Sling Models

* Used as replacement for WCMUsePojo’s since AEM 6.0
* Sling models/usePojo’s – both are used to map to a resource in JCR and allows us to access the resource’s properties
* Annotation driven
* @Inject is used to inject all the required objects (PageManager, services ….)
* Service ranking is important because Sling Models injectors are invoked in order of their service ranking, from lowest to highest.

**Dependency in pom.xml**

<dependency>

<groupId>org.apache.sling</groupId>

<artifactId>org.apache.sling.models.api</artifactId>

<version>1.3.2</version>

<scope>provided</scope>

</dependency>

**Maven bundle plugin**

<plugin>

<groupId>org.apache.felix</groupId>              <artifactId>maven-bundle-plugin</artifactId>

               <extensions>true</extensions>

              <configuration>

                   <instructions>

                       <Sling-Model-Packages>

                           com.sling.models.core.models – location where you code exists in repository

                       </Sling-Model-Packages>

                   </instructions>

                 </configuration>

</plugin>

* <http://wemappauth-devi3-03:4502/system/console/adapters> - to check if the model is picked up or not.

**Available Injectors**

* After sling model 1.3.2, injection specific annotations are available
* Below are the various objects that can be injected using various annotations available as part of 1.3.2

|  |  |
| --- | --- |
| ScriptVariable | currentPage,pageManager,Design, PageProperties |
| ChildResource | provides specific child (by name) of a resource |
| ValueMapValue | Directly injects the property value of specified name from map<propertyname, value> on the resource |
| ResourcePath | used to inject a path/ injects a property from the resource whose value is valid jcr path |
| OSGiService | same as @Reference, to inject services |
| SlingObject | used for injecting objects belonging to sling family - request,response,scriptHelper,resourceResolver |
| RequestAttributes | to obtain the request parameters sent from Models with exact names used in sightly |

* The adaptables parameter specifies that this model can be adapted by the request. The adapters parameter allows the implementation class to be registered under the Card interface.

**Model Annotation and available elements**

@Model(

        adaptables = {SlingHttpServletRequest.class}, - which objects can adopt to this model

        adapters = {Byline.class}, - to make the model available only via interface (abstraction)

        resourceType = {BylineImpl.RESOURCE\_TYPE}, - <https://sling.apache.org/documentation/bundles/models.html#associating-a-model-class-with-a-resource-type-since-130>

        defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL

)

**Models Available in WEM for Reference**

TechnicalStandardsModel.java - <https://gitscm.cisco.com/projects/CDC/repos/wem-git/browse/osgi/CPDMAuthoring/src/main/java/com/cisco/wem/cpdm/models/TechnicalStandardsModel.java?at=BR_CPDM_FEATURE>

ResourceModel.java - <https://gitscm.cisco.com/projects/CDC/repos/wem-git/browse/osgi/CPDMAuthoring/src/main/java/com/cisco/wem/cpdm/models/ResourceModel.java?at=BR_CPDM_FEATURE>

Useful Links

<https://apiltamang.wordpress.com/2017/02/17/unit-testing-for-sling-models-using-aem-mocks/>

<https://sourcedcode.com/aem-sling-models-unit-test-junit-4-with-examples>

<http://www.sgaemsolutions.com/2017/08/deep-dive-on-sling-models-part-1.html>

<http://keysandstrokes.info/aem-apache-sling-testing/>

<https://vdocuments.mx/1-aem-tdd-its-so-boring-august-6-2015-2-testing-aem-as-jcr-based.html>

<https://stackoverflow.com/questions/36087728/get-reference-to-mockresourceresolverfactory?answertab=oldest#tab-top>

<https://cwiki.apache.org/confluence/display/SLING/Scripting+variables>

<https://blogs.perficient.com/2017/06/26/how-to-switch-from-wcmusepojo-to-sling-models-in-aem-part-three-custom-injectors/>