AEM-HTL-Builder vs Manual Component Creation: An Efficiency Test

Introduction

Welcome to the AEM-HTL-Builder efficiency test!

The goal of this document is to evaluate the time difference and challenges encountered between manually creating a component (the current company process) and using the aem-htl-builder module to create the same component. You will have 25 minutes for each scenario. If you run out of time, don't worry; you'll still be asked about the difficulties faced during the process.

Below you will find an html file which you have to use to create PizzeriaModel component and requirements for each part of the component

By participating in this test, you will help me evaluate and potentially optimize AEM-HTL-Builder module, thereby enhancing the overall efficiency of my project.

So let's get started!

Contents

Prerequisites

- Local AEM project set up
- AEM instance is up and running
- Aem-htl-builder module installed and configured (https://github.com/Dept-DTNL/aem-htl-builder.git)

If you haven't yet set up a local AEM project, you can generate a basic one by executing the following command. Please note, you should run this command with administrator privileges to avoid potential build failures.

```
mvn -B org.apache.maven.plugins:maven-archetype-plugin:3.2.1:generate \
-D archetypeGroupId=com.adobe.aem \
-D archetypeArtifactId=aem-project-archetype \
-D archetypeVersion=39\
-D appTitle="My Site" \
-D appId="mysite" \
-D groupId="com.mysite"
```

Specifications/Requirements

HtmlFile

```
<div>
    <div>
        <p>Welcome to</p>
        <p id="title">Pizza Paradise!</p>
        <div>
            <p id="intro">
                Welcome to Pizza Delight, where you'll discover a
delectable assortment of freshly baked pizzas crafted
                with the finest ingredients.
                Indulge in our mouthwatering flavors and experience a slice
of pizza perfection delivered right to your
                doorstep.
            </p>
        </div>
    </div>
    <div>
        <a class="link-special" href="">
            <img class="img-specialImg" src="/imgs/specials/special1.png"
alt="">
            <strong>Specials</strong>
            <h3>Family Deals</h3>
        </a>
    </div>
    <p>The list containing the vega pizzas:</p>
    <ul class="list-pizzaVega">
        <li>
            <div>
                <h3 class="title-pizza">Margarita</h3>
                <p>Vega</p>
            </div>
            <p class="ingredients">Tomato sauce and cheese.</p>
            <a href="#"
class="pizzaLink">www.website.com/en/pizzeria/pizzas/vega</a>
            <p class="price">10.99 €</p>
        </li>
        <li>
            <div>
```

```
                <h3 class="title-pizza">Four cheeses</h3>
                <p>Vega</p>
            </div>
            <p class="ingredients">Four cheeses and tomato sauce.</p>
            <a href="#"
class="pizzaLink">www.website.com/en/pizzeria/pizzas/vega</a>
            <p class="price">12.99 €</p>
        </li>
    </ul>
</div>
```

Sling Models:

Sling model name : PizzeriaModel

Fields:

- title : String
- intro : String
- special : String
- specialCheckbox : Boolean
- specialImgReference : String
- pizzaVegaModel : List<PizzaVegaModel>

Important:

- *Use optional injectionStrategy for all fields*
- *Create a getter for each field*
- *Protect all fields and make getters public*

Sling model name : PizzaVegaModel

Fields

- title : String
- ingredients : String
- pizzaLink : String
- pizzaLinkCheckbox : Boolean
- price : String

Important:

- *Use OPTIONAL as the defaultInjectionStrategy*
- *Apply Lombok Getter instead of creating getters for PizzaVegaModel*
- *Protect all fields*
- *Include @ValueMapValue and @include for each field*

Component Dialog:

Refer to the provided figures for the expected structure and appearance of the component dialog files.

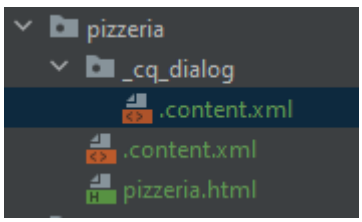

Figure 1 - component dialog file structure

Note: When using the aem-htl-builder, the structure will appear similar, with one notable exception. An additional file called pizzeriaOriginal.html will be present. This file is a copy of the original HTML file and is included to help you easily navigate and refer to the original content of the HTML.

The component consists of two tabs:

Tab 1 => General

Fields:

- title : textfield (add field description : “The name of the pizzeria”)
- intro : textarea
- specialImg : img
- specialContainer (an xml container holding):
    - specialCheckbox : checkbox (value: true)
    - special : pathfield

Tab 2 => Lists

Fields:

- pizzaVegaModelList : Multifield
  Inside the multifield you must include:
    - title : textfield
    - ingredients : textarea
    - pizzaLinkContainer (an xml container holding):
        - pizzaLinkCheckbox : checkbox (value: true)
        - pizzaLink : pathfield
    - price : textfield

Expected result:

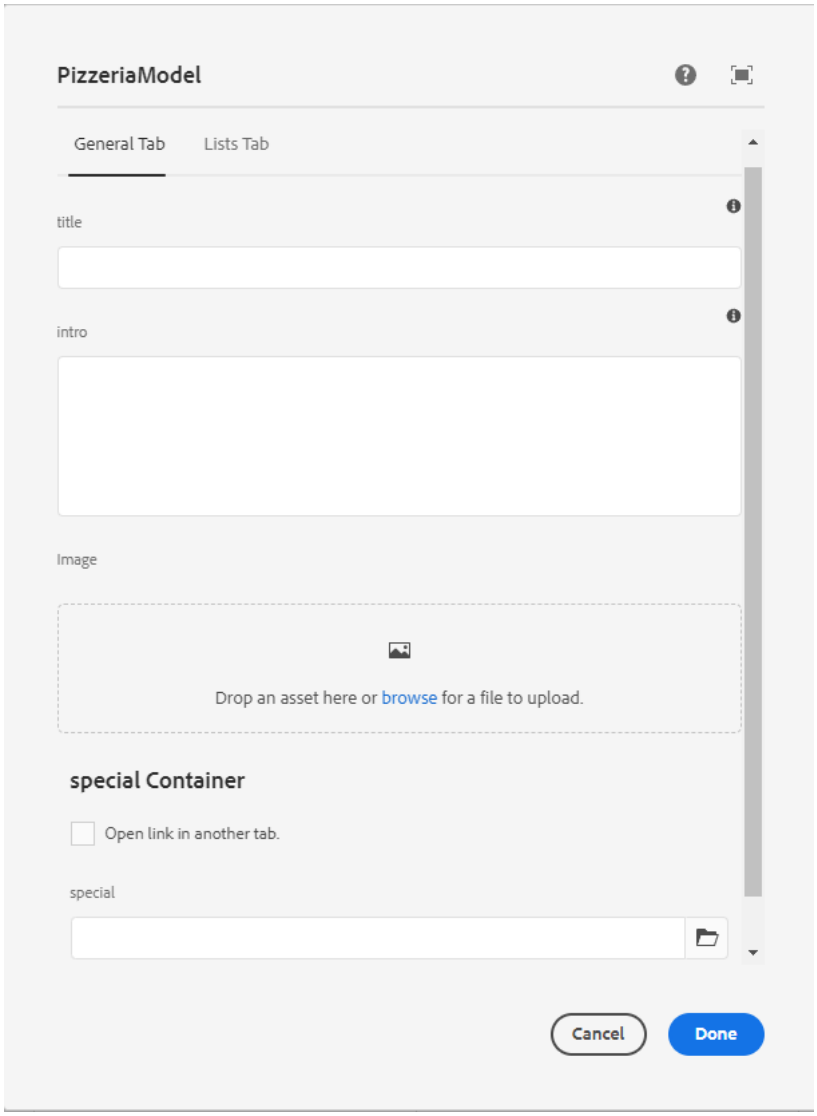

Figure 2 - How the end result of general tab dialog should look like

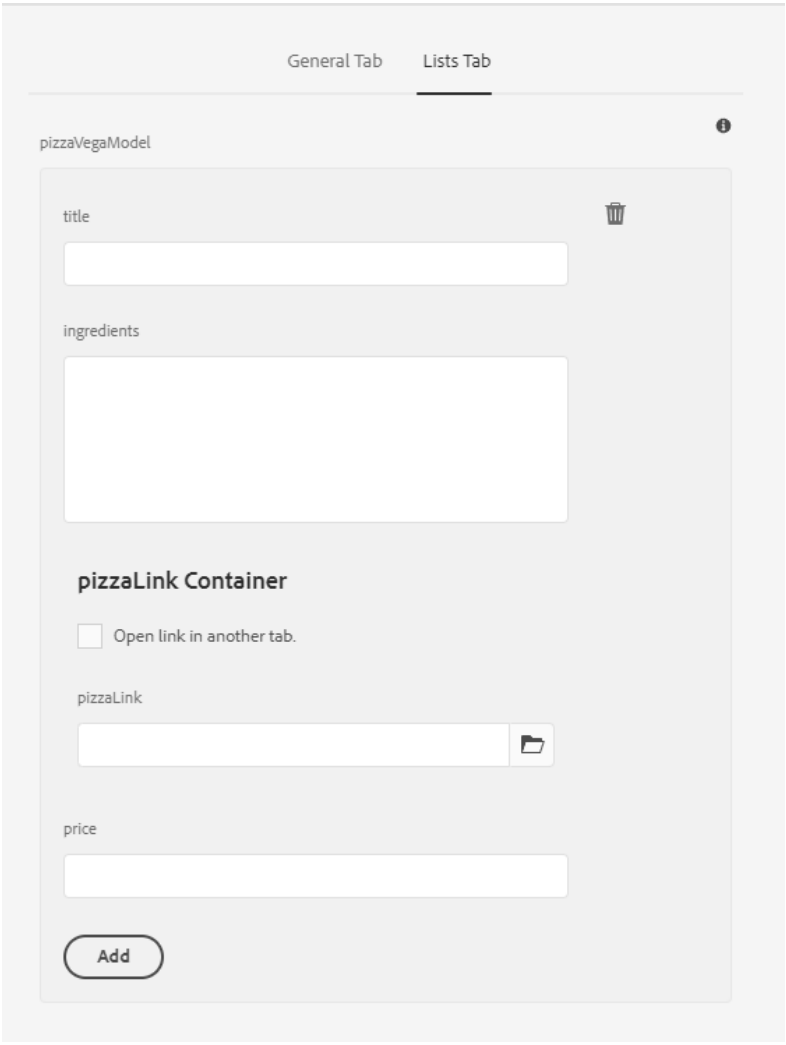

Figure 3 - How the end result of the Lists Tab dialog should look like

Sightly File:

Important :

- Make sure you make reference to component sling model
- Each link (<a> tag) must contain the target attribute. The value of the target is as follows:
    - If the specialCheckbox is **true** return '_blank', if **false** return '_self'
- Modify the behaviour of the img by adding the ` @context='uri' ` in the img src

Test Scenarios

Scenario 1 : Manually create AEM component

1. Create Sling Models for PizzeriaModel and PizzaVegaModel as detailed in the specifications.
2. Develop the Component Dialogs as per the requirements.
3. Ensure the Sightly file is set up correctly, taking note of the special requirements for link and image tags.

Scenario 2 : AEM-htl-builder

Most likely you may have found the manual component creation process to be challenging, time-consuming, and perhaps even tedious. This might have resulted in not finishing the component within the allocated time. However, there's no need to worry.

The aem-htl-builder module is here to streamline this process, making it considerably simpler and more enjoyable.

In this scenario, you will recreate the component from Scenario 1, but this time you'll utilize the aem-htl-builder module. Rather than creating all files and elements manually, you will

1. Edit the HTML file by inserting custom attributes where necessary (https://github.com/Dept-DTNL/aem-htl-builder/blob/main/README.md)

For instance, take a look at this html element:

Html Original:

```
<a class="link-special" href="">
    <img class="img-specialImg" src="/imgs/specials/special1.png" alt="">
    <strong>Specials</strong>
    <h3>Family Deals</h3>
</a>
```

After modifying it with the custom attributes, it should look like this::

```
<a link-special class="link-special" href="">
    <img img-specialImg class="img-specialImg"
src="/imgs/specials/special1.png" alt="">
    <strong>Specials</strong>
    <h3>Family Deals</h3>
</a>
```

Notice the addition of two attributes: **link-special** and **img-specialImg**. In these attributes:

- The first part (e.g., link and img) corresponds to the field type.
- The second part (e.g., special and specialImg) corresponds to the field name.

These attributes signal to the aem-htl-builder module how these HTML elements should be processed.

2. Run the aem-htl-builder convert.
3. Input model name : pizzeria
4. Select that you want to have multiple tabs. Input 2 and first tab will be 'General' and second 'List Tab'

And that is it! The module should now automatically generate all the files (java, xml and html) in your AEM project with the content already inside.

Important!

*For the list to be in separate tab you will need to manually move it in **_cq_dialog/.content.xml** file (Figure 1 - component dialog file structure) to the section below and remove it from the original position:*

In order to move the list to a separate tab, you'll need to modify the `_cq_dialog/.content.xml` file manually (refer to Figure 1 - component dialog file structure).

Your task involves extracting the `pizzaVegaModelList` section from its original location and inserting it into a new area of the XML structure, as described below:

1. *Locate the `pizzaVegaModelList` code in the XML file.*
2. *Copy the entire `pizzaVegaModelList` section.*
3. *Find the following section in the XML file:*

```
<ListTab
        jcr:primaryType="nt:unstructured"
        jcr:title="ListTab Tab"

sling:resourceType="granite/ui/components/coral/foundation/fixedcolumns"
        margin="{Boolean}true">
    <items jcr:primaryType="nt:unstructured">
        <column
                jcr:primaryType="nt:unstructured"

sling:resourceType="granite/ui/components/coral/foundation/container">
            <items jcr:primaryType="nt:unstructured">
                <!--Paste in here pizzaVegaModelList-->
            </items>
        </column>
    </items>
</ListTab>
```

4. *Replace the comment <!--Paste in here pizzaVegaModelList--> with the pizzaVegaModelList section you cut or copied earlier*

Troubleshooting and Assistance

Should you run out of time or encounter any difficulties, I've got you covered. I understand that this may be a new process for some, and it's absolutely okay if you need some assistance.

Below you'll find example how the edited HTML file should look like before running the aem-htl-builder convert command.

Please use this example only if you're running out of time or are truly struggling with the task. It's important that you try your best to edit the HTML file independently, as the aim of this test is to understand your proficiency and the challenges you encounter in this process. Your honest participation ensures the validity and usefulness of this test.

Thank you for your cooperation!

Edited HTML file:

```
<div>
    <div>
        <p>Welcome to</p>
        <p textfield-title description="The name of the pizzeria"
id="title">Pizza Paradise!</p>
        <div>
            <p textarea-intro id="intro">
                Welcome to Pizza Delight, where you'll discover a
delectable assortment of freshly baked pizzas crafted
                with the finest ingredients.
                Indulge in our mouthwatering flavors and experience a slice
of pizza perfection delivered right to your
                doorstep.
            </p>
```

```
        </div>
    </div>
    <div>
        <a link-special class="link-special" href="">
            <img img-specialImg class="img-specialImg"
src="/imgs/specials/special1.png" alt="">
            <strong>Specials</strong>
            <h3>Family Deals</h3>
        </a>
    </div>
    <p>The list containing the vega pizzas:</p>
    <ul list-pizzaVega class="list-pizzaVega">
        <li>
            <div>
                <h3 textfield-title class="title-pizza">Margarita</h3>
                <p>Vega</p>
            </div>
            <p textarea-ingredients class="ingredients">Tomato sauce and
cheese.</p>
            <a link-pizzaLink href="#"
class="pizzaLink">www.website.com/en/pizzeria/pizzas/vega</a>
            <p textfield-price class="price">10.99 €</p>
        </li>
        <li>
            <div>
                <h3 class="title-pizza">Four cheeses</h3>
                <p>Vega</p>
            </div>
            <p class="ingredients">Four cheeses and tomato sauce.</p>
            <a href="#"
class="pizzaLink">www.website.com/en/pizzeria/pizzas/vega</a>
            <p class="price">12.99 €</p>
        </li>
    </ul>
</div>
```

Publish component

Once you are done you can just run the following command in your AEM project which will publish the newly created component

```
mvn clean install -PautoInstallPackage -DskipTests
```

Test Results

Contact information

Please send the test results to

Email: marek.stryjenski@deptagency.com

Scenario 1: Manual Creation

Time :

Most common challenges/difficulties:

Scenario 2: Aem-htl-builder module

Time :

Most common challenges/difficulties: