AEM

Contents

[**Setup** 1](#_Toc162253646)

[**Setup enviproment** 1](#_Toc162253647)

[**Debuger in aem set up** 1](#_Toc162253648)

[**Add adobe repo tool** 1](#_Toc162253649)

[**Component and template** 1](#_Toc162253650)

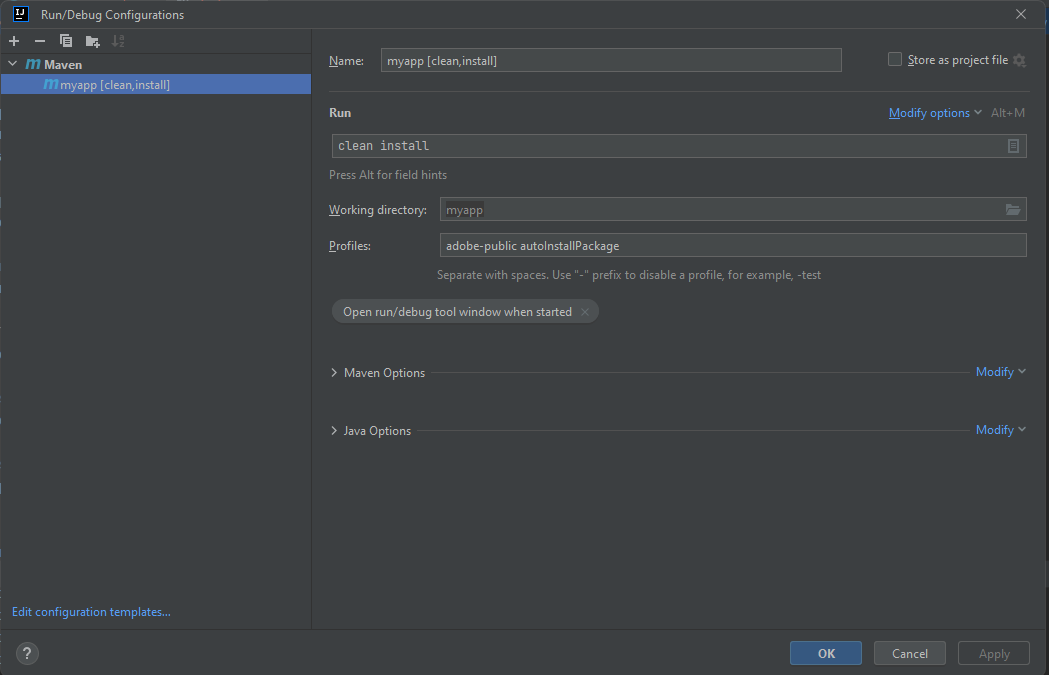
[**Component** 1](#_Toc162253651)

# **Setup**

## **Setup enviproment**

**Create AEM with maven:**

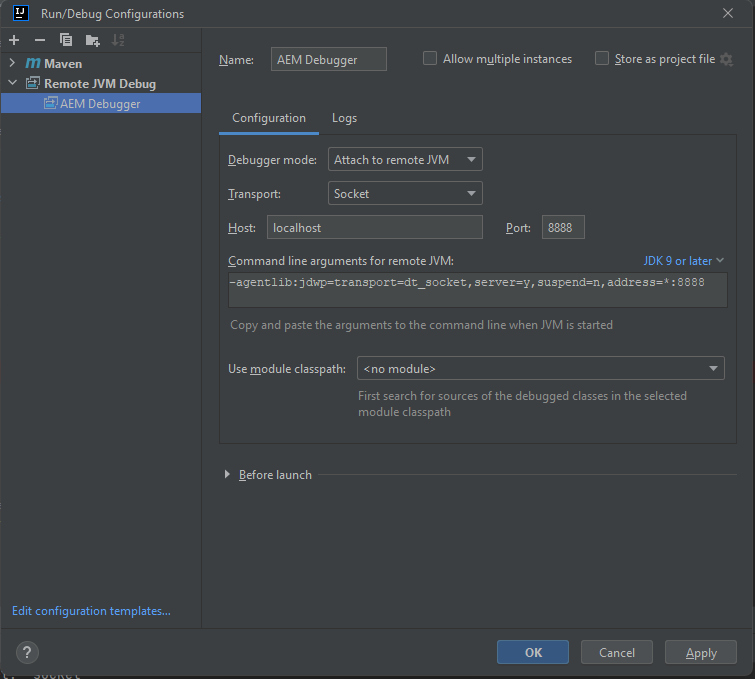
mvn -B archetype:generate -D archetypeGroupId=com.adobe.aem -D archetypeArtifactId=aem-project-archetype -D archetypeVersion=26 -D appTitle="My App Aem Tutorial" -D appId="myapp" -D groupId="com.myapp" -D frontendModule="none" -D aemVersion=6.5.0

after create: mvn install -> set up maven Run 

## **Debuger in aem set up**

Code run debuger in aem : java -Xmx2048M -agentlib:jdwp=transport=dt\_socket,address=8888,server=y,suspend=n -jar aem-author-p4502.jar

Setup intejj Debugger



Take pointer in to there want debug and restart page constain component or service

## **Add adobe repo tool**

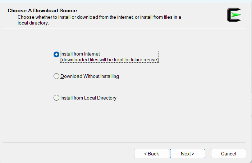
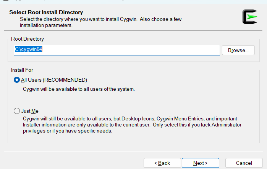
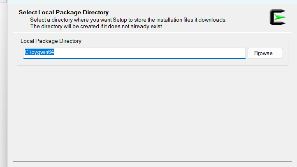
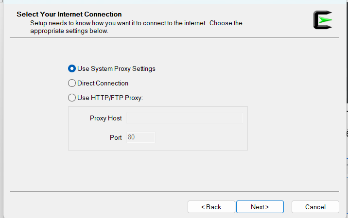
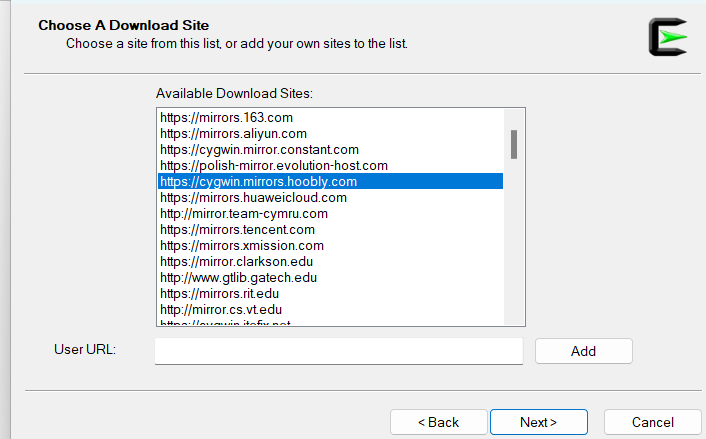
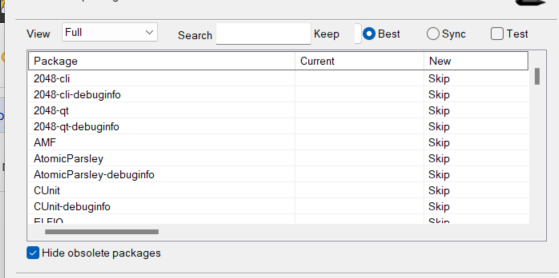
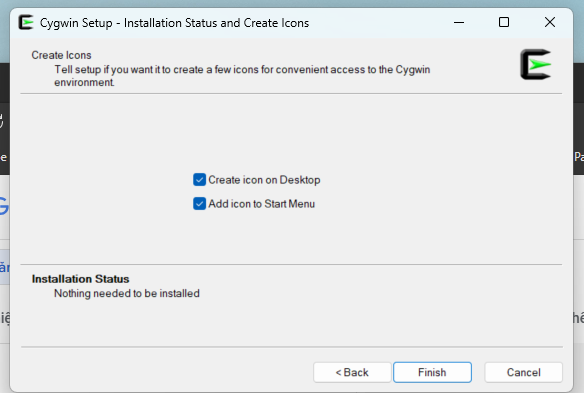
Link download: [tools/repo at master · Adobe-Marketing-Cloud/tools · GitHub](https://github.com/Adobe-Marketing-Cloud/tools/tree/master/repo)

Setup In intelejj:

File -> setting -> tools -> External tools

|  |  |
| --- | --- |
|  | Các method |
|  | Program (Cygwin64): C:\cygwin64\bin\bash.exe  Argument : -l F:\work-space\aem\tools-repo-1.4\repo\repo get -f "$UnixSeparators($FilePath$)$"  Working directory: $ProjectFileDir$ |
|  | Program(Cygwin64): C:\cygwin64\bin\bash.exe  Argument: -l F:\work-space\aem\tools-repo-1.4\repo\repo put -f "$UnixSeparators($FilePath$)$"  Working directory: $ProjectFileDir$ |

Program: đường dẫn của Cygwin64(là environment chứa các thư viện để chạy repo tool) bạn cần tải và cofig các thư viện cần thiết

Set Cygwin64: -> ->->-> chọn <https://cygwin.mirrors.hoobly.com> ->  search thử viện cần tải chọn phiên bản và tích vào next ->finish

Library (thiếu thì khi chạy repo sẽ lỗi có thể add thử viện theo lỗi):

* Zip
* Curl

# **Component and template**

## **Component**

**Example :**

* **package com.myapp.core.models.** **GridImageModel (**List Nested list)(path in to DAM Assest to get link image)
* **package com.myapp.core.models.NavModel (List)**
* Component Content/tab **(dialog tab and list)(to handle you need write model to map with attribute on dialog)**

Key word:

* Sling model ( to create model)
* Resource ( to get Resource on Tree structure of base resouce)
* [Form — Granite UI 1.0 documentation (adobe.com)](https://developer.adobe.com/experience-manager/reference-materials/6-5/granite-ui/api/jcr_root/libs/granite/ui/components/coral/foundation/form/index.html) ( To write Dialog in XML for Component)

|  |  |
| --- | --- |
| **@Model** | This is the primary annotation used to declare a **Sling Model**. It is used at the class level to define the adaptable classes (usually Resource or SlingHttpServletRequest) and the resource types that the model is applicable for |
| **@Inject** | Used to inject a property value from the current resource or request into a field or method. It supports injection of simple types, arrays, collections, and even other Sling Models. |
| **@Optional** | This annotation is used alongside **@Inject** to indicate that the injection is not mandatory. If the property is not found, the field will be left null (or with its default value) without causing an exception. |
| **@Named** | Allows specifying the exact property name to be injected, especially useful when the field name in the Java model does not match the property name in the JCR |
| **@Default** | Provides default values for injected fields if the property is not present in the resource. It can be used with simple types and arrays |
| **@Via** | Specifies an intermediary to use when injecting a property. It can be used to adapt to another type or use a resource superType for injection |
| **@PostConstruct** | Designates a method that should be called after all injection has occurred but before the model is returned for use. It's useful for initialization logic that requires injected fields to be set |
| **@ResourcePath** | Allows injecting a resource into a model field based on a path. This is particularly useful for loading fixed resources. |
| **@ChildResource** | Injects a child resource relative to the current resource. Useful for hierarchical data structures. |
| **@Self** | Injects the adaptable instance itself, often used to re-inject a request or resource with different adaptations. |
|  |  |

Note: SlingHttpServletRequest and Sling.Resource