Meecrowave Testing

JUnit

Coordinates:

```
<dependency>
  <groupId>org.apache.meecrowave</groupId>
  <artifactId>meecrowave-junit</artifactId>
  <version>${meecrowave.version}</version>
  </dependency>
```

Rules and Runners

Meecrowave provides two flavors of JUnit integration: standalone or runners/rules. The standalone one will ensure there is a single container for the whole JVM. It also fits standalone environments where you want to control the lifecycle. The other one will follow the JUnit lifecycle (per class or test rule).

Here how to use the standalone flavor:

```
@RunWith(MonoMeecrowave.Runner.class)
public class MonoMeecrowaveRuleTest {
    /* or
    @ClassRule
    public static final MonoMeecrowave.Rule RULE = new MonoMeecrowave.Rule();
    */

    @MonoMeecrowave.Runner.ConfigurationInject
    private Meecrowave.Builder config;

    @Test
    public void test() throws IOException {
        // use "http://localhost:" + config.getHttpPort()
    }
}
```

When using the standalone, <code>@MonoMeecrowave.Runner.ConfigurationInject</code> allows to still access the configuration and random HTTP port.

For the configuration, the standalone runner will use a global configuration shared by all tests. To load it it will use a standard ServiceLoader on type org.apache.meecrowave.Meecrowave\$ConfigurationCustomizer.

And here is the one bound to the JUnit lifecycle

```
public class MeecrowaveRuleTest {
    @ClassRule // started once for the class, @Rule would be per method
    public static final MeecrowaveRule RULE = new MeecrowaveRule();

@Test
    public void test() throws IOException {
        // use "http://localhost:" + RULE.getConfiguration().getHttpPort()
    }
}
```

As usual with JUnit rules, you can decide whereas the Meecrowave instance is bound to the entire test class or a method by using @ClassRule or @Rule.

JUnit 5

JUnit 5 integrates a new Extension system. It is not yet very well supported by IDEs but you can already use it with Gradle and Maven (see http://junit.org/junit5/docs/current/user-guide/#running-tests).

The usage has two annotations: <code>@MeecrowaveConfig</code> which remaps most of the configuration of Meecrowave and <code>@MonoMeecrowaveConfig</code> which is close to <code>MonoMeecrowave.Runner</code> in term of usage.

```
@MeecrowaveConfig /*(some config)*/
public class MeecrowaveConfigTest {
    @ConfigurationInject
    private Meecrowave.Builder config;

@Test
    public void run() throws MalformedURLException {
        final String base = "http://localhost:" + config.getHttpPort();
        // asserts
    }
}
```

Or

```
@MonoMeecrowaveConfig
public class MeecrowaveConfigTest {
    // ...
}
```



JUnit 5 integration provides an <code>QAfterFirstInjection</code> method and <code>QAfterLastTest</code> which can be used to setup/reset some environment using injections once for a set of test methods. The methods must not have any parameter.

Arquillian Container

Container dependency:

```
<dependency>
  <groupId>org.apache.meecrowave</groupId>
   <artifactId>meecrowave-arquillian</artifactId>
   <version>${meecrowave.version}</version>
  </dependency>
```

For the configuration check Core configuration.

Here is a sample:

```
<?xml version="1.0" encoding="UTF-8"?>
<arquillian xmlns="http://jboss.org/schema/arquillian"</pre>
                        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        xsi:schemaLocation="http://jboss.org/schema/arquillian
http://jboss.org/schema/arquillian/arquillian_1_0.xsd">
   <container qualifier="meecrowave" default="true">
        <configuration>
            cproperty name="arquillianProtocol">Servlet 3.1/property>
            property name="cdiConversation">false
            clientAuth">
            conf">
            cyroperty name="cxfServletParams">
                hide-service-list-page=true
            </property>
            cproperty name="defaultSSLHostConfigName">
            cproperty name="deleteBaseOnStartup">true/property>
            property name="dir">
            property name="host">localhost/property>
            property name="http2">false/property>
            property name="httpPort">-1
            property name="httpsPort">8443
            cproperty name="initializeClientBus">true
            cproperty name="jaxrsDefaultProviders"></property>
            property name="jaxrsLogProviders">false/property>
            property name="jaxrsMapping">/*
            cproperty name="jaxrsProviderSetup">true</property>
            cproperty name="jaxwsSupportIfAvailable">true/property>
            cycle content in the cycle content
            cyroperty name="jsonbEncoding">UTF-8
            cyproperty name="jsonblJson">false/property>
            property name="jsonbNamingStrategy">
            property name="jsonbNulls">false
            cyroperty name="jsonbOrderStrategy">
```

```
<property name="jsonbPrettify">false</property>
     cyroperty name="jsonpBufferStrategy">QUEUE
     cproperty name="jsonpMaxReadBufferLen">65536</property>
     property name="jsonpMaxStringLen">10485760/property>
     property name="jsonpMaxWriteBufferLen">65536/property>
     cproperty name="jsonpPrettify">false/property>
     cyroperty name="jsonpSupportsComment">false/property>
     coperty name="keyAlias">
     property name="keystoreFile"></property>
     property name="keystorePass"></property>
     cyroperty name="loggingGlobalSetup">true/property>
     <property name="loginConfig">authMethod=BASIC;realmName=app
     cproperty name="properties">
      jpa.property.openjpa.RuntimeUnenhancedClasses=supported
      jpa.property.openjpa.jdbc.SynchronizeMappings=buildSchema
     </property>
     cyroperty name="quickSession">true
     cproperty name="realm"
>org.apache.catalina.realm.JAASRealm:configFile=jaas.config;appName=app</property>
     property name="roles">
      admin=admin
      limited=admin,other
     </property>
     conningExcludes">
     cproperty name="scanningIncludes"></property>
     cproperty name="scanningPackageExcludes"></property>
     cproperty name="scanningPackageIncludes"></property>
     constraints"
>collection=sc1:/api/*:POST;authRole=**|collection=sc2:/priv/*:GET;authRole=*</propert
y>
     cproperty name="serverXml"></property>
     cproperty name="sharedLibraries"></property>
     cproperty name="skipHttp">false</property>
     cyroperty name="ssl">false/property>
     cproperty name="sslProtocol"></property>
     property name="stopPort">-1
     cproperty name="tempDir"
>/var/folders/cc/5g6j5x3x74154g2rmmtpz 4w0000gn/T/meecrowave 184680423404202</property
>
     cproperty name="tomcatAccessLogPattern">
     property name="tomcatAutoSetup">true
     property name="tomcatFilter">
     property name="tomcatNoJmx">true
     property name="tomcatScanning">true
     cproperty name="tomcatWrapLoader">true
     cproperty name="useLog4j2JulLogManager">true
     cproperty name="useShutdownHook">false/property>
     cyroperty name="useTomcatDefaults">true/property>
```