



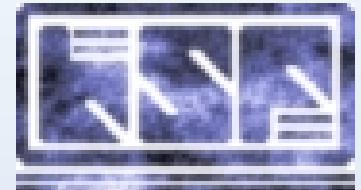
# CACTUS

The thorn in your bug's side

David Matějček

[matejcek@hsf.cz](mailto:matejcek@hsf.cz)

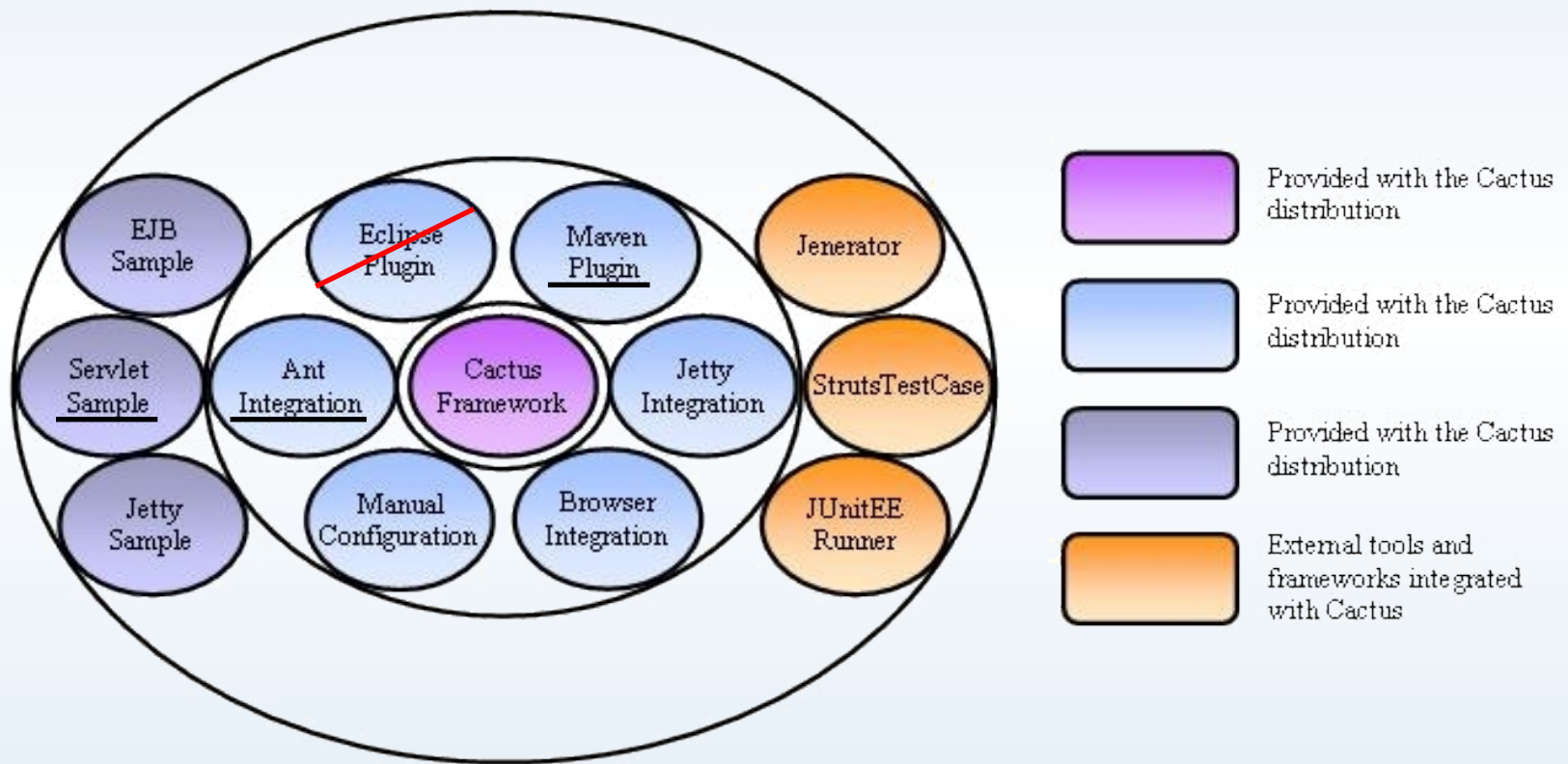
4. dubna 2006

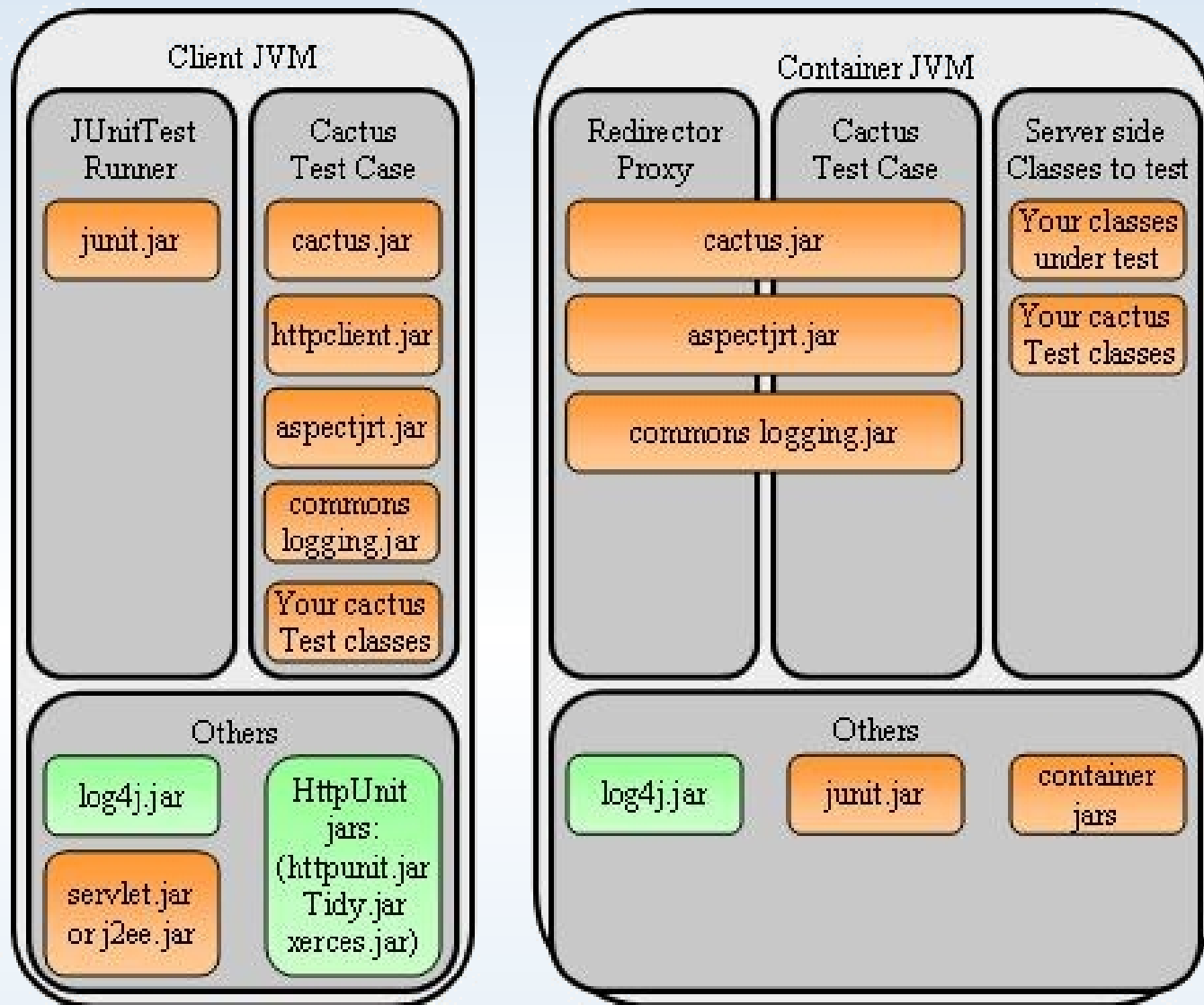


# Co je to?

- Jednoduchý testovací framework pro testování javovského kódu na straně serveru (Servlets, EJBs, Tag Libs, Filters, ...).
- Používá projekt JUnit a rozšiřuje ho.
- Integrační testy, interakce s kontejnerem
- Možnost spouštět testy proti více variantám serverů

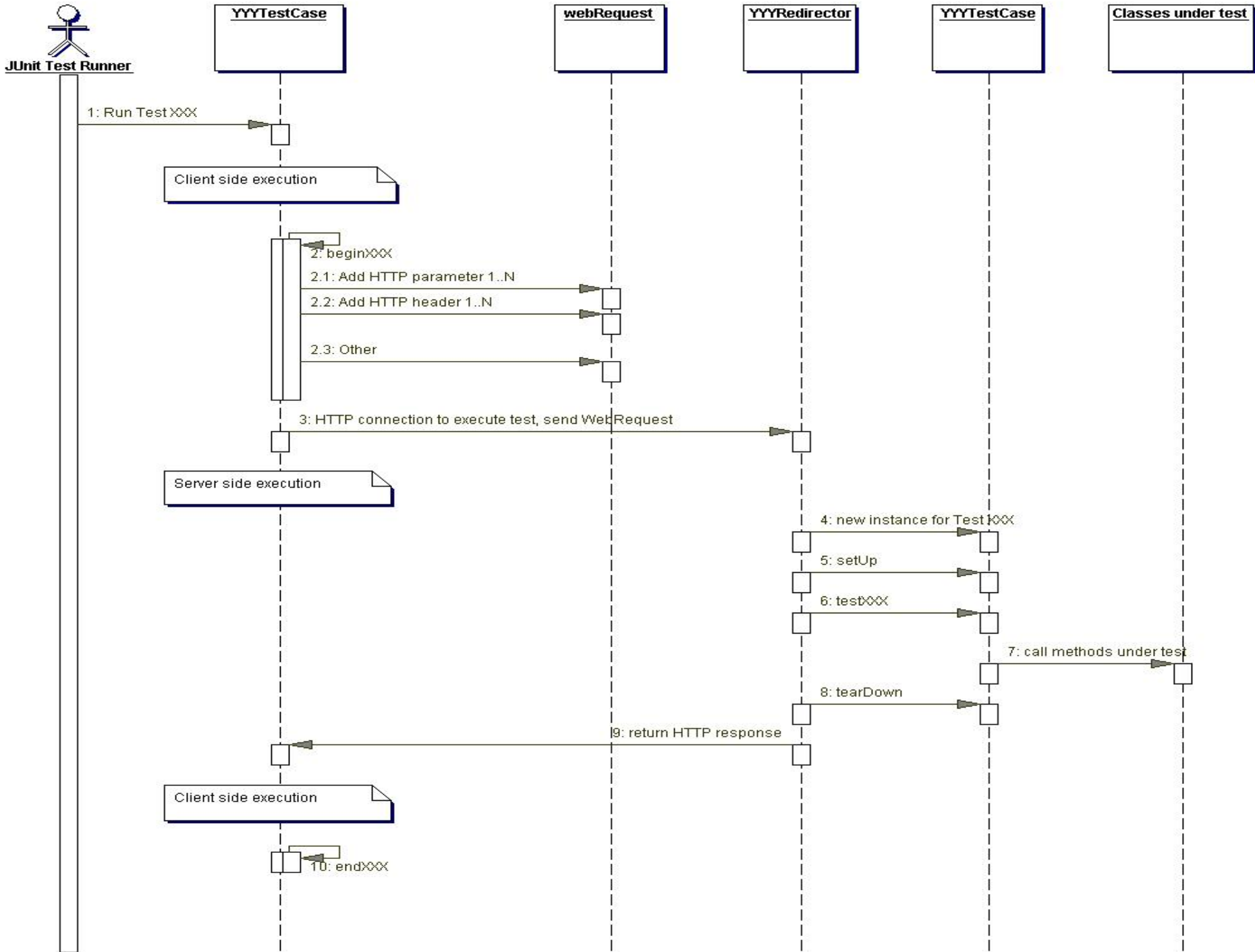
# „Ekosystém“





 Mandatory  
 Optional

**Figure 1 : Classpath configuration**



# Příklady

## 1) První pokus

[http://jakarta.apache.org/cactus/integration/howto\\_tomcat.html](http://jakarta.apache.org/cactus/integration/howto_tomcat.html)

~~[http://jakarta.apache.org/cactus/integration/integration\\_browser.html](http://jakarta.apache.org/cactus/integration/integration_browser.html)~~

## 2) Užitečný příklad pro Ant a Maven

- Ant – složitější, funkčnost téměř stejná
- Maven – snadno, rychle a stejně dobře

# První příklad – první problémy

- Tomcat; popis na stránce:

[http://jakarta.apache.org/cactus/integration/howto\\_tomcat.html](http://jakarta.apache.org/cactus/integration/howto_tomcat.html)

+ osvětluje princip

– doporučené XSLT - IE ok, ostatní mají problémy

– „rozbagrování“ konfigurace Tomcatu, ztráta času

– prakticky nepoužitelný

```
javac -classpath .\servlet-api.jar SampleServlet.java
```

```
javac -classpath .\servlet-api.jar;.\junit.jar;.\cactus-1.7.1.jar; .\ TestSampleServlet.java
```

## 1) „testovaná“

[tomcat-root]/webapps/test/WEB-INF/classes/SampleServlet.class

```
public class SampleServlet extends HttpServlet {  
  
    public void saveToSession(HttpServletRequest request) {  
  
        String testparam = request.getParameter("testparam");  
        request.getSession().setAttribute("testAttribute", testparam);  
  
    }  
}
```

---

Hodnota parametru `testparam`, získaná z požadavku, získaného od klienta, se uloží do relace klienta.



## 2) „testující“

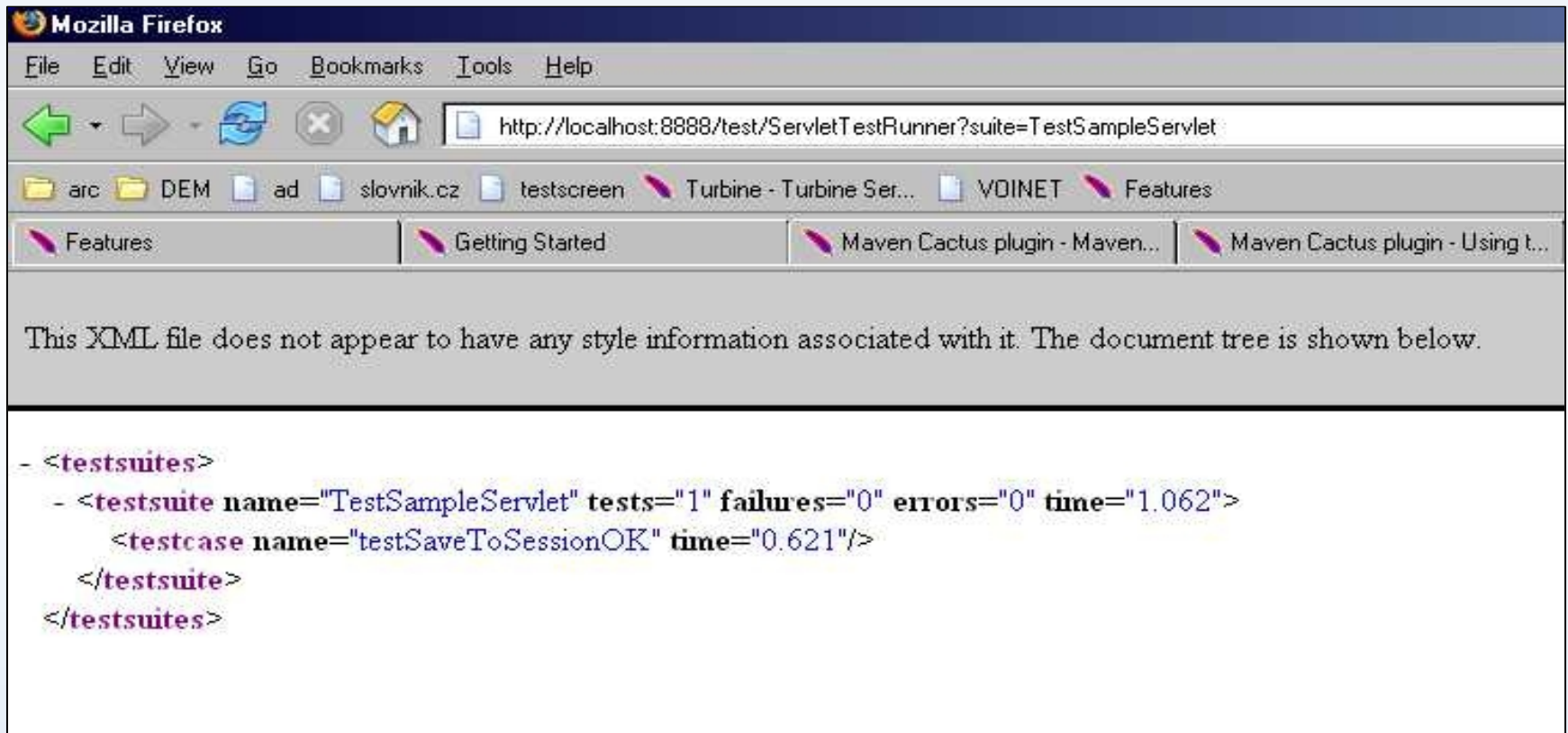
[tomcat-root]/webapps/test/WEB-INF/classes/TestSampleServletJava.class

```
public class TestSampleServlet extends ServletTestCase {
    public TestSampleServlet(String theName) {
        super(theName);
    }
    public static Test suite() {
        return new TestSuite(TestSampleServlet.class);
    }

    // klient - odešle požadavek s parametrem
    public void beginSaveToSessionOK(WebRequest webRequest) {
        webRequest.AddParameter("testparam", "it works!");
    }

    // server - přijme požadavek, vyvolá testovanou metodu a zkontroluje jí
    public void testSaveToSessionOK() {
        SampleServlet servlet = new SampleServlet();
        servlet.saveToSession(request);
        assertEquals("it works!", session.getAttribute("testAttribute"));
    }
}
```

# Výstup bez XSLT



The screenshot shows a Mozilla Firefox browser window. The address bar contains the URL `http://localhost:8888/test/ServletTestRunner?suite=TestSampleServlet`. The browser's menu bar includes File, Edit, View, Go, Bookmarks, Tools, and Help. The toolbar shows navigation buttons and a search icon. The bookmarks bar lists several folders and files, including 'arc', 'DEM', 'ad', 'slovník.cz', 'testscreen', 'Turbine - Turbine Ser...', 'VOINET', and 'Features'. The browser's status bar at the bottom displays the message: "This XML file does not appear to have any style information associated with it. The document tree is shown below." Below this message, the XML document tree is displayed in a preformatted text view:

```
- <testsuites>
  - <testsuite name="TestSampleServlet" tests="1" failures="0" errors="0" time="1.062">
    <testcase name="testSaveToSessionOK" time="0.621"/>
  </testsuite>
</testsuites>
```

# Metody I

- `begin(WebRequest request)`
  - klient, veškerá inicializace klienta, objektů atp. Znovu se spouští před každým testem, není povinná.
- `beginGetMainPage(WebRequest request)`
  - klient, nastavení požadavku. Cactus vzápětí požadavek odešle.
- `public void setUp()`
  - server, inicializace serveru. Znovu se spouští před každým testem, lze provádět první kontroly. Implementována Cactusem, vhodné překrýt.
- `public void test...()`
  - server, hlavní testy. Cactus opakuje sekvenci metod pro každou metodu, začínající test....

# Metody II

- `public void tearDown()`
  - server, vypnutí serveru. Znovu se spouští po každém testu. Implementována Cactusem, vhodné překrýt.
- `public void endGetMainPage(WebResponse resp)`
  - klient, kontrola obdržené odpovědi.
- `public void end()`
  - klient, ukončování klienta, není povinná.

# Užitečný příklad

- Upravený příklad, který je ve zdrojových kódech Cactusu
- Příklady jsou ve skutečnosti tři – Filter, JSP a Servlet; upravený je třetí jmenovaný, zbývajících se prezentace nezabývá
- Plnohodnotný příklad pro Maven obsahuje nastavení logování klienta i serveru, také všechny použitelné metody (byť některé prázdné)
- V prezentaci je jen krátký náznak, vyzkoušejte si ho sami ;-)

## 1) „testovaná“

### SampleServlet.java

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

// Vraci true, pokud atribut authenticated ze session je true
public class SampleServlet extends HttpServlet {

    private static final long serialVersionUID = -986177106059251875L;

    public boolean isAuthenticated(HttpServletRequest theRequest) {
        HttpSession session = theRequest.getSession(false);

        if (session == null) {
            return false;
        }

        String authenticationAttribute =
            (String) session.getAttribute("authenticated");

        return Boolean.valueOf(authenticationAttribute).booleanValue();
    }
}
```

## 2) „testující“

### TestSampleServlet.java

```
public class TestSampleServlet extends ServletTestCase {

    // lze hodnotu zaznamenat do session?
    public void testIsAuthenticatedAuthenticated() {
        SampleServlet servlet = new SampleServlet();
        session.setAttribute("authenticated", "true");

        assertTrue(servlet.isAuthenticated(request));
    }

    // klient - vypnutí session
    public void beginIsAuthenticatedNoSession(WebRequest theRequest) {

        theRequest.setAutomaticSession(false);
    }

    // server - viz min. strana - žádná session => servlet vrátí false
    public void testIsAuthenticatedNoSession() {
        SampleServlet servlet = new SampleServlet();

        assertFalse(servlet.isAuthenticated(request));
    }
}
```

# Cactus: build.properties

# port, kde má Cactus spouštět testy

cactus.port = 8888

# seznam servlet-kontejnerů, na kterých testovat

cactus.home.tomcat4x = C:/home/dmatej/Tomcat41

cactus.home.tomcat5x = C:/home/jindra/sources/tomcat

# soubory s nastavením logování klienta a serveru

cactus.logging.config.client = \${basedir}/conf/cactus.client.properties

cactus.logging.config.server = \${basedir}/conf/cactus.server.properties

---

Další možná (nepovinná) nastavení viz:

<http://jakarta.apache.org/cactus/integration/maven/properties.html>



# Výsledky testů

cactus:test-war:

```
[cactus] -----
```

```
[cactus] Running tests against Tomcat 5.0.28 @ http://localhost:8888
```

```
[cactus] -----
```

```
[cactus] Deleting 719 files from C:\DOCUME~1\matejcek\LOCALS~1\Temp\cactus\tomcat5x
```

```
[cactus] Deleted 47 directories from
```

```
C:\DOCUME~1\matejcek\LOCALS~1\Temp\cactus\tomcat5x
```

```
... ..
```

```
[cactus] Running org.apache.maven.cactus.sample.TestSampleServlet
```

```
[cactus] Tests run: 2, Failures: 0, Errors: 0, Time elapsed: 0,79 sec
```

```
[cactus] Testsuite: org.apache.maven.cactus.sample.TestSampleServlet
```

```
[cactus] Tests run: 2, Failures: 0, Errors: 0, Time elapsed: 0,79 sec
```

```
[cactus]
```

```
[cactus] Testcase: testIsAuthenticatedAuthenticated took 0,7 sec
```

```
[cactus] Testcase: testIsAuthenticatedNoSession took 0,03 sec
```

```
... ..
```

# Pros And Cons

- + Možnost testování klient – server aplikací
- + Umí zapouzdřit i testy JUnit
- + Testy probíhají na skutečném balíku – Cactus analyzuje WAR a poté si sestaví svůj.
- + Neovlivňuje kód stávajícího projektu
- Z opakování sestavení vyplývá celkem velká časová náročnost (analýza + opětovné sestavení + spouštění kontejneru)

---

Závěr: pro testování během vývoje používat Mock objekty a JUnit, Cactus je vhodný na integrační testy (interakce s okolím a komponent navzájem).

# Odkazy

- Cactus: <http://jakarta.apache.org/cactus>
- Maven: <http://maven.apache.org>
- Ant: <http://ant.apache.org>
  
- Jetty: <http://jetty.mortbay.org/jetty>  
- na světě není jen Tomcat ;-)
- Jenerator: <http://www.visioncodified.com>  
- XSLT transformace XML na kód

**Děkuji za pozornost**