Table of Contents

[#1 2](#_Toc111423660)

[RSS Feed 2](#_Toc111423661)

[Component Design 2](#_Toc111423662)

[Screenshots 3](#_Toc111423663)

[Dynamic List 3](#_Toc111423664)

[Fixed List 4](#_Toc111423665)

[#2 6](#_Toc111423666)

[Content Structure Generic MSM Blueprint 6](#_Toc111423667)

[Test Rollout 7](#_Toc111423668)

[Result 7](#_Toc111423669)

[Creating Experience Fragments with MSM Capability 8](#_Toc111423670)

[OOTB Rollout Feature 8](#_Toc111423671)

[Custom Experience Fragment Rollout 9](#_Toc111423672)

[Approach 1 9](#_Toc111423673)

[Approach 2 10](#_Toc111423674)

[#3 11](#_Toc111423675)

[Technical Documentation 11](#_Toc111423676)

[Project Creation 11](#_Toc111423677)

[Command to deploy the entire project: 11](#_Toc111423678)

[Environment: 11](#_Toc111423679)

# #1

# RSS Feed

Component to read external RSS Feed & display most recent feeds

## Component Design

**RSS Feed Component** (rss-feed) – A touch UI enabled AEM component that displays RSS Feeds based on dropdown selection – Fixed/Dynamic as per component dialog design below

**RSSFeedModel** – Represents a single RSS Feed

**RSSFeedListModel** – Sling model associated to the AEM component ‘RSS Feed Component (rss-feed)’ – to fetch dynamic list of RSS Feeds based on the URL provided

* If the Feed is empty or URL is incorrect, the component will suggest inputting a fixed list of RSS Feeds as per the dialog design below
* The Model sorts the feed by most recent and limits the feeds based on the count provided in RSS Feed Component (rss-feed) component dialog

RSS Source – <https://sports.ndtv.com/rss/cricket>

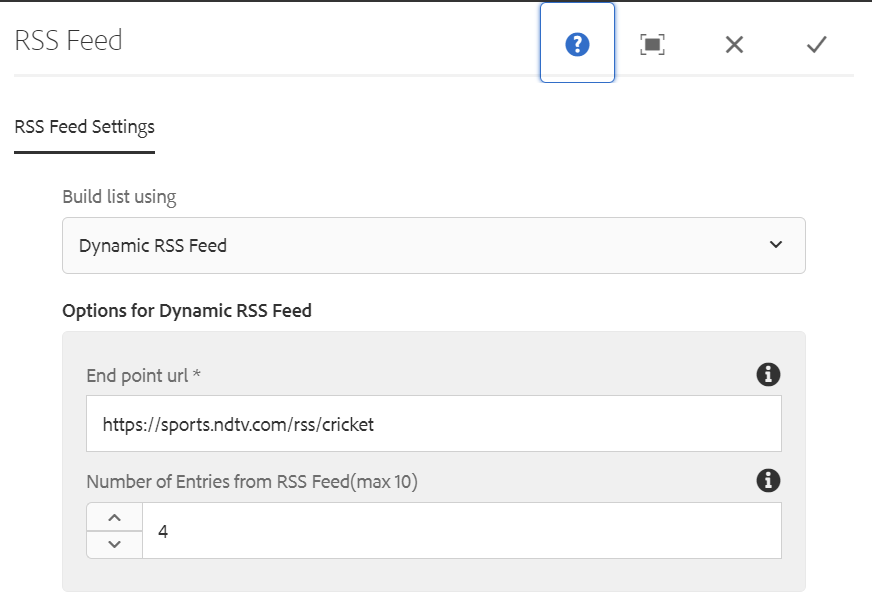
Touch UI Dialog Design

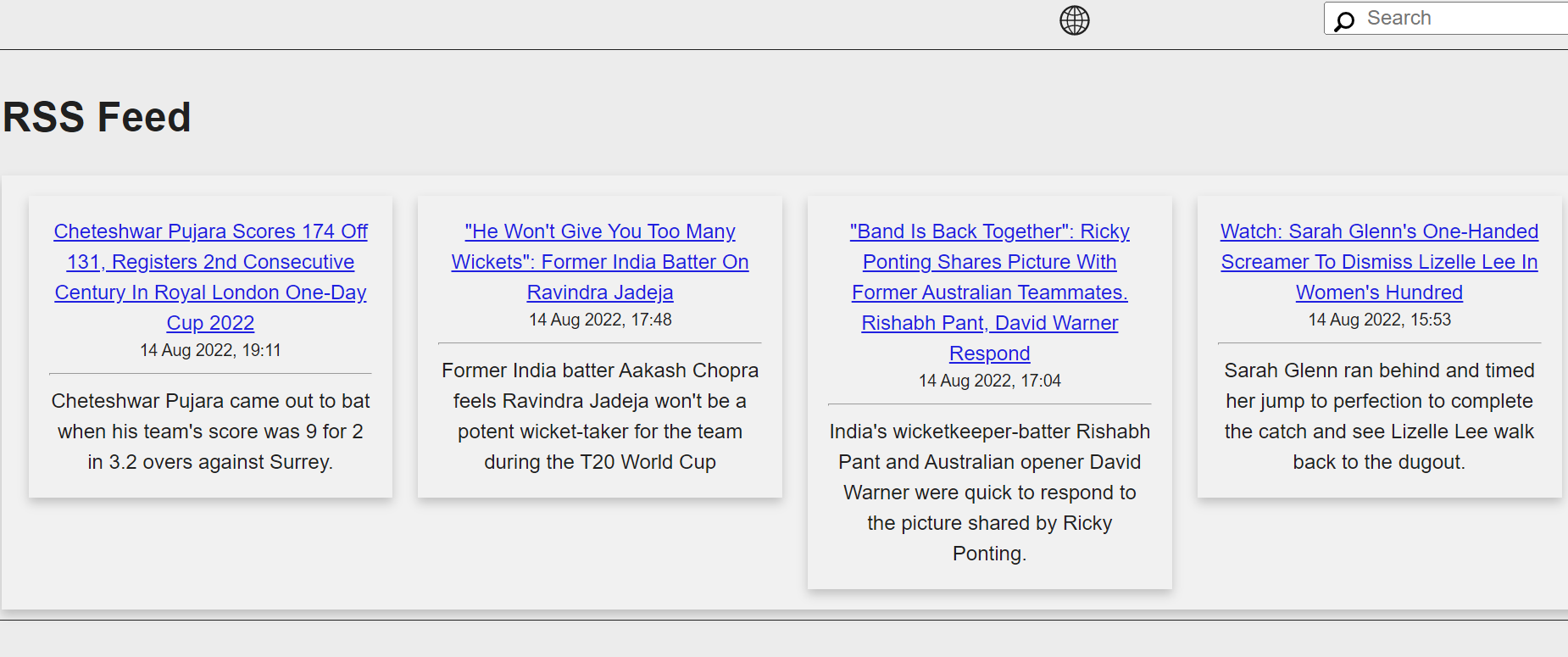
|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Type | Value | Description |
| Build List by | Dropdown | Dynamic/Fixed List | Based on selection will either fetch feed dynamically or option build a fixed list |
| Dynamic Options   * END Point URL * Number of Feeds | Single Fields   * Textfield * NumberField |  | * RSS Feed URL * Number of feeds to display dynamically |
| Fixed List Options   * Title * Description * Link * Published Date | Multifield   * Textfield * TextArea * Textfield * DateField |  | Display Feed based on values provided in fixed options |

Assumptions: Single component to display based on dropdown selection, a separate component can be created for individual feeds as well based on further requirement analysis.

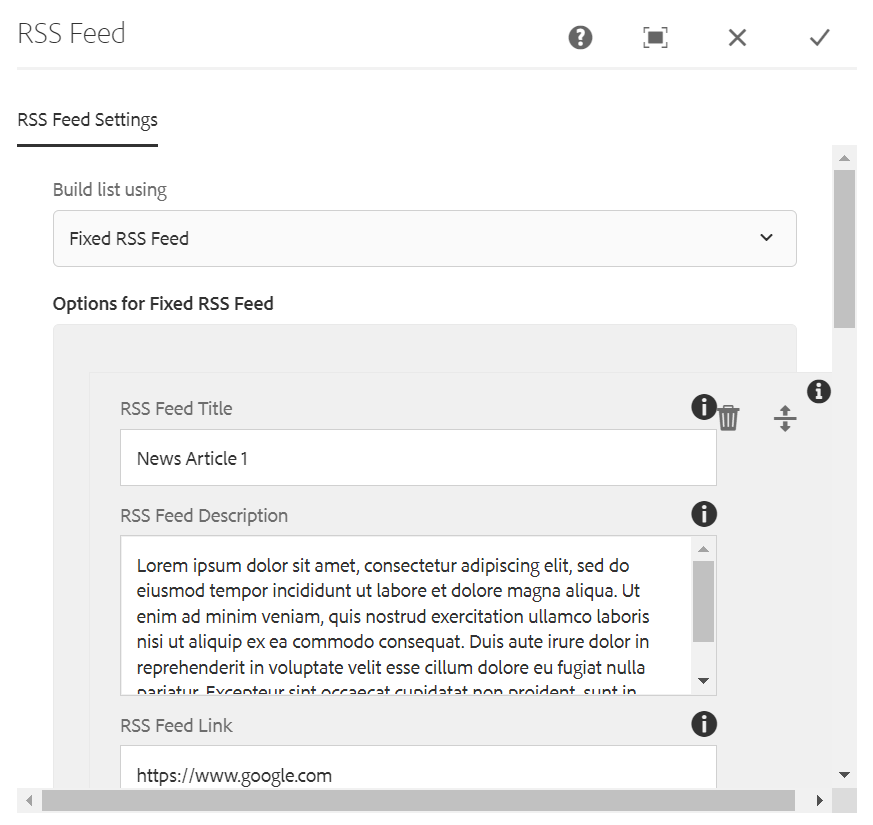
# Screenshots

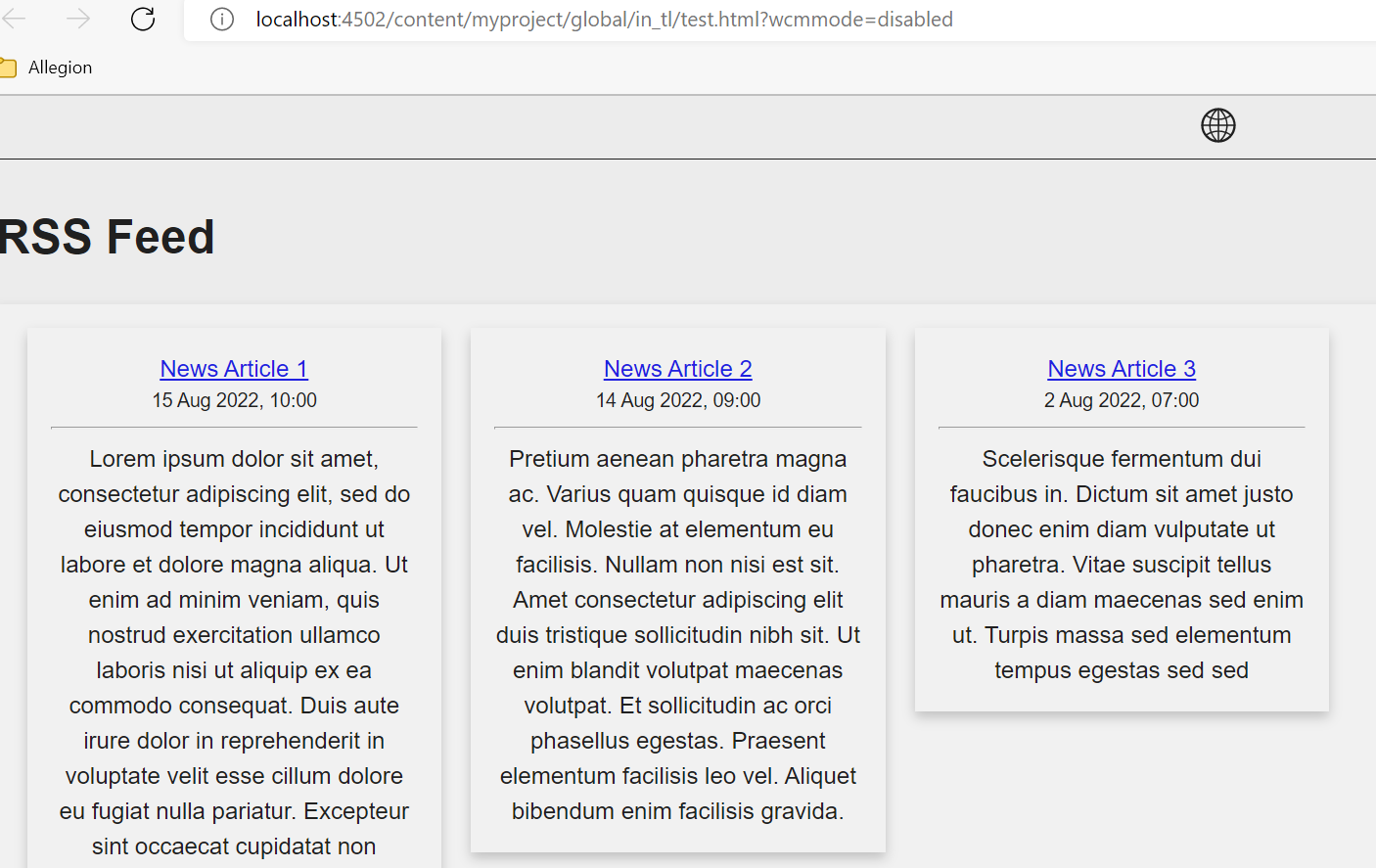
## Dynamic List





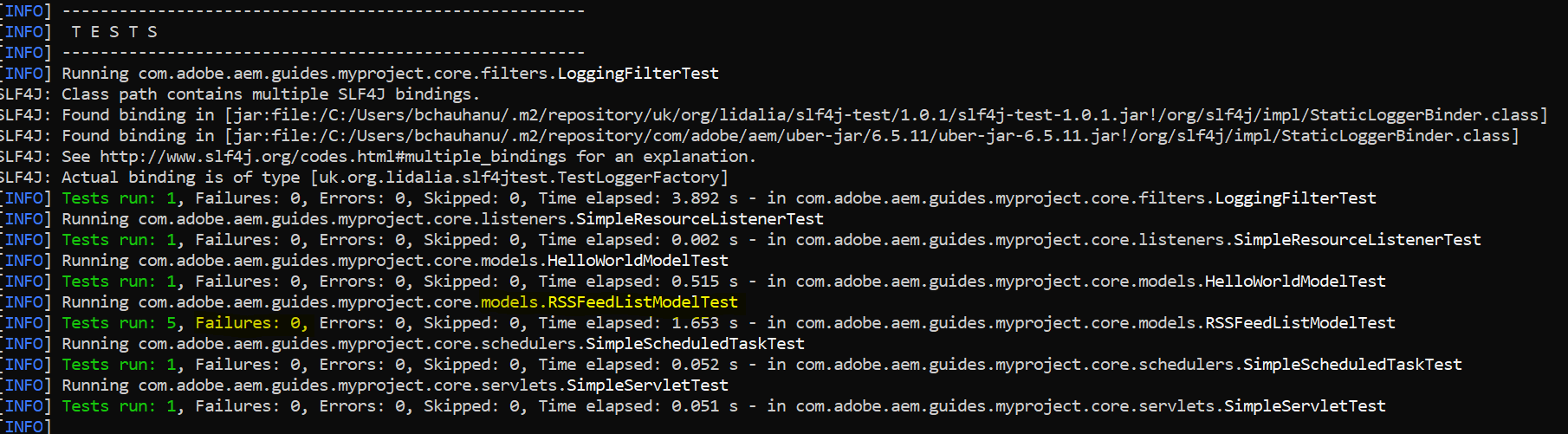
## Fixed List





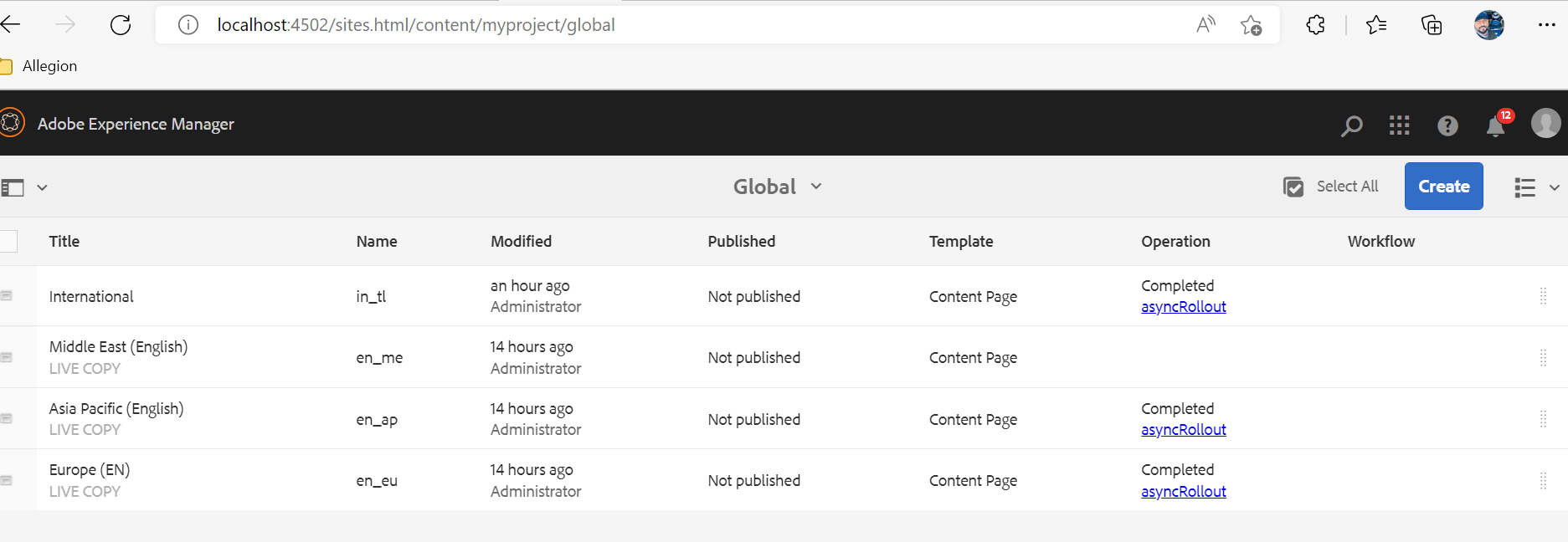
JUnit Test Cases  
Environment: JUnit5, AemContext

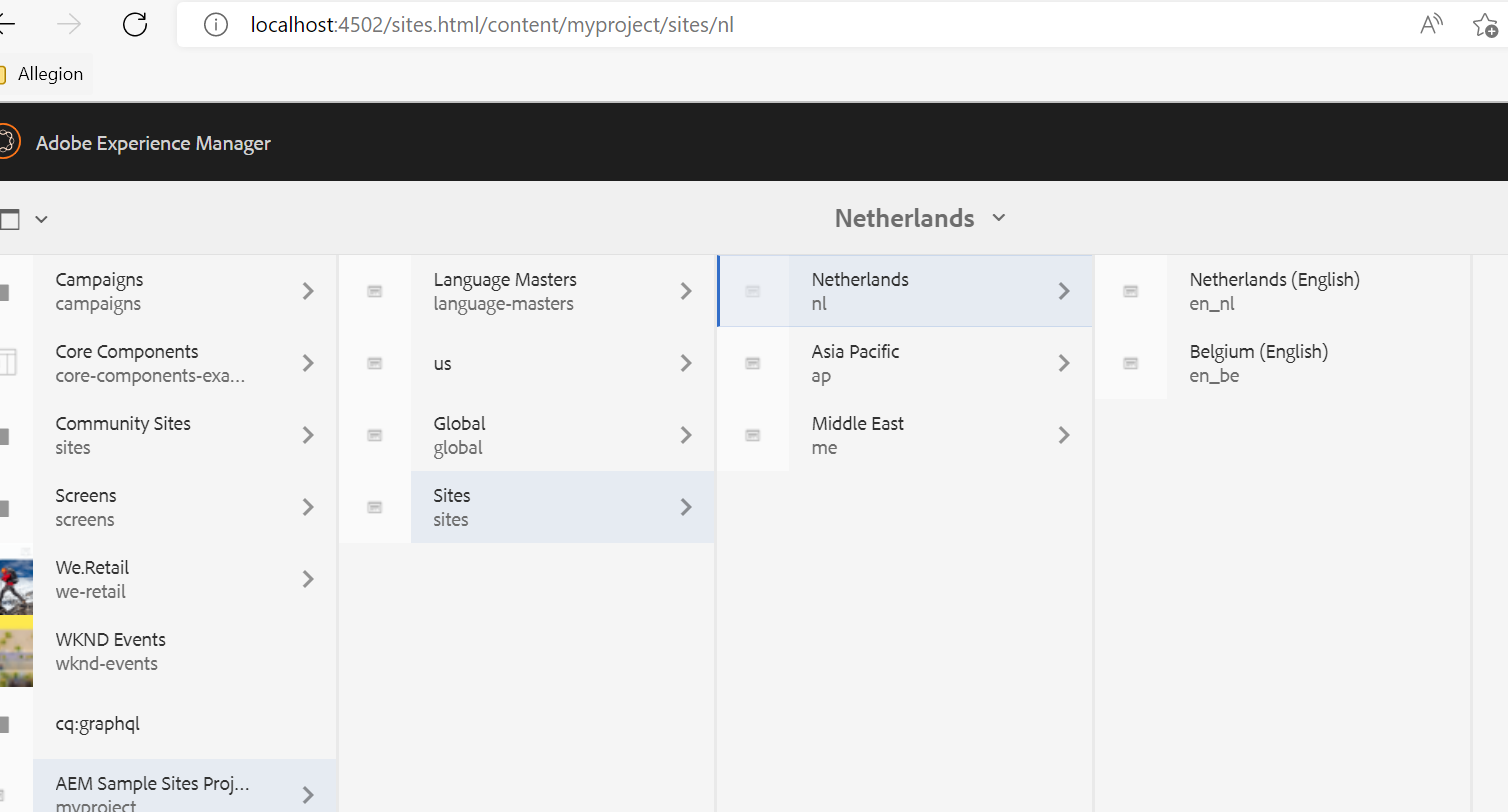
1. Test EndPoint URL
2. Check for valid EndPoint URL
3. Test count of feeds
4. Test Dialog dropdown value
5. Test if Fixed list of feeds exist

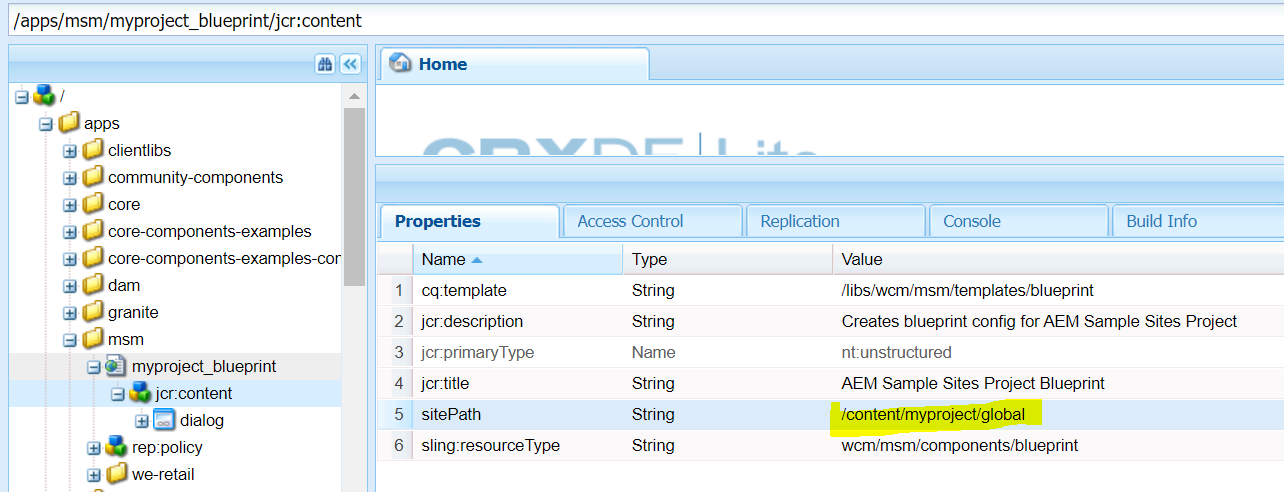


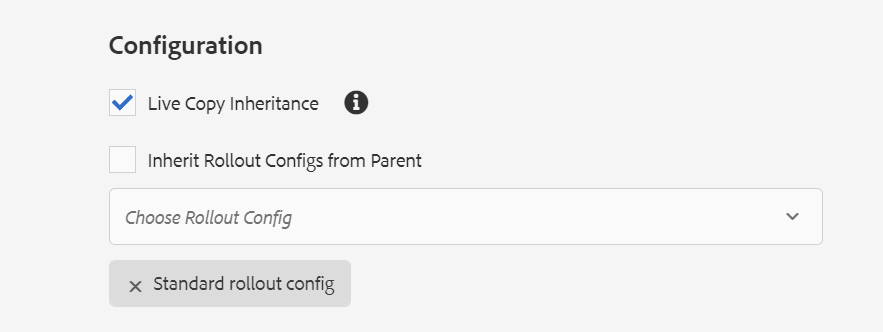
# #2

## Content Structure Generic MSM Blueprint

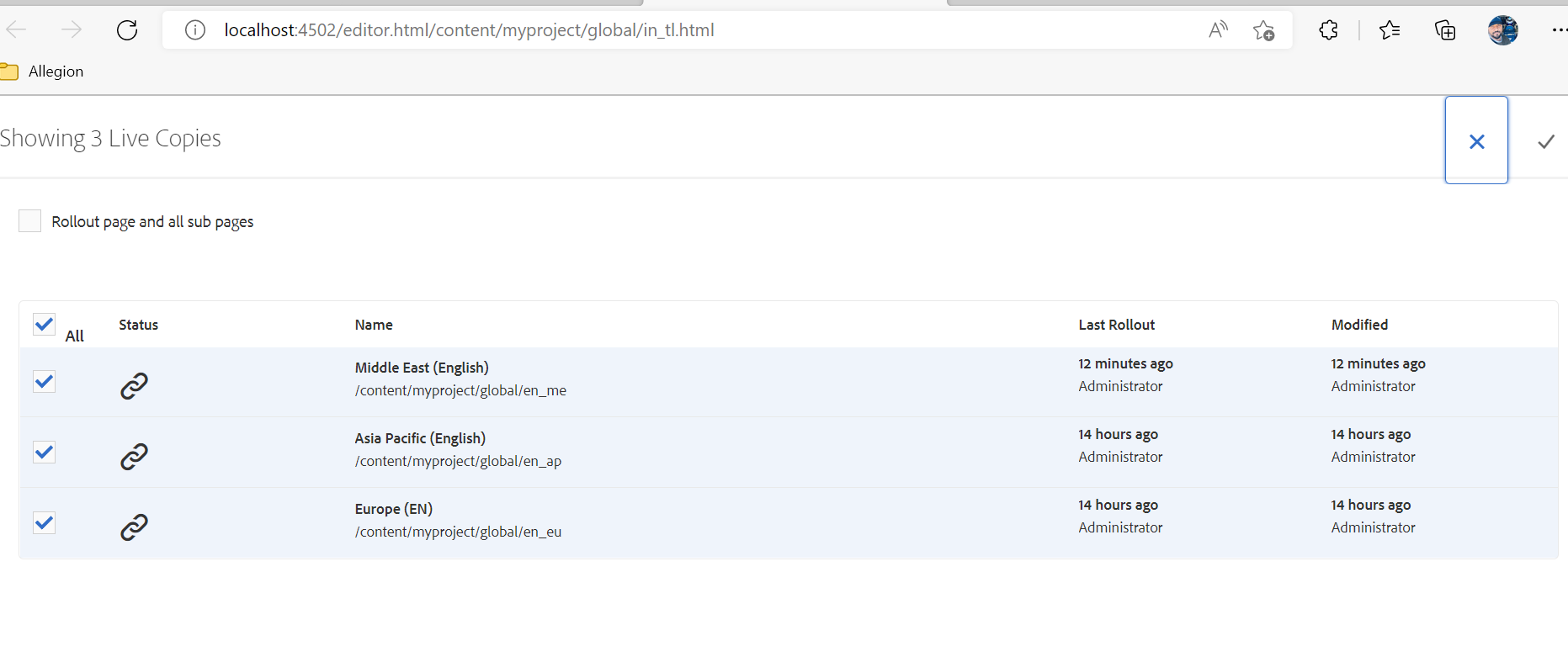
Language Master Content Structure  


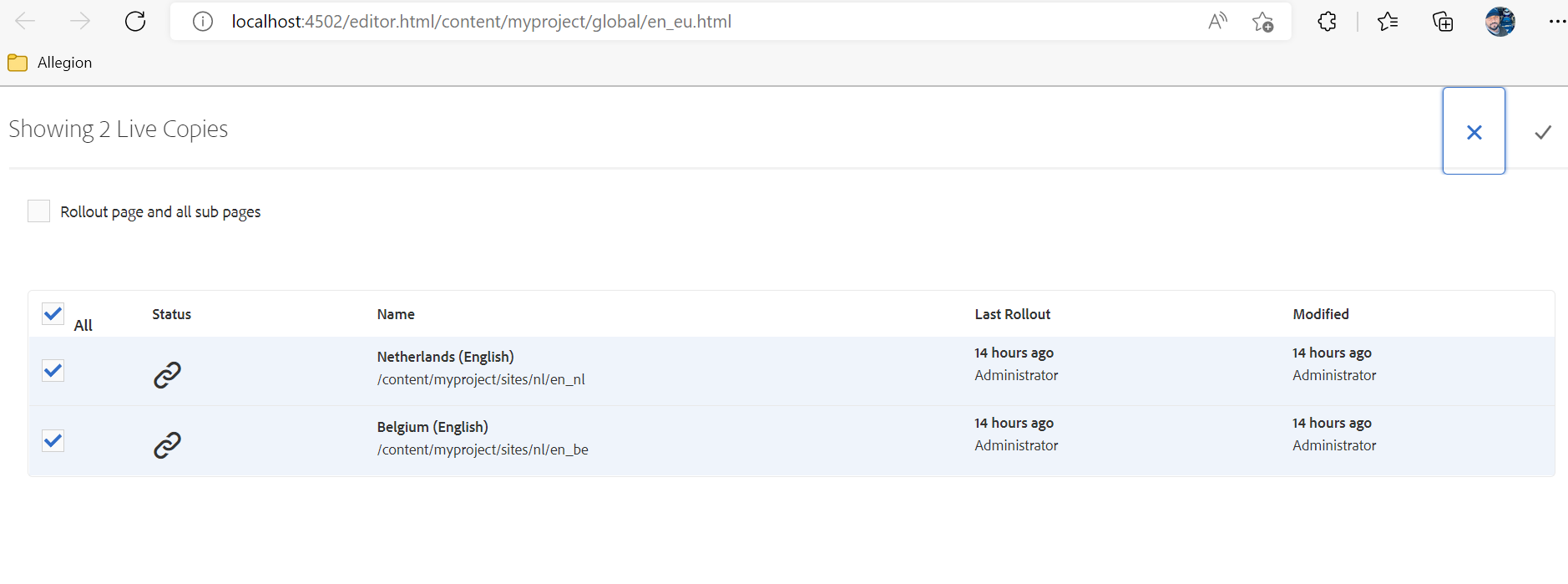
Sites content structure  


Blueprint Source: /content/myproject/global  


Rollout configurations: Standard  


## Test Rollout

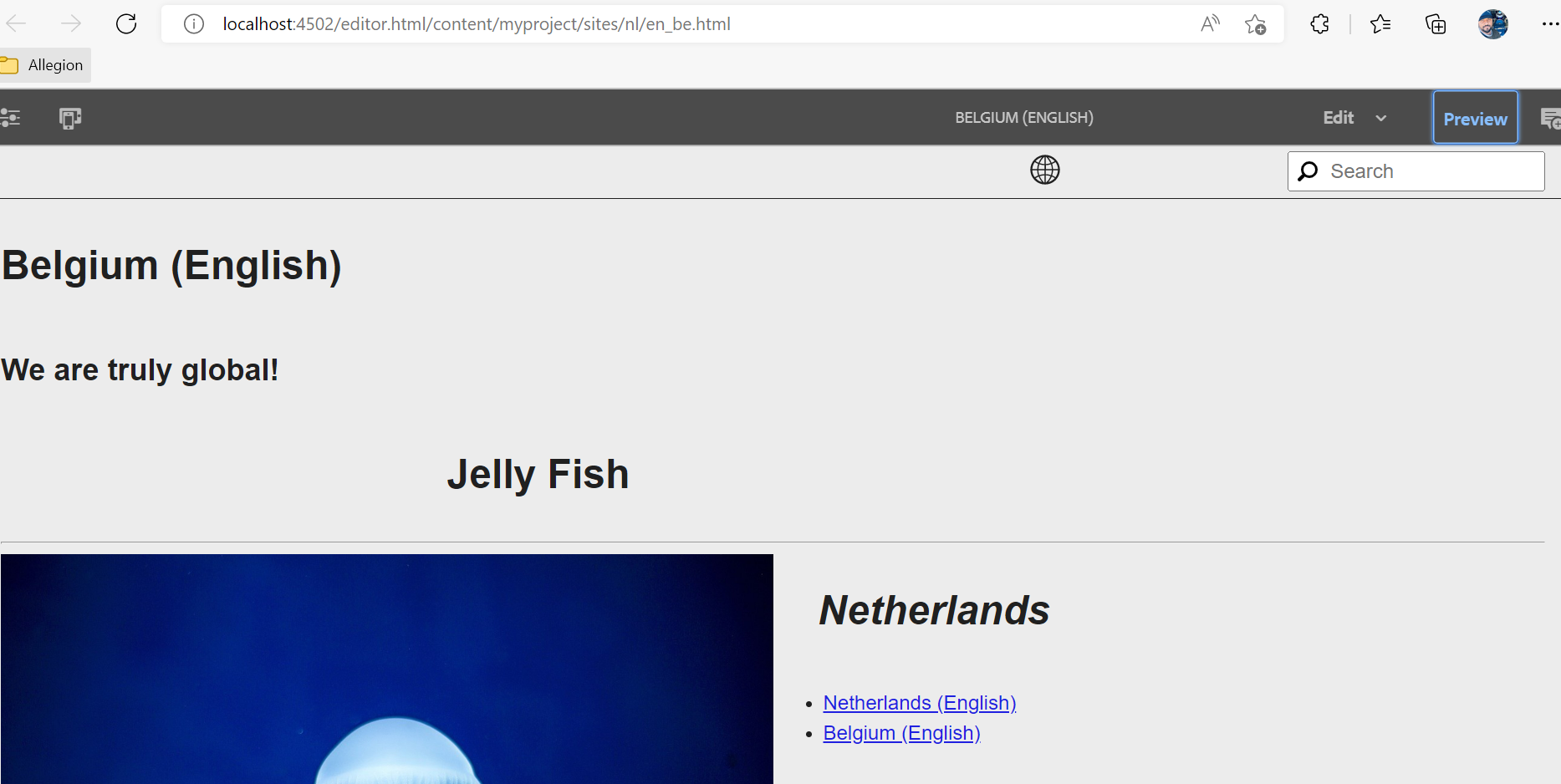
From Page Options – Select Rollout from in\_tl -> en\_me, en\_ap, en\_eu  


Further Rollout changes to individual sites:  
From en\_eu -> en\_nl, en\_be  


## Result

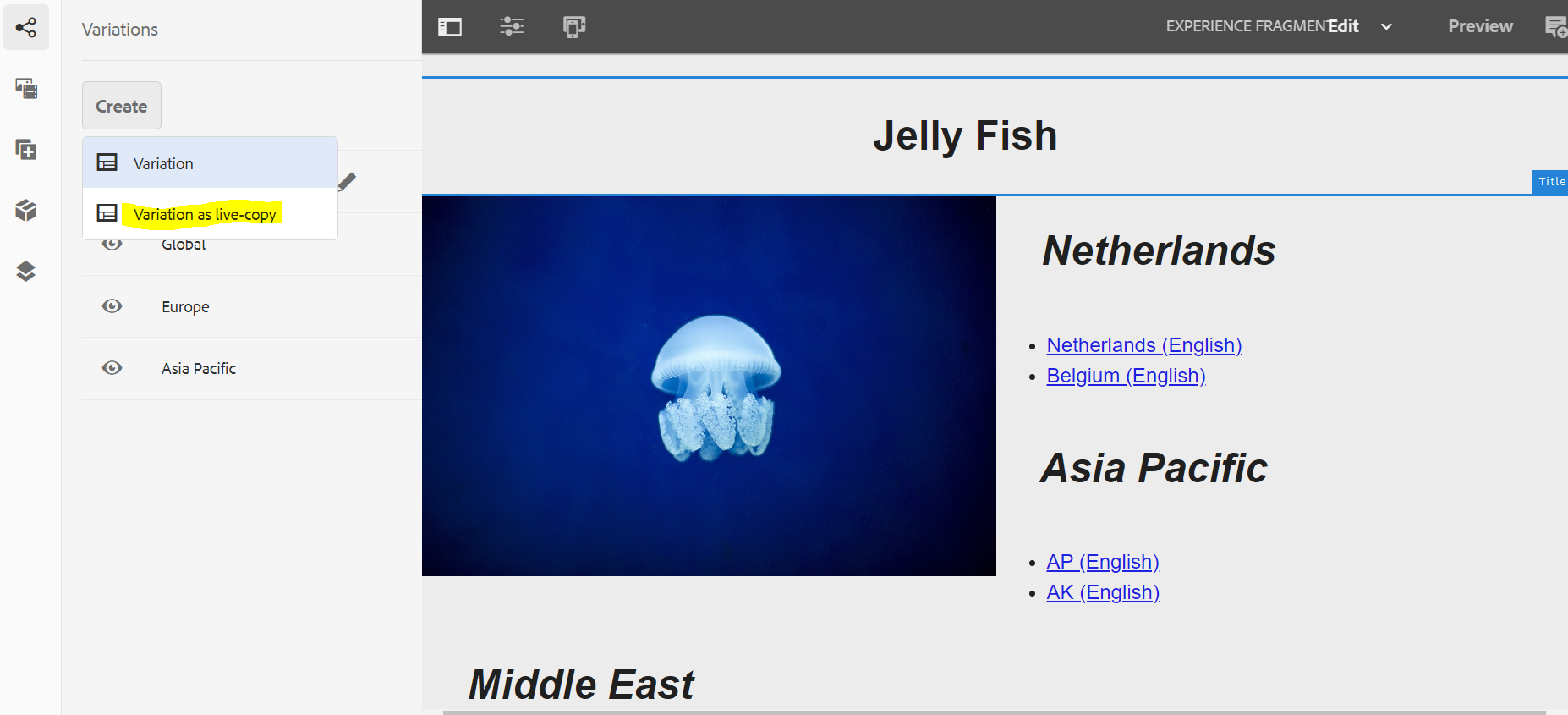
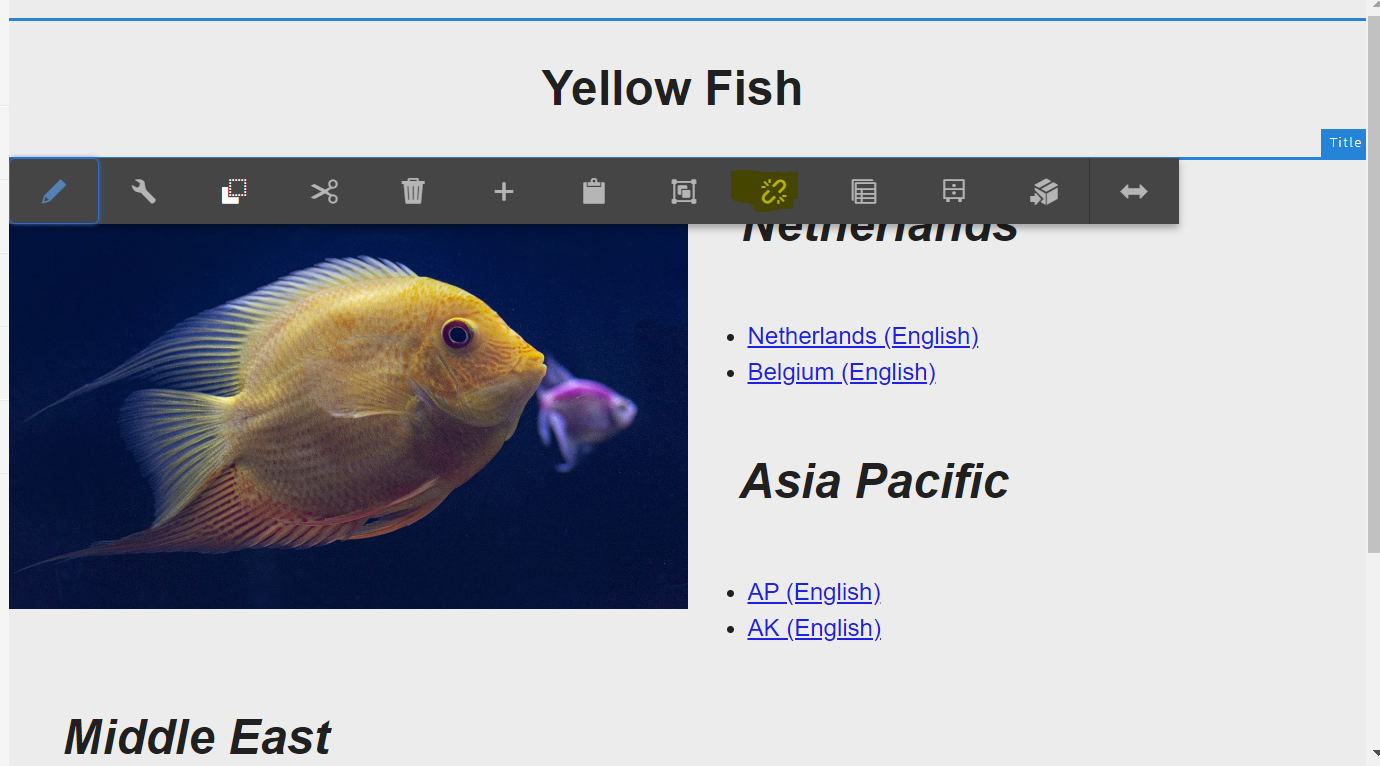
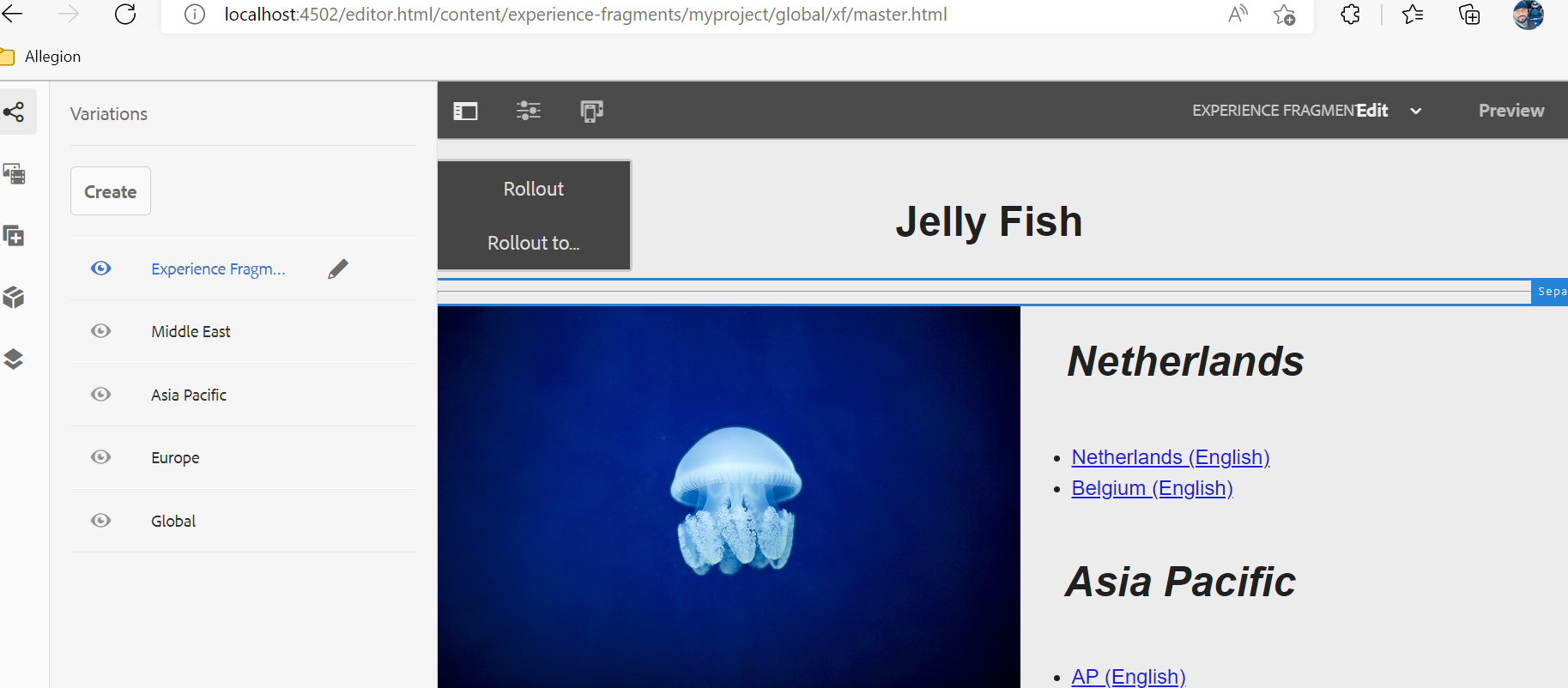
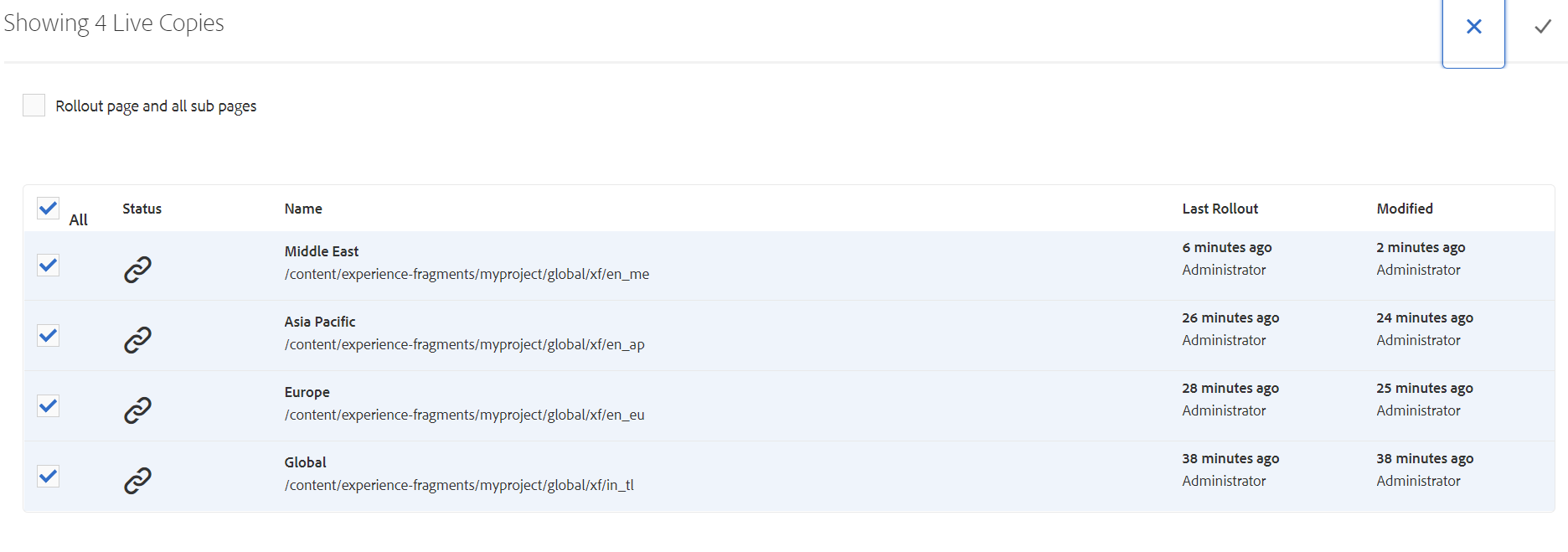
Content is pushed to live copies accordingly.

Sample – Belgium (English) site



# Creating Experience Fragments with MSM Capability

## OOTB Rollout Feature

1. Create experience fragment under folder structure - /content/experience-fragments/myproject/global
2. Create Master variation by adding relevant content
3. Create as many variations as live copy of master variation   
   
4. Update content as necessary by cancelling inheritance as shown  
   
5. Add a separator component to master variation and rollout to all variations  
     
     
   
6. Updates to master variation will be rolled out to all live copies.

## Custom Experience Fragment Rollout

By default, we will not be able to rollout experience fragments to sites. I am detailing steps that can be taken to achieve this

## Approach 1

1. Create a custom blueprint configuration  
   <?xml version=*"1.0"* encoding=*"UTF-8"*?>

<jcr:root xmlns:sling=*"http://sling.apache.org/jcr/sling/1.0"* xmlns:cq=*"http://www.day.com/jcr/cq/1.0"*

xmlns:jcr=*"http://www.jcp.org/jcr/1.0"* xmlns:nt=*"http://www.jcp.org/jcr/nt/1.0"*

cq:template=*"/libs/wcm/msm/templates/blueprint"*

jcr:primaryType=*"nt:unstructured"*

jcr:title=*"AEM Sample Sites Project Experience Fragment Blueprint"*

jcr:description=*"Creates blueprint config for AEM Sample Sites Experience Fragment Project"*

sling:resourceType=*"wcm/msm/components/blueprint"*

sitePath=*"/content/experience-fragments/global"*>

<dialog/>

</jcr:root>

1. Create Custom Rollout Action   
   /apps/msm/wcm/rolloutconfigs/myproject/xfSitesUpdate  
     
   This rollout from any experience fragment will retrieve all of its variations and references, then start publishing to sites based on the language\_code

## Approach 2

1. Assuming the experience fragments are created with a consistent language\_code structure
2. Create Custom Rollout Action Configuration  
   /apps/msm/wcm/rolloutconfigs/myproject/xfReferenceUpdater  
   OR
3. Create a Custom Process Workflow to Update XF References for a particular payload   
     
   This rollout from base site(in\_tl) or any live copy source will update all the referred experience fragments in live copy sites based on their language code

Either approach will need some additional effort/further discovery in development – hope the approach can be validated.

# #3

# Technical Documentation

## Project Creation

Maven Command used to create the project

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=35 -D appTitle="AEM Sample Sites Project" -D appId="myproject" -D groupId="com.adobe.aem.myproject" -D artifactId="myproject" -D package="com.adobe.aem.guides.myproject" -D version="0.0.1-SNAPSHOT" -D aemVersion="6.5.11" -D includeDispactcherConfig=n -D frontendModule=general -D singleCountry=n -D includeExamples=y

## Command to deploy the entire project:

mvn clean install -PautoInstallPackage

## Environment:

Java Version: 11

Maven Version: 3.8.6

AEM Version: 6.5.11

Maven ArcheType: 35

Package/Dependencies:

Core WCM Components v2.17.12 built along with project

Content - myproject-content-backup-1.0.zip

Examples Included for speed of content build