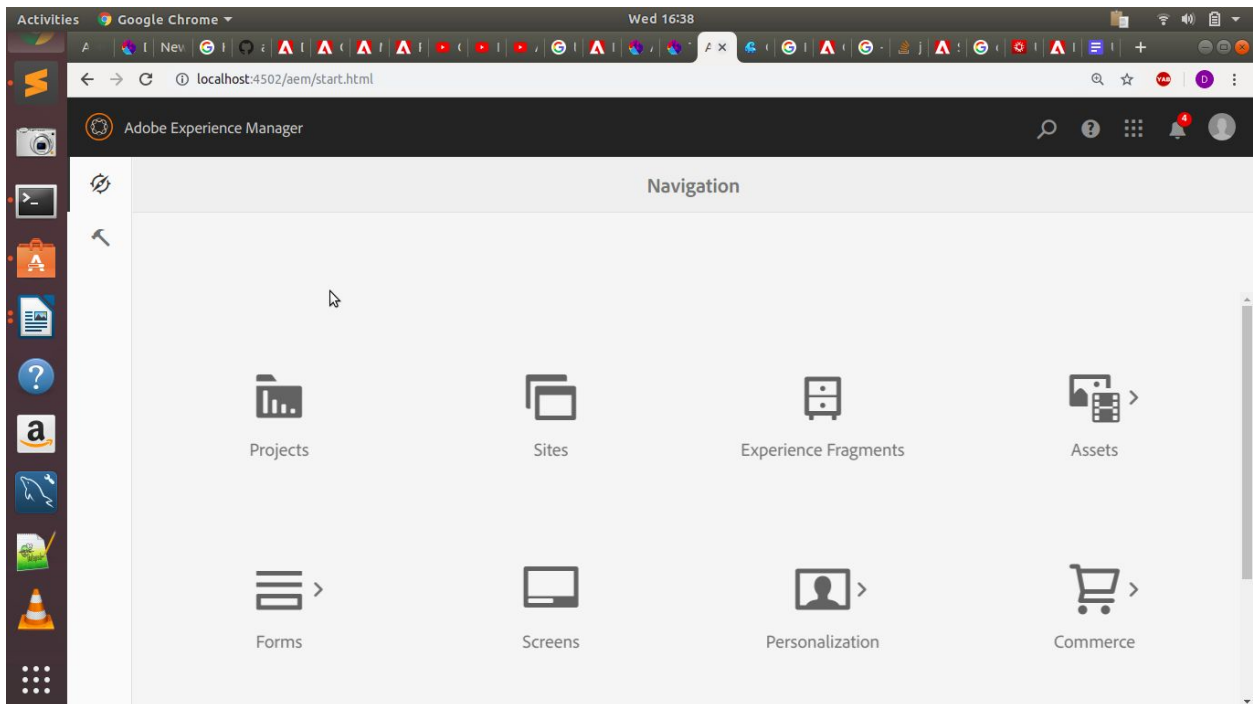


1. Open AEM in debug mode

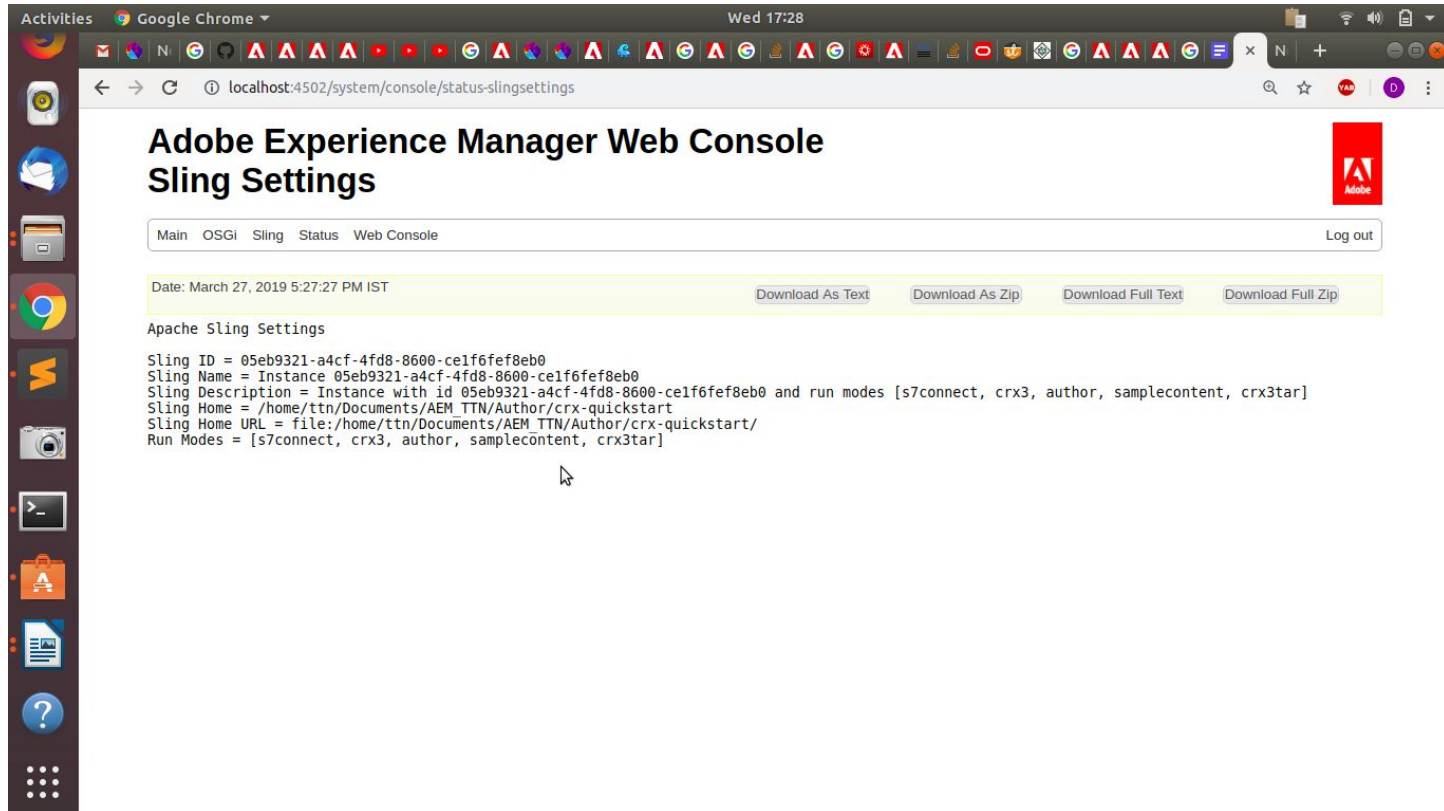
```
ttn@ttn:Author $ java -jar cq-quickstart-6.4.0.jar -fork -forkargs -- -Xdebug -Xrunjdp:transport=dt_socket,address=59865,suspend=n,server=y -Xmx1520m -XX:MaxPermSize=512m -XX:-UseSplitVerifier
Loading quickstart properties: default
Loading quickstart properties: instance
Low-memory action set to fork
Using 64bit VM settings, min.heap=1024MB, min permgen=256MB, default fork arguments=[-Xmx1024m, -XX:MaxPermSize=256m]
The JVM reports a heap size of 1756 MB, meets our expectation of 1024 MB +/- 20
Preparing to fork JVM, OS name=Linux, isWindows=false
Forking JVM: [/home/ttn/.sdkman/candidates/java/8.0.202-amzn/jre/bin/java, -Xdebug, -Xrunjdp:transport=dt_socket,address=59865,suspend=n,server=y, -Xmx1520m, -XX:MaxPermSize=512m, -XX:-UseSplitVerifier, -jar, /home/ttn/Documents/AEM_TTN/Author/cq-quickstart-6.4.0.jar, -nofork, -pt, CHILD, -fork, -forkargs, --, -Xdebug, -Xrunjdp:transport=dt_socket,address=59865,suspend=n,server=y, -Xmx1520m, -XX:MaxPermSize=512m, -XX:-UseSplitVerifier]
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=512m; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option UseSplitVerifier; support was removed in 8.0
Listening for transport dt_socket at address: 59865
Loading quickstart properties: default
Loading quickstart properties: instance
Low-memory action set to fork
Using 64bit VM settings, min.heap=1024MB, min permgen=256MB, default fork arguments=[-Xmx1024m, -XX:MaxPermSize=256m]
The JVM reports a heap size of 1351 MB, meets our expectation of 1024 MB +/- 20
Not forking JVM as -nofork option is set
Setting properties from filename '/home/ttn/Documents/AEM_TTN/Author/cq-quickstart-6.4.0.jar'
Verbose option not active, closing stdin and redirecting stdout and stderr
Redirecting stdout to /home/ttn/Documents/AEM_TTN/Author/crx-quickstart/logs/stdout.log
Redirecting stderr to /home/ttn/Documents/AEM_TTN/Author/crx-quickstart/logs/stderr.log
Press CTRL-C to shutdown the Quickstart server...
```



2. How to identify if AEM is Publish or Author?

i). Got to the below mentioned url to find out the AEM Status

<http://localhost:4502/system/console/status-slingsettings>



```
<%  
pageContext.setAttribute("runModes",  
sling.getService(SlingSettingsService.class).getRunModes().toString());  
%>
```

Solution

You can use the following code in your components and templates to find out if the running instance is an author or a publish

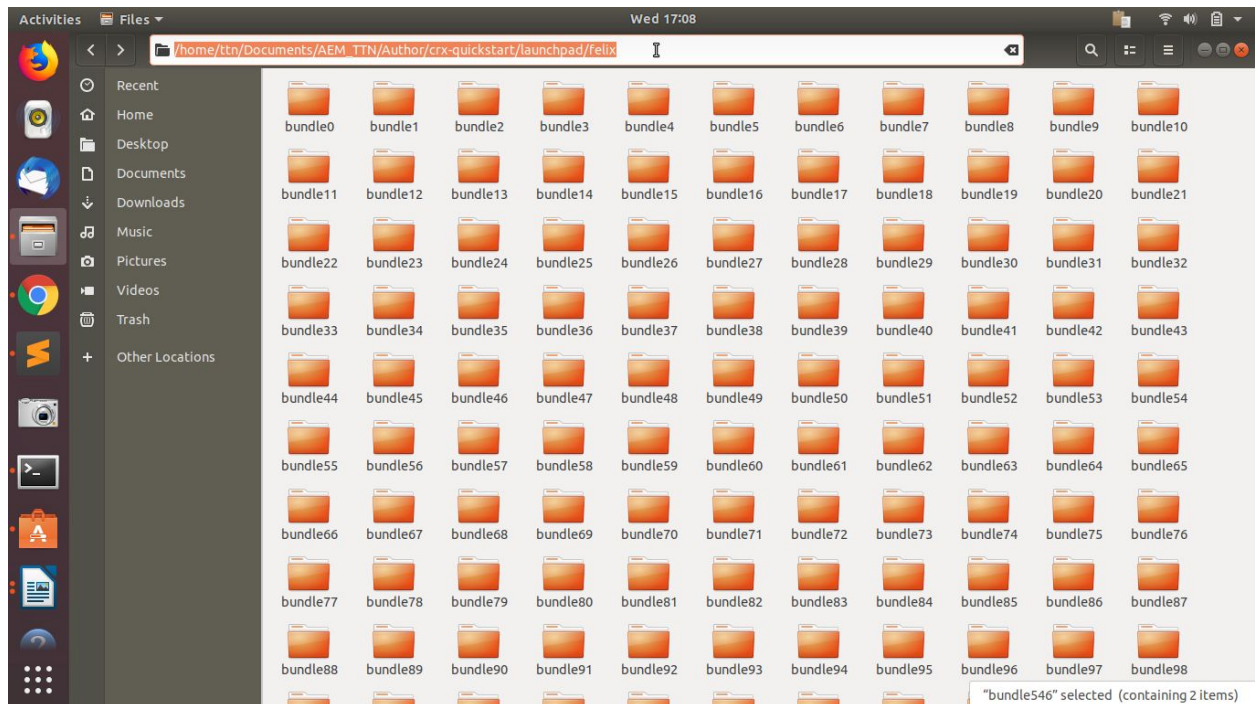
```
DeliveryHttpServletRequest cqRequest =  
(DeliveryHttpServletRequest) request;  
  
// flag to indicate whether instance is Author or Publish  
boolean isAuthor =  
!cqRequest.getCmsService().getDisplayControlSet(cqRequest).equals(  
DisplayControlSet.SHOW);  
  
// flag to indicate whether instance is in preview/show mode  
boolean isShowMode =  
(cqRequest.getCmsService().getDisplayControlSet(cqRequest).equals(  
DisplayControlSet.SHOWMODE) ||  
"ShowMode".equals(cqRequest.getParameter("Show"))) ;
```

This method is better than checking the URL, because this code does not rely on a specific context, which can change. Also, the author server can be switched to a publish mode by setting parameter "Show" to 1 in the request. So, the request to /author/somepage.html?Show=1 renders as publish. Therefore, relying on the context is not a reliable solution.

3. a). Where are all the bundles installed in AEM resides in system?

Ans. All the bundles installed in AEM resides in system at the location:

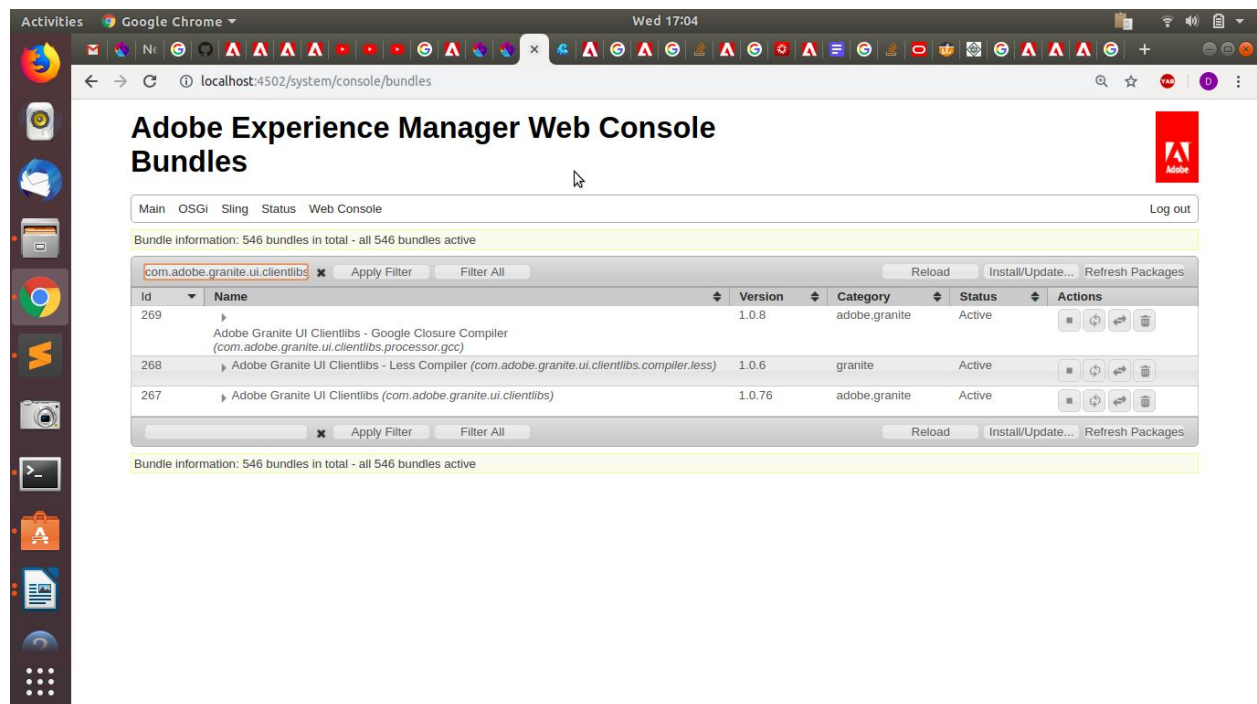
crx-quickstart/launchpad/felix



b). Search for bundle “*com.adobe.granite.ui.clientlibs*” and decompile it

i) Search for `com.adobe.granite.ui.clientlibs` in

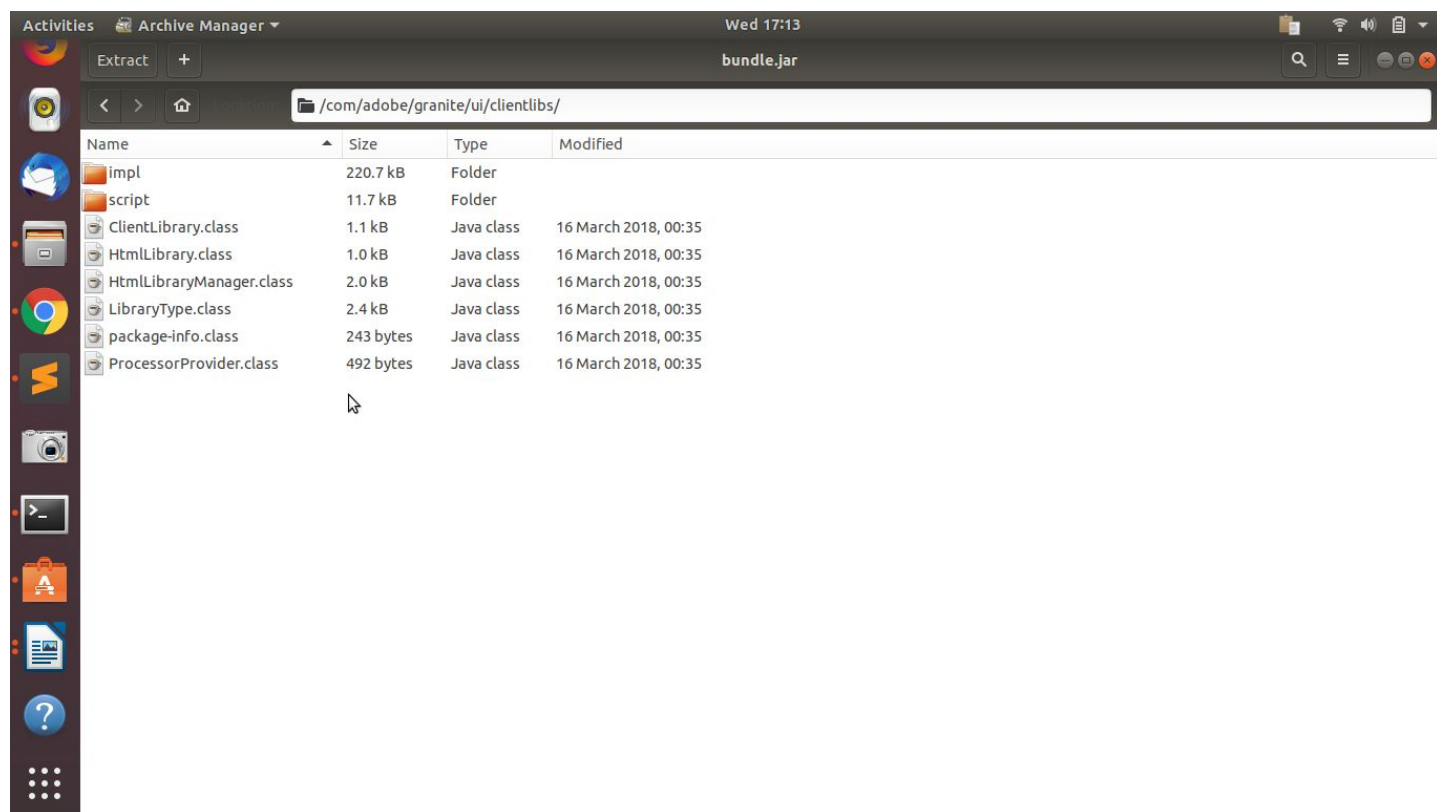
<http://localhost:4502/system/console/bundles>



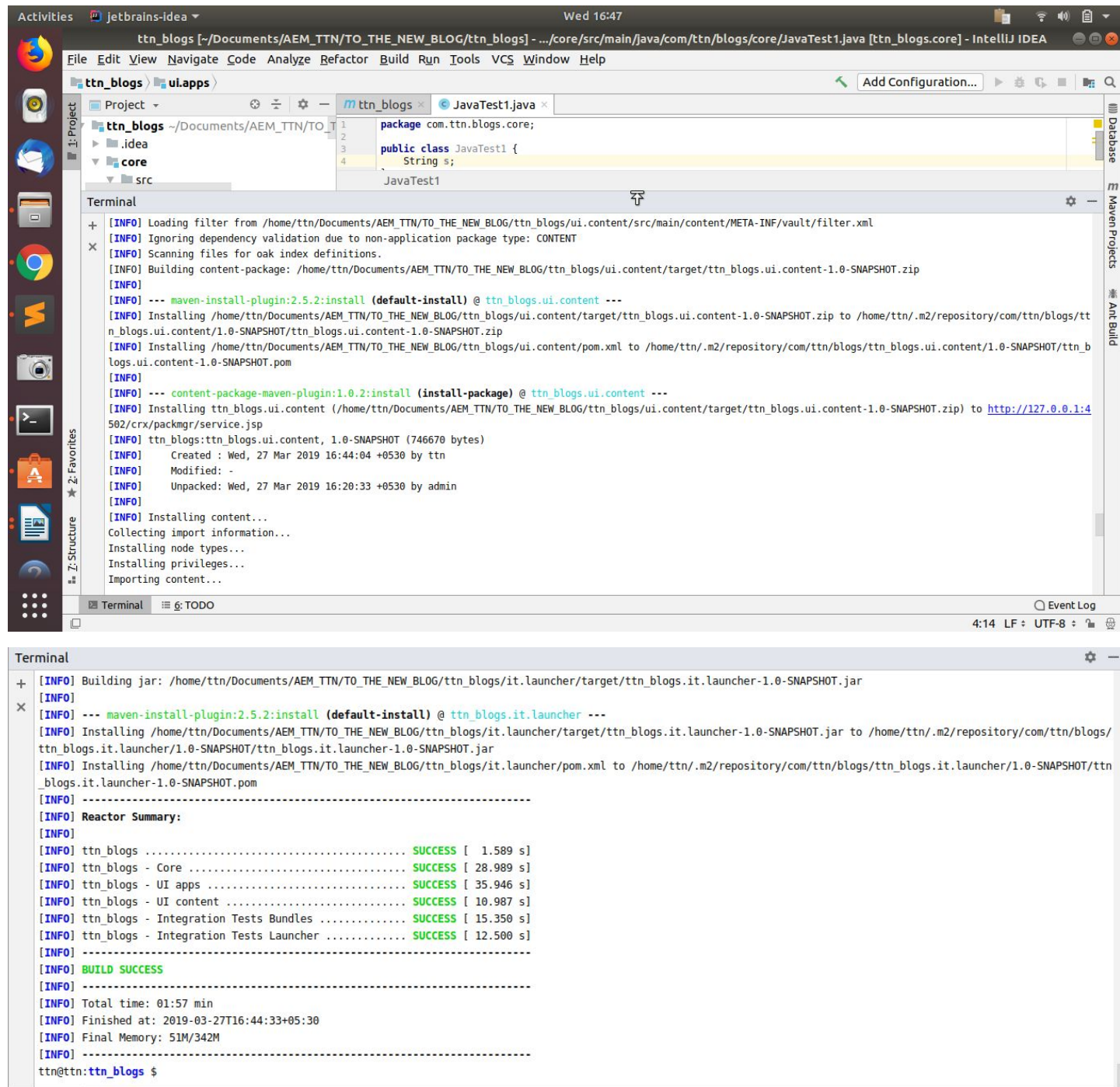
ii) Now search for bundle 267 in `crx-quickstart/launchpad/felix`

iii) Search for:

`crx-quickstart/launchpad/felix/bundle267/version0.0`
`/com/adobe/granite/ui/clientlibs/`




```
[INFO] -----
ttn@ttn:ttn_blogs $ mvn clean install -PautoInstallPackage -Daem.host=127.0.0.1 -Daem.port=4502
```



Activities Google Chrome Wed 16:48

127.0.0.1:4502/crx/de/index.jsp#/apps/ttn_blogs/install/ttn_blogs.core-1.0-SNAPSHOT.jar/jcr%3Acontent

CRXDE | Lite

Save All Create ... Delete Copy Paste Move ... Rename ... Overlay Node ... Mixins ... Tools Administrator@crx.default

/apps/ttn_blogs/install/ttn_blogs.core-1.0-SNAPSHOT.jar/jcr:content

Home

CRXDE | Lite

Content Repository Extreme Development Environment Lite

Enter search term to search the repository

Repository Information

Apache Jackrabbit Oak 1.8.2 by
The Apache Software Foundation

Developer Resources

- Documentation
- Developer Blog
- Knowledge Base

Properties Access Control Replication Console Build Info

	Name	Type	Value	Protected	Mandatory	Multiple	Auto Created
1	jcr:data	Binary	view	false	true	false	false
2	jcr:lastModified	Date	2019-03-27T16:20:22.225+05:30	false	false	false	true
3	jcr:lastModifiedBy	String	admin	false	false	false	true
4	jcr:mimeType	String	application/java-archive	false	false	false	false
5	jcr:primaryType	Name	nt:resource	true	true	false	true
6	jcr:uuid	String	3c6988c8-de6c-4f5b-a283-b30c4738bc34	true	true	false	true

Name Type String Value Multi Add >>

5. Why do we specify SNAPSHOT in the project version?

Snapshot is a special version which indicates the current development copy of the project which is being worked on. For each build, maven always checks out for a SNAPSHOT of the project.

Hence, whenever maven finds a newer SNAPSHOT of the project, it downloads and replaces the older .jar file of the project in the local repository

A snapshot version in Maven is one that has not been released.

The idea is that before a 1.0 release (or any other release) is done, there exists a 1.0-SNAPSHOT. That version is what might become 1.0. It's basically "1.0 under development". This might be close to a real 1.0 release, or pretty far (right after the 0.9 release, for example).

The difference between a "real" version and a snapshot version is that snapshots might get updates. That means that downloading 1.0-SNAPSHOT today might give a different file than downloading it yesterday or tomorrow.

Usually, snapshot dependencies should only exist during development and no released version (i.e. no non-snapshot) should have a dependency on a snapshot version.

https://docs.oracle.com/middleware/1212/core/MAVEN/maven_version.htm#MAVEN402