Unit Testing with AEM Mocks

Ask the AEM Community Expert – Stefan Seifert

PRO!VISION
SOFTWARE CRAFTSMANSHIP

About the Speaker

- AEM Developer
- Apache Sling PMC
- Maintainer of wcm.io
- CTO of pro!vision GmbH

PRO!VISION
SOFTWARE CRAFTSMANSHIP

https://www.pro-vision.de

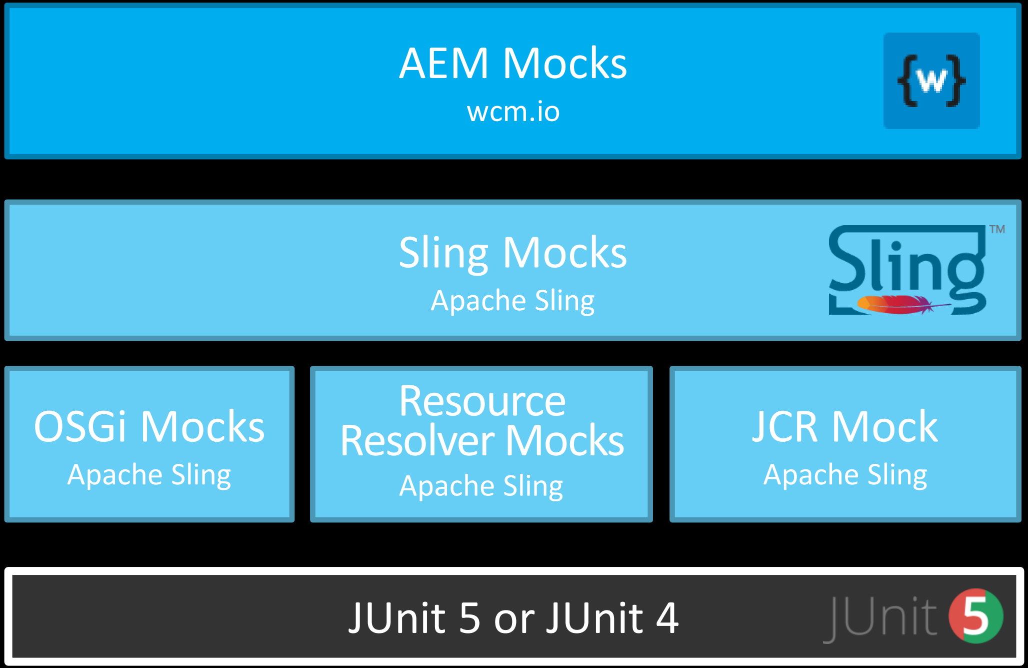
Stefan Seifert

About AEM Mocks

About AEM Mocks

- Unit Tests for your AEM Application
- Provides an in-memory AEM environment suitable for Unit Tests
- Covers 90% of what is required for typical AEM applications
- Can be combined with Mockito and others
- Fast test execution

Tech Stack



AEM Mocks
wcm.io

Sling Mocks
Apache Sling

OSGi Mocks
Apache Sling

Resource Resolver Mocks
Apache Sling

JCR Mock
Apache Sling

JUnit 5 or JUnit 4

What you can test

- All Java classes of your AEM project
  - OSGi Services, Sling Models, Servlets etc.
- Accessing content
  - Read/write resources in content repository
- Interaction with AEM and Sling APIs
  - AEM Sites and Assets, Sling API

What you cannot test

- Not supported:
  - Rendering your components and pages
  - Testing the Script output
  - Integration Tests, UI Tests

Choose Resource Resolver implementation

FASTER

Resource Resolver Type	Sling API	JCR API	Node Types	Obser-vation	JCR Query	Lucene Fulltext
RESOURCE RESOLVER_MOCK	✓	✗	✗	✗ (Sling only)	✗	✗
JCR_MOCK	✓	✓	✗	✗	✗ (mocked)	✗
JCR_OAK	✓	✓	✓	✓	✓	✗

more features

Using AemContext in JUnit 5

Use “junit5” package

Define @ExtendWith

```
import io.wcm.testing.mock.aem.junit5.AemContext;
import io.wcm.testing.mock.aem.junit5.AemContextExtension;

@ExtendWith(AemContextExtension.class)
class MyUnitTest {

  private AemContext context = new AemContext();

  @BeforeEach
  void setUp() { /*...*/ }

  @Test
  void testMyCode() { /*...*/ }

}
```

Optional:
RESOURCERESOLVER_MOCK
JCR_MOCK
JCR_OAK

Using AemContext in JUnit 4

Use “junit” package

Define
@Rule

```
import io.wcm.testing.mock.aem.junit.AemContext;

public class MyUnitTest {

  @Rule
  public AemContext context = new AemContext();

  @Before
  public void setUp() { /*...*/ }

  @Test
  public void testMyCode() { /*...*/ }

}
```

Optional:
RESOURCERESOLVER_MOCK
JCR_MOCK
JCR_OAK

AEM Mocks Features

OSGi Features

- Register and run OSGi services in a mocked OSGi environment
- Configuration, References
- Eventing

Sling Features

- Read and write resources
  - Simplified API for creating test content on-the-fly
  - Load test content from JSON files
- Simulate Sling Request and Response
- Sling Models
- Context-Aware Configuration

AEM Features

- Sites API: Page, Template, Component, Tag
- Assets API: Asset, Rendition
- Generate test pages and assets on-the-fly
- Run modes and WCM Modes
- Current page/current resource

Demo

- Test a Sling Model
- Test a Servlet
- Mock an OSGi reference

LIVE DEMO

Make your project ready for AEM Mocks

Adobe AEM Project Archetype

- If you have set up your project using the Adobe AEM Project Archetype you have to include AEM Mocks manually
- It’s easy, follow this guide https://wcm-io.atlassian.net/wiki/x/AQBiPg

Other projects

- Include the AEM Mock Dependency
- Recommended: Include wcm.io AEM Dependencies “Import” POM
- Proper configuration of maven-bundle-plugin, see Bundle Plugin FAQ
- Alternative: Use wcm.io AEM Archetype

“No OSGi SCR metadata” problem

- If your unit test fail with an exception “No OSGi SCR metadata found for class …”:
  - You have either an insufficient maven-bundle-plugin configuration
  - Or – if this happens only in Eclipse – make a “Project Clean” to ensure metadata is generated

See also https://wcm-io.atlassian.net/wiki/x/AYCJPg

Using AemContextBuilder

```java
@ExtendWith(AemContextExtension.class)
class MyUnitTest {

  private AemContext context = new AemContextBuilder()
      // register plugins
      .plugin(CACONFIG)
      .plugin(WCMIO_CACONFIG)
      // shared context setup code for all tests
      .<AemContext>afterSetUp(context -> {
        // register sling models
        context.addModelsForPackage("com.myproject");
      })
      .build();

  @BeforeEach
  void setUp() { /*...*/ }
  @Test
  void testMyCode() { /*...*/ }
}
```

Using **`AemContextBuilder`** not possible when passing in as method parameters

Recommendation: make this code reusable in a separate class

Use AEM Mocks Context Plugins

- Some functionality that is not included in AEM Mocks by default can be added using “Context Plugins”
- List of AEM Context Plugins: https://wcm-io.atlassian.net/wiki/x/EQBjPg

References

References 1/2

- wcm.io AEM Mocks
  http://wcm.io/testing/aem-mock/

- adaptTo() Talks
  adaptTo() 2018 Talk: JUnit 5 and Sling/AEM Mocks
  adaptTo() 2016 Talk: Unit Testing with Sling & AEM Mocks
  adaptTo() 2014 Lightning Talk: Mock AEM & Co for Unit Tests

- Apache Sling Mocks
  http://sling.apache.org/documentation/development/sling-mock.html
  http://sling.apache.org/documentation/development/osgi-mock.html
  http://sling.apache.org/documentation/development/jcr-mock.html

- Demo code for this session
  https://github.com/stefanseifert/2019-atace-unit-testing-with-aem-mocks

References 2/2

- Need Help? Use the Mailing Lists
  http://wcm.io/mailing-lists.html
- Trouble Shooting Articles
  How to use AEM Mocks in Adobe AEM Project Archetype
  Migrate AEM Mocks Unit Tests from JUnit 4 to JUnit 5
  AEM Mocks fails with "No OSGi SCR metadata found for class ..."
  List of AEM Mock Context Plugins
- Examples for Unit Tests using AEM Mocks:
  Take a look at the repositories in https://github.com/wcm-io - most of them contains lot's of test code using AEM Mocks

adaptTo() 2019 in Berlin / Germany

- Conference for Apache Sling & AEM
  https://adapt.to/
- Speakers wanted!
  https://adapt.to/cfp

PRO!VISION
SOFTWARE CRAFTSMANSHIP