIES** tab
6. **Add** the following value to the end of **Templates** list

```
,Media Object,$(contextRoot)/static/features/[BBHOST]/feature-training-media-
article-template/templates/media-article.html
```

Widget Chromes

Widget Extras

Widget Chromes


Mustache Template

widgetInstanceId

widgetContent

widgetTitle

preferences

Widget Chromes

```
<div
  data-pid="{{widgetInstanceId}}"
  class="panel-chrome bp-widget {{#enableDND}}bp-ui-dragRoot{{/enableDND}}" >

  <div class="bp-widget-head">
    <div class="panel-heading">
      <div class="panel-title pull-left">
         {{#pref.icon}}
              <i class="lp-icon lp-icon-{{pref.icon.value}}"></i>
         {{/pref.icon}}
         {{widgetTitle}}
      </div>
    </div>
    <div class="panel-body bp-widget-body">{{{widgetContent}}}</div>
  </div>

</div>
```

Change your Widget Chrome

1. Open Explorer App
2. Select one widget instance in your portal tree.
3. Click on **properties** tab
4. Change **Widget Chrome** value property to

```
$(contextRoot)/static/backbase.com.2012.aurora/html/chromes/widget_default.html
```

5. Go to the page to see the widget instance, and you will see a different wrapper that you can drag and drop in live mode.

Change your Widget Chrome

1. Open Explorer App
2. Select one widget instance in your portal tree.
3. Click on **properties** tab
4. Change **Widget Chrome** value property to

```
$(contextRoot)/static/backbase.com.2012.aurora/html/chromes/widget_default.html
```

5. Go to the page to see the widget instance, and you will see a different wrapper that you can drag and drop in live mode.

Create a feature to store the new chrome

1. Navigate to **statics/collection-training/prebuilt**
2. Create a **feature-training-custom-chrome-template** folder and navigate inside.
3. Create a **model.xml** file inside with the following code

```
<catalog>
    <feature>
        <name>feature-training-custom-chrome-template</name>
        <properties>
            <property label="Title" name="title" viewHint="admin,designModeOnly">
                <value type="string">Training Custom Chrome Template</value>
            </property>
            <property label="Version" name="version" viewHint="designModeOnly">
                <value type="string">0.0.1</value>
            </property>
        </properties>
        <tags/>
    </feature>
</catalog>
```

Create a feature to store the new chrome

1. Create a **templates** folder and navigate inside
2. Inside **templates** create a new **training-custom-chrome.html** file with the following code:

```
<div data-pid="{{widgetInstanceId}}" class="bp-widget {{#enableDND}}bp-ui-dragRoot{{/enableDND}} panel panel-default">
    <div class="bp-widget-head panel-heading {{#enableDND}}bp-ui-dragGrip{{/enableDND}}">
        <h3 class="panel-title">{{widgetTitle}}</h3>
    </div>
    <div class="bp-widget-pref"></div>
    <div class="bp-widget-body panel-body">{{{widgetContent}}}</div>
    <div class="bp-widget-foot panel-footer">
        My Footer
    </div>
</div>
```

Import to portal

1. Navigate to **statics**
2. Use bb-cli to package the whole collection:
    1. Run:

       ```
       bb-package collection-training/src --prebuilt=collection-training/prebuilt -b sass
       ```

    2. Check a **package.zip** file has been created.
3. Use bb-cli to import the whole collection:

   ```
   bb-import package.zip
   ```

4. Use bb-cli to import the mock collection data:

   ```
   bb-import collection-retail-banking-mocks-2.2.0.zip
   ```

Change the template list on widget level

1. Open **Site Explorer App**
2. Search for an instance of your widget
3. Click on the **PROPERTIES** tab
4. Replace the **widgetChrome** value to:

```
$(contextRoot)/static/features/[BBHOST]/feature-training-custom-chrome-template/templates/training-custom-chrome.html
```

Here is a list of other interesting Backbase concepts you might want to read about:

Perspectives

https://my.backbase.com/docs/product-documentation/documentation//portal/5.6.2/cxpweblibrary_perspective.html

Behaviors

https://my.backbase.com/docs/product-documentation/documentation//portal/5.6.2/events_behaviors.html

Thank you!

www.backbase.com
sales-eu@backbase.com

New York: +1 646 478 7538
Amsterdam: +31 20 465 8888