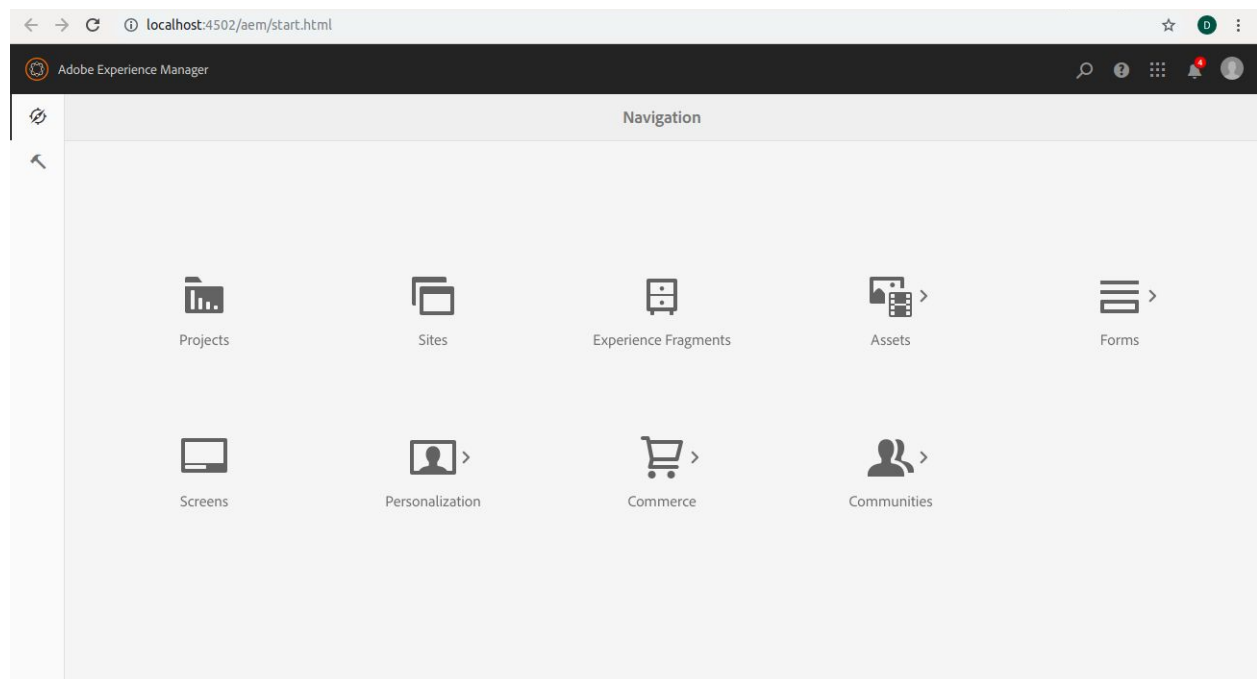


# Introduction to AEM

## Q.1. Open AEM in debug mode.

```
Redirecting stdout to /home/ttn/ttnAem/crx-quickstart/logs/stdout.log
Redirecting stderr to /home/ttn/ttnAem/crx-quickstart/logs/stderr.log
Press CTRL-C to shutdown the Quickstart server...
^C
ttn@ttn:ttnAem $
ttn@ttn:ttnAem $ java -jar c
ttn@ttn:ttnAem $ java -jar cq-quickstart-6.4.0.jar crx-quickstart/
ttn@ttn:ttnAem $ java -jar cq-quickstart-6.4.0.jar -debug 10123
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 3548 MB, meets our expectation of 1024 MB +/- 20
Debugging requested on port 10123, forcing fork
Preparing to fork JVM, OS name=Linux, isWindows=false
Forking JVM: [/home/ttn/.sdkman/candidates/java/8.0.202-amzn/jre/bin/java, -agentlib:jdwp=transport=dt_socket,address=10123,server=y,suspend=n, -Xmx1024m, -XX:MaxPermSize=256m, -jar, /home/ttn/ttnAem/cq-quickstart-6.4.0.jar, -nofork, -pt, CHILD]
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=256m; support was removed in 8.0
Listening for transport dt_socket at address: 10123
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]
**** WARNING: Insufficient heap memory ****
The JVM reports 910 MB but we recommend at least 1024 MB +/- 20
Use your JVM's heap size option (like -Xmx1024M) to set that size.
Will fork a JVM to get enough memory.
*****
Available memory below specified limits and low-memory action set to fork, will fork to get enough memory
Not forking JVM as -nofork option is set
Setting properties from filename '/home/ttn/ttnAem/cq-quickstart-6.4.0.jar'
Verbose option not active, closing stdin and redirecting stdout and stderr
Redirecting stdout to /home/ttn/ttnAem/crx-quickstart/logs/stdout.log
Redirecting stderr to /home/ttn/ttnAem/crx-quickstart/logs/stderr.log
Press CTRL-C to shutdown the Quickstart server...
```



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

We can run the following command to identify the current instance we are running.

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

Also, we can check it using run modes as shown below:



The screenshot shows the Adobe Experience Manager Web Console Sling Settings page. The browser address bar indicates the URL is localhost:4502/system/console/status-slingsettings. The page title is "Adobe Experience Manager Web Console Sling Settings". The navigation bar includes links for Main, OSGi, Sling, Status, and Web Console, with a "Log out" button on the right. Below the navigation bar, there is a date stamp "Date: March 27, 2019 5:27:22 PM IST" and four download buttons: "Download As Text", "Download As Zip", "Download Full Text", and "Download Full Zip". The main content area is titled "Apache Sling Settings" and displays the following information:

```
Sling ID = f527f7e3-4bb4-4a0b-9d58-bb1f10036f9d  
Sling Name = Instance f527f7e3-4bb4-4a0b-9d58-bb1f10036f9d  
Sling Description = Instance with id f527f7e3-4bb4-4a0b-9d58-bb1f10036f9d and run modes [s7connect, crx3, author, samplecontent, crx3tar]  
Sling Home = /home/ttn/ttnAem/crx-quickstart  
Sling Home URL = file:/home/ttn/ttnAem/crx-quickstart/  
Run Modes = [s7connect, crx3, author, samplecontent, crx3tar]
```

**Q.3. Where are all the bundles installed in AEM resides in system? Search for bundle “com.adobe.granite.ui.clientlibs” and decompile it.**

All Bundles in AEM reside in :

**crx-quickstart/launchpad/felix**



```
Terminal x Terminal x Terminal x Terminal x Terminal x Terminal x Terminal x Terminal x
ttn@ttn:tothenewBlog $ cd ~
ttn@ttn:~ $ ls
AEMBootcamp  assignments  divva_clone  Downloads    IdeaProjects  MajorProject  Pictures      Templates
aemImages    Bootcamp    divva_ttn    examples.desktop  IntelliJ      Music         Public       ttnAem
aem_maven    Desktop     Documents    Firefox_wallpaper.png KnowledgeSession.groovy myfile.txt   publishDemo  Videos

ttn@ttn:~ $ cd ttnAem/
ttn@ttn:ttnAem $ ls
cq-quickstart-6.4.0.jar  crx-quickstart  EXAMPLE1  license.properties  org.apache.sling.starter-11.jar  sling  tothenewBlog

ttn@ttn:ttnAem $ cd crx-quickstart/
ttn@ttn:crx-quickstart $ ls
app  conf  eula-en_US.html  eula-ja_JP.html  logs  monitoring  readme.txt  threaddumps
bin  eula-de_DE.html  eula-fr_FR.html  launchpad  metrics  opt  repository

ttn@ttn:crx-quickstart $ cd launchpad/
ttn@ttn:launchpad $ ls
conf  config  felix  installer  org.apache.sling.launchpad.base.jar  sling_bootstrap.txt  startup

ttn@ttn:launchpad $ cd felix/
ttn@ttn:felix $ ls
bundle0  bundle14  bundle181  bundle222  bundle265  bundle306  bundle348  bundle39  bundle430  bundle472  bundle513  bundle61
bundle1  bundle140  bundle182  bundle223  bundle266  bundle307  bundle349  bundle390  bundle431  bundle473  bundle514  bundle62
bundle10  bundle141  bundle183  bundle224  bundle267  bundle308  bundle35  bundle391  bundle432  bundle474  bundle515  bundle63
bundle100  bundle142  bundle184  bundle225  bundle268  bundle309  bundle350  bundle392  bundle433  bundle475  bundle516  bundle64
bundle101  bundle143  bundle185  bundle226  bundle269  bundle31  bundle351  bundle393  bundle434  bundle476  bundle517  bundle65
bundle102  bundle144  bundle186  bundle227  bundle27  bundle310  bundle352  bundle394  bundle435  bundle477  bundle518  bundle66
bundle103  bundle145  bundle187  bundle228  bundle270  bundle311  bundle353  bundle395  bundle436  bundle478  bundle519  bundle67
bundle104  bundle146  bundle188  bundle229  bundle271  bundle312  bundle354  bundle396  bundle437  bundle479  bundle52  bundle68
bundle105  bundle147  bundle189  bundle23  bundle272  bundle313  bundle355  bundle397  bundle438  bundle48  bundle520  bundle69
bundle106  bundle148  bundle19  bundle230  bundle273  bundle314  bundle356  bundle398  bundle439  bundle480  bundle521  bundle7
```

Extract	+	bundle.jar			🔍	☰	⌵	⌵	⌵
<div>&lt; &gt; 🏠 /com/adobe/granite/ui/clientlibs/</div>									
Name	▲	Size	Type	Modified					
impl		220.7 kB	Folder						
script		11.7 kB	Folder						
ClientLibrary.class		1.1 kB	Java class	16 March 2018, 00:35					
HtmlLibrary.class		1.0 kB	Java class	16 March 2018, 00:35					
HtmlLibraryManager.class		2.0 kB	Java class	16 March 2018, 00:35					
LibraryType.class		2.4 kB	Java class	16 March 2018, 00:35					
package-info.class		243 bytes	Java class	16 March 2018, 00:35					
ProcessorProvider.class		492 bytes	Java class	16 March 2018, 00:35					

← → ↺ ⓘ localhost:4502/system/console/bundles/267

☆ ⓘ ⋮

## Adobe Experience Manager Web Console Bundles

Main OSGi Sling Status Web Console

Log out

Bundle information: 548 bundles in total - all 548 bundles active

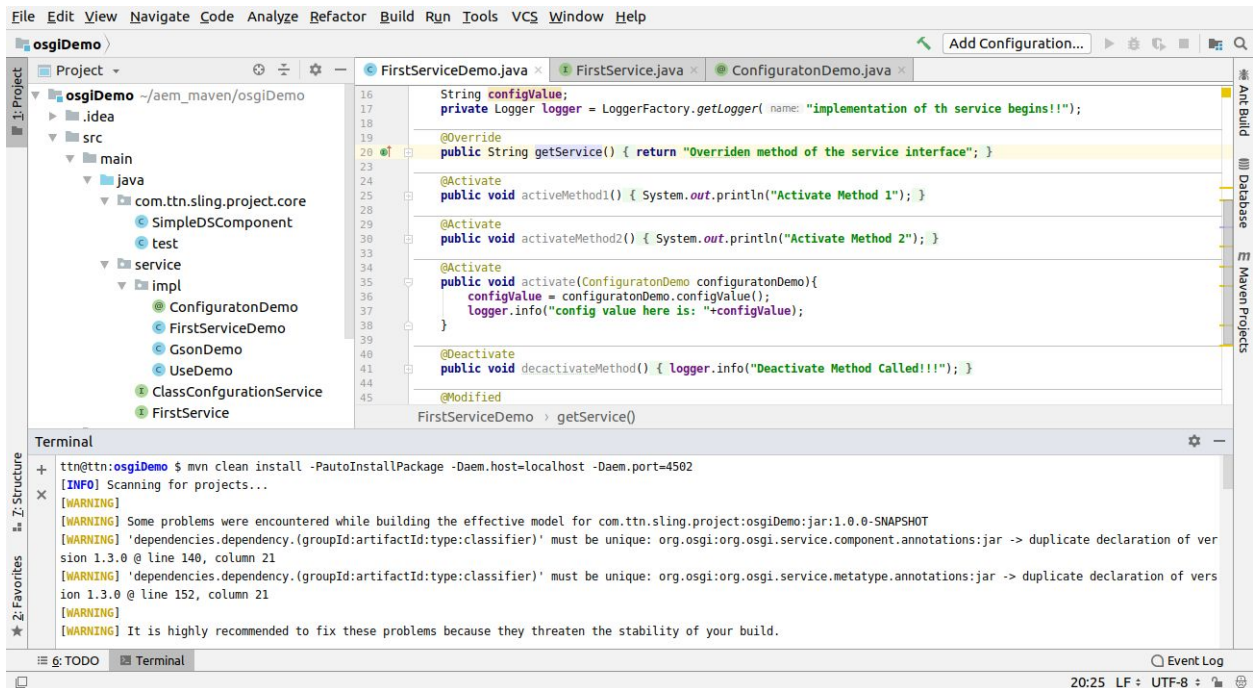
✕ Apply Filter Filter All

Reload Install/Update... Refresh Packages

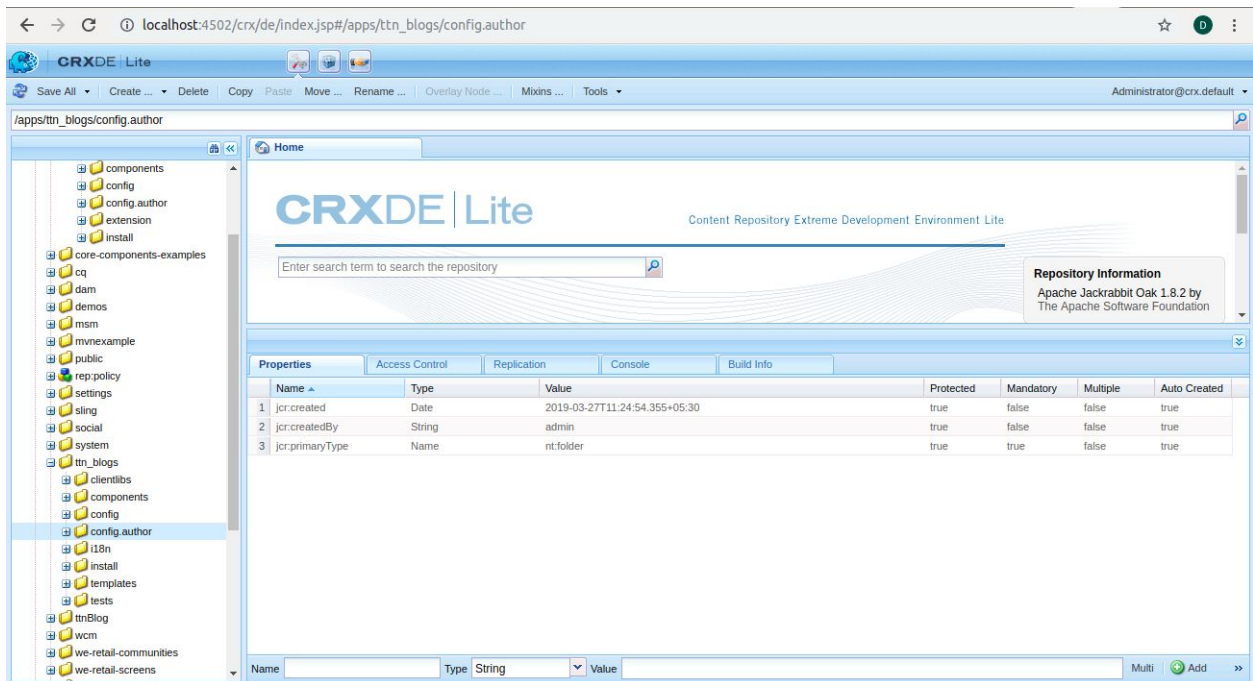
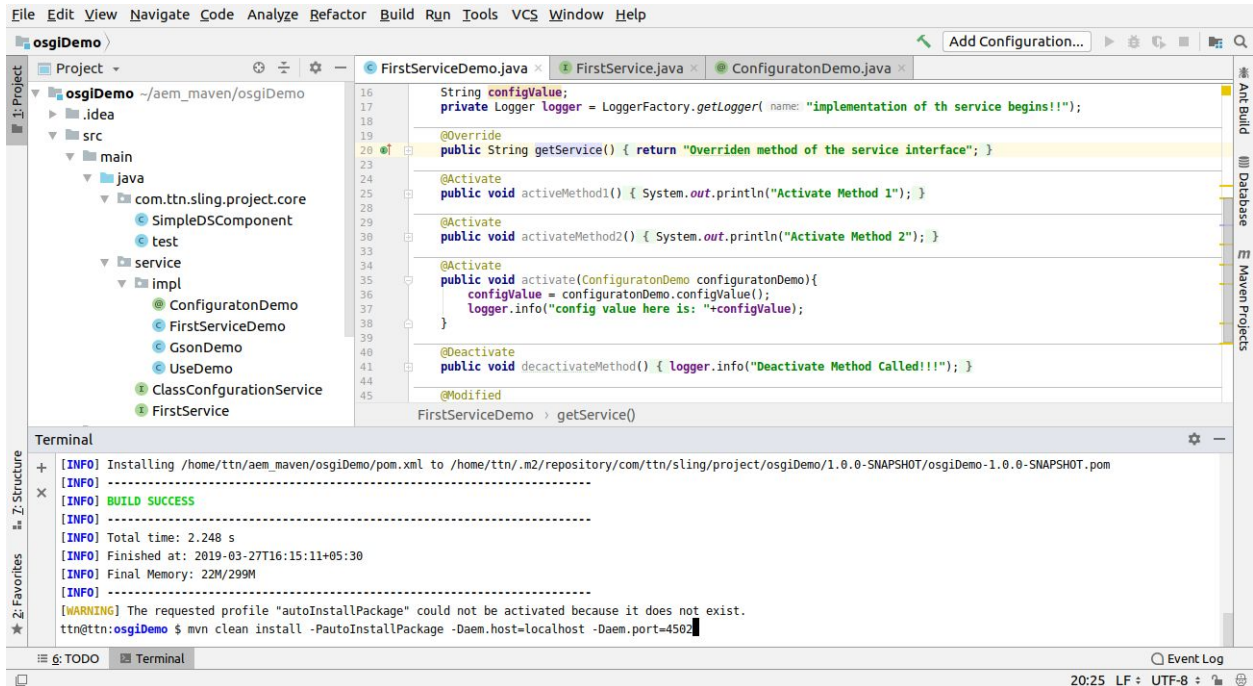
Id	Name	Version	Category	Status	Actions
267	Adobe Granite UI Clientlibs (com.adobe.granite.ui.clientlibs)	1.0.76	adobe.granite	Active	⌵ ⌵ ⌵ ⌵
	Symbolic Name	com.adobe.granite.ui.clientlibs			
	Version	1.0.76			
	Bundle Location	launchpad.resources/install/0/com.adobe.granite.ui.clientlibs-1.0.76.jar			
	Last Modification	Mon Mar 11 17:34:32 IST 2019			
	Bundle Documentation	https://wiki.corp.adobe.com/display/granite			
	Vendor	Adobe Systems Incorporated			
	Description	The Adobe Granite clientlibs manager			
	Start Level	20			
	Bundle Classpath	..yuicompressor-2.4.8-granite-003.jar			
	Exported Packages	com.adobe.granite.ui.clientlibs,version=1.2.0 com.adobe.granite.ui.clientlibs.script,version=1.2.1			
	Imported Packages	javax.annotation,version=0.0.0.1_008_JavaSE from org.apache.felix.framework (0) javax.jcr,version=2.0.0 from org.apache.sling.jcr-jcr-wrapper (127) javax.jcr,version=1.1.0 from org.apache.sling.jcr-jcr-wrapper (127) javax.jcr.observation,version=2.0.0 from org.apache.sling.jcr-jcr-wrapper (127) javax.jcr.observation,version=1.1.0 from org.apache.sling.jcr-jcr-wrapper (127) javax.jcr.query,version=2.0.0 from org.apache.sling.jcr-jcr-wrapper (127) javax.jcr.query,version=1.1.0 from org.apache.sling.jcr-jcr-wrapper (127) javax.servlet,version=2.6.0 from org.apache.felix.http.servlet-api (43) javax.servlet,version=3.1.0 from org.apache.felix.http.servlet-api (43) javax.servlet,version=3.0.0 from org.apache.felix.http.servlet-api (43) javax.servlet.http,version=2.6.0 from org.apache.felix.http.servlet-api (43) javax.servlet.http,version=3.1.0 from org.apache.felix.http.servlet-api (43) javax.servlet.http,version=3.0.0 from org.apache.felix.http.servlet-api (43) javax.swing,version=0.0.0.1_008_JavaSE from org.apache.felix.framework (0) javax.swing.border,version=0.0.0.1_008_JavaSE from org.apache.felix.framework (0) javax.swing.event,version=0.0.0.1_008_JavaSE from org.apache.felix.framework (0) javax.swing.filechooser,version=0.0.0.1_008_JavaSE from org.apache.felix.framework (0) javax.swing.table,version=0.0.0.1_008_JavaSE from org.apache.felix.framework (0) javax.swing.text,version=0.0.0.1_008_JavaSE from org.apache.felix.framework (0) javax.swing.tree,version=0.0.0.1_008_JavaSE from org.apache.felix.framework (0) org.apache.commons.io,version=1.4.9999 from org.apache.commons.io (15) org.apache.commons.lang,version=2.5.0 from org.apache.commons.lang2 (58) org.apache.jackrabbit.api.observation,version=2.3.1 from org.apache.jackrabbit.jackrabbit-api (118)			

**Q. 4. How can we specify in the command line the ‘port’ and ‘host’ while installing the project.**

java -jar cq-quickstart-6.4.0.jar -debug (portNo).







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

A large software application generally consists of multiple modules and it is common scenario where multiple teams are working on different modules of the same application. For example, consider a team is working on the front end of the application as app-ui project (app-ui.jar:1.0) and they are using data-service project (data-service.jar:1.0).

Now it may happen that team working on data-service is undergoing bug fixing or enhancements at a rapid pace and they are releasing the library to remote repository almost every other day.

Now if data-service team uploads a new version every other day, then the following problems will arise –

- data-service team should tell app-ui team every time they have released an updated code.
- app-ui team required to update their pom.xml regularly to get the updated version.

To handle such kind of situation, SNAPSHOT concept comes into play. **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. Snapshot version always gets some updates/changes being made on the project. Also, Snapshot should exist only during development phase and it will not be having a release version. This implies that the build of the project will be changed at any time and it is still under the development process.