

Working with AEM in IDEs



Tyler Maynard

AEM DEVELOPER

@TylersDesk www.tylermaynard.com



Overview



Setting up Eclipse and IntelliJ for development

Running Maven builds from our IDEs

Debugging AEM with Eclipse and IntelliJ

AEM specific plugins



Configuring IDEs for AEM Development



Importing and Configuring

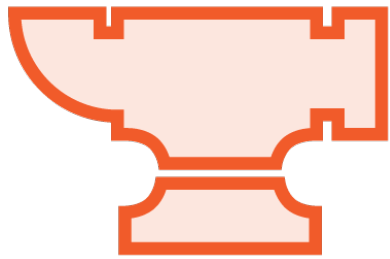


Load Maven project

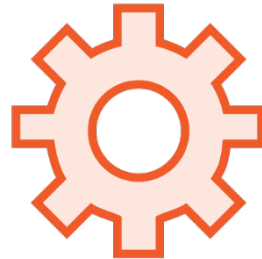


Configure IDE for our builds

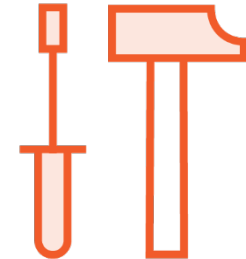
Configuring IDE



JDK



Maven



Run Configuration

Loading and Configuring an AEM Project with Eclipse



Demo



Load project into Eclipse

Configure JDK & Maven

Setup a run configuration

Build entire project

Build specific module



Loading and Configuring an AEM Project with IntelliJ



Demo



Import project into IntelliJ

Configure JDK & Maven

Running and activating builds

Build entire project

Build specific module



Note for Students Using AEM 6.2



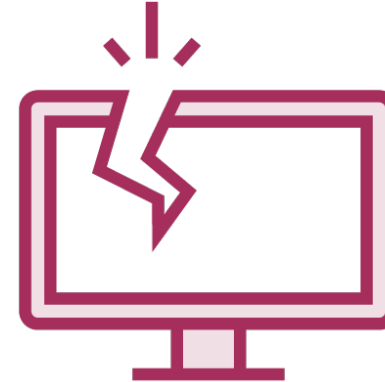
Missing dependency in
aem-project-archetype
projects that are using
AEM 6.2



Broken Component in AEM 6.2



Missing Dependency



Broken Component

Missing Dependency

```
<dependency>  
  <groupId>org.apache.geronimo.specs</groupId>  
  <artifactId>geronimo-atinject_1.0_spec</artifactId>  
  <version>1.0</version>  
  <scope>provided</scope>  
</dependency>
```



Demo



Investigate broken component

Look at OSGi bundle from project

Use AEM's depfinder

Add missing dependency to POM.xml

Rebuild our project



Debugging Java in AEM



Steps to Setup Debugging



Set AEM to accept debugger connections



Setup an IDE debug configuration



JVM Options to Enable Debug

-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=3000



Create Break Points in the IDE



Set a break point



Server will stop executing at a break

Demo



Locate AEM start scripts

Add debug JVM options

Start AEM via command line



Debugging AEM with Eclipse



Demo



Create a debug configuration

Add a breakpoint in our code

Visit a page that calls our breakpoint

Step through the code



Debugging AEM with IntelliJ



Demo



Create a debug configuration

Add a breakpoint in our code

Visit a page that calls our breakpoint

Step through the code



AEM Developer Tools for Eclipse



AEM Developer Tools Explained



AEM perspective



View JCR properties



Sync content to and
from JCR

Demo



Install AEM developer tools for Eclipse

Connect AEM to our AEM developer tools



AEM Perspective in AEM Developer Tools



Demo



Convert modules to content project
Investigate JCR properties in Eclipse



Syncing Content with AEM Developer Tools



Demo



Syncing files on save

Importing and exporting to the JCR



IntelliVault for IntelliJ



Demo



Install IntelliVault

Importing and exporting to the JCR



Summary



Configuring IDEs for development

Built AEM Maven projects in IDEs

Debugging in IDEs

IDE tools for AEM development

