What is the purpose of the core module in AEM?

The purpose of the core module is to provide the backend logic and include java classes and also has set of server side codes that is essential for server side programming and makes the process efficient.

What kind of files and code can be found in the core folder?

The kind of files and code that can be found includes java classes which is servlets and models. The next set of files includes contents and models.

Explain the role of ui.apps in AEM projects.

The role of ui.apps is to primarily focus on the client side layer of the aem project. It includes the user interface and includes the components and also client libraries. The component in the ui.apps can be simple as an html file.

How are components structured in the ui.apps folder?

The Components is structured in such a way that in ui.apps folder, each component contains the Markupfile and an xml file. It also includes the client side libraries that bundles the css and javascript for styling.

**Hello World Component:**

* Where is the Hello World component located in both core and ui.apps?
  + In the core folder, Helloworld is located . In ui.apps it is located in C:\Users\05 PROJECTS\myTraining-main\myTraining\ui.apps\src\main\content\jcr\_root\apps\myTraining\components\helloworld
* Explain the Java class (in core) for the Hello World component.
  + The java class in core folder handles the backend logic where the javaclass is a sling model that generates the message. It displays the message “ hello world” and the path of the current page.
* How does the HTL script work in ui.apps for Hello World?
  + The htl script in the hello world defines the markup for rendering the page. It displays the output as hello world. The working of htl is to render dynamic content in the pages.
* How are properties and dialogs defined for this component?

What are the different types of AEM modules (core, ui.apps, ui.content, etc.)?

The different types of AEM modules include ui.apps which is responsible for the serverside; ui.apps which focuses on the client side and has clientlibs; ui.content module contains the content of the pages and static assets. The ui.frontend modules has frontend specific resources. The ui.test module is to test the frontend components and compile the test code.

How does Maven build these modules?

Maven builds these modules with multi module structure where the project contains a parent pom.xml file with overall dependencies. And each of the module(ui.apps, ui.frontend..) has it’s own pom.xml file which has specific built dependcies.

It is done by running the command mvn clean install from the root directory to build these modules.

Explain the build lifecycle of Maven in the context of AEM.

1. Clean – removes the old built artifacts(modules).
2. Validate – checks for the project structre.
3. Compile – compiles the source code.
4. Test – run units tests with junit.
5. Package – convert the code into .jar files.
6. Install – installs the package in local repo.
7. Deploy – deployment to server.

How are dependencies managed in pom.xml?

The dependencies are managed under the pom.xml file under the <dependencies> section where it is specified with the version required. The Scope is also included, while the default being the compile, where the dependency is need at all time. Other scopes like test and runtime can also be defined. There is no need to handle transitive dependencies.

Why is Maven used instead of other build tools?

Maven is used because of the well defined process of build cycle. It handles the dependencies very well which is the crucial part in any module. The main reason to use maven is because of the easy integration of maven with AEM as it has a big community that provides tools and plugins and easy integration.

What advantages does Maven offer for AEM development?

Maven has many advantages over other which includes automatic dependency handling, where it downloads the dependencies from a central repository.

Also, Maven handles transitive dependencies, Where when one dependency needs other dependencies, maven takes care of the other dependency.

How does Maven help in managing dependencies and plugins in AEM projects?

Maven helps in managing dependencies by automatically managing the third party libraries and components. It is done by getting from a central repository and also tools are available to build, pack and deploy the projects.

What does mvn clean install do in an AEM project?

The mvn calls for the maven, clean – clears the previously built artifacts and dependencies and install – call for installation by compiling, testing and packing it into jar files.

How to deploy packages directly to AEM using Maven commands?

The Command mvn clean install -PautoInstallPackage is used to deploy packages directly to AEM.

Explain the purpose of different Maven profiles in AEM (autoInstallPackage, autoInstallBundle).

Different maven profiles helps to control the build process at different stages. The autoinstallpackage is a profile that is used to automatically deploy a content package to a AEM after the build is completed.

Autoinstallbundle automates the process of deployment of osgi bundles into the AEM. Because some of the Javaclasses might need to be installed into the osgi container.

What is the purpose of dumplibs in AEM?

The dumplibs is used to list all the libraries and dependencies that are bundled with the project. Also if there are any issues with a particular bundle, the dumplibs command can be used to identify the problems.

How can you view client libraries using dumplibs?

By using the command mvn clean install -Dclientlib.dump = true. Which will print all the details about the client libraries that are used and their contents with the dependencies.

Explain how client libraries are structured in AEM.

The Client libraries are stored in the clientlib folder and it consists of javascript and css files which are bundled together. They are bundled for the efficient loading process into the aem when a page is rendered.