



adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 26-28 SEPTEMBER 2016

Sling Http Testing Framework
Valentin Olteanu, Andrei Dulvac - Adobe Basel

About

An open-source framework

<https://github.com/apache/sling/tree/trunk/testing/http/clients>

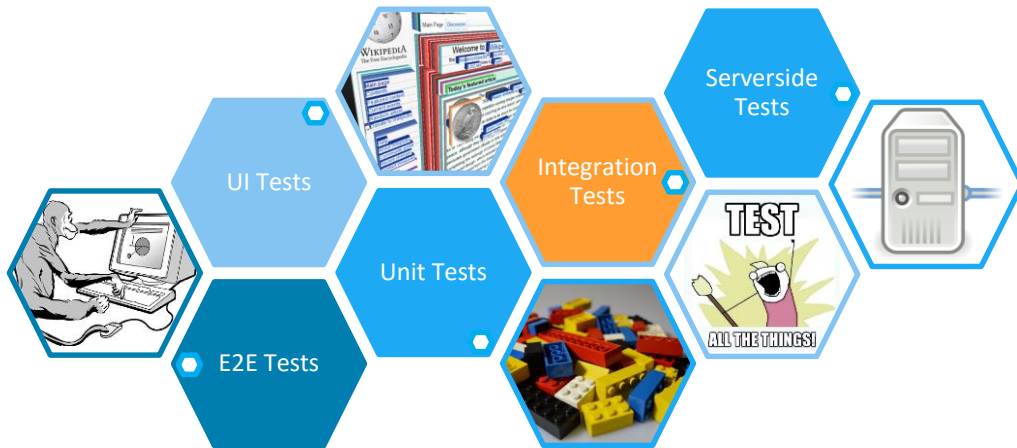
used in

<https://github.com/apache/sling/tree/trunk/testing/junit/rules>

replaces

<https://github.com/apache/sling/tree/trunk/testing/tools>

Yet another testing framework?



Testing:

- At http level
- Sling topologies
- Existing instances
- Apps on top of Sling
- Performance

Design principles

- Easy to use
- Sensible defaults
- Customizable
- Specialized
- Extensible
- Thread-safe

```
@Immutable
public class AbstractSlingClient implements HttpClient {
    ...
}

public class SlingClient extends AbstractSlingClient {
    ...
    public SlingHttpResponse createNode(
        final String path,
        final String nodeType) throws ClientException {
    }
    ...
}
```

How to use

How to use the clients

■ Instantiating

```
SlingClient c1 = new SlingClient(URI.create("localhost:8080"), "admin", "admin");
SlingClient c2 = SlingClient.Builder.create("localhost:8080", "admin", "admin").build();
OsgiConsoleClient c3 = c2.adaptTo(OsgiConsoleClient.class);
```

■ Using

```
SlingHttpResponse response = c1.createNode("/content/myNode", "nt:unstructured", 201);
```

■ Extending

```
public class MyClient extends SlingClient {
    public MyClient(URI server, String user, String password)...
    public MyClient(CloseableHttpClient h, SlingClientConfig c)...
    // optional
    public final static class Builder extends InternalBuilder<MyClient>...
}
```

How to use the junit rules

```
public class MySimpleIT {
    @ClassRule
    public static SlingInstanceRule instanceRule = new SlingInstanceRule();

    @Rule
    public SlingRule methodRule = new SlingRule(); // config timeout, description

    @Test
    public void testChangeOSGiConfig() {
        SlingClient client = instanceRule.getAdminClient();

        client.createNode("/content/myNode", "nt:unstructured");
        Assert.assertTrue("Node should be there", client.exists("/content/myNode"));
    }
}
```


HApi Client

Hypermedia Api

- **Hypermedia:** a nonlinear medium of information which includes graphics, audio, video, plain text and hyperlinks.
- **Microdata:** HTML specification used to nest metadata within existing content on web pages
- **HApi tools:** automatically add metadata information to sling HTML components

```
<div itemscope itemtype="http://schema.org/Movie">
  <h1 itemprop="name">Avatar</h1>
  <span itemprop="genre">Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html" itemprop="trailer">Trailer</a>

  <div itemprop="director" itemscope itemtype="http://schema.org/Person">
    Director: <span itemprop="name">James Cameron</span >
  </div>
</div>
```

■ Design

```
public class MicrodataClient extends SlingClient implements HtmlClient {...}
```

■ Usage

```
MicrodataClient client = slingClient.adaptTo(MicrodataClient.class);  
MicrodataDocument home = client.enter("index.html");  
assertEquals(home.items().length() == 5);  
MicrodataDocument details = home.link("details").follow();
```

Conclusions

- Version 1.0 already released
- Sample Sling tests available
- 2000+ tests (AEM):
 - 3-level hierarchy clients
 - testing multiple versions
- Please contribute!