



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék
Hibatűrő Rendszerek Kutatócsoport

KRITIKUS ARCHITEKTÚRÁK LABORATÓRIUM
2015/2016 TAVASZI FÉLÉV

Automatikus tesztfuttatás

Mérési feladatok
(v1.3)

Készítette: Oláh János, Honfi Dávid
(janos.olah,honfi@mit.bme.hu)

2016. május 5.

A MÉRÉS SORÁN ELVÉGZENDŐ FELADATOK

Kiindulás

A virtuális gépen telepítve van minden szoftver, ám némelyhez előzetes konfiguráció szükséges. Minden újraindítás után a következő parancsokat kell kiadni a dotCMS elindításához.

```
sudo service tomcat8 stop
sudo bash ~/dotcms/bin/startup.sh
```

1. feladat: Egyszerű Selenium szkript rögzítése

Miután elindította a virtuális gépet (*meres/LaborImage*), győződjön meg arról, hogy fut a dotCMS tartalomkezelő. (<http://localhost:8081/admin>.) Lépjen be az admin felületen (*admin@dotcms.com/admin*), röviden vizsgálja át az adminisztrációs lehetőségeket. A felső sorban egy külön lapon található az IRC kliens, melyet a mérésen tesztelni fogunk.

Amennyiben a belépési képernyőn "500 Internal Server Error" üzenet jelenik meg, akkor a következő parancsokat kell kiadni.

```
sudo bash ~/dotcms/bin/shutdown.sh
rm -r -d ~/dotcms/dotserver/tomcat-8.0.18/webapps/ROOT/dotsecure
sudo bash ~/dotcms/bin/startup.sh
```

Miután végiggondolta, hogy hogyan tudna ehhez olyan teszteseteket készíteni, melyekkel le lehet fedni a teljes funkcionalitást, indítsa el a Selenium IDE-t az Eszközök menüből.

Az IDE segítségével hozzon létre néhány (3-4) tesztesetet, és próbálja őket lefuttatni. A tesztesetek létrehozásánál érdemes arra figyelni, hogy azok a Firefox elindításával indulnak a későbbiekben (így pl. minden teszteset induljon azzal, hogy elnavigálunk a dotCMS adminfelületére, beírjuk a felhasználónevet/jelszót, stb.) Próbáljon ki több változatot is, ismerkedjen meg az IDE-vel és a konfigurációs lehetőségeivel.

Az elkészült teszteseteket és a tapasztalatait dokumentálja a jegyzőkönyvbe!

2. feladat: Szkript exportálása és futtatása JUnit tesztként

A Selenium IDE ugyan nagyszerű eszköz, de nem éppen a legcélszerűbb komolyabb tesztelési projektek esetén. Ezért szeretnénk a teszteseteket könnyebben menedzselhetővé illetve végrehajthatóvá tenni.

Vizsgálja meg az exportálási lehetőségeket, majd exportálja az előző feladatban elkészült teszteseteket JUnit Java fájlként. Mivel mi a következőkben a Selenium WebDriver-t fogjuk használni, válassza ezt a lehetőséget.

Hozzunk létre tehát egy új Maven projektet az Eclipsen belül. A varázsló első két oldalán fogadjuk el az alapértelmezett értékeket (de azért megvizsgálhatjuk, milyen sokféle előredefiniált projektfelépítés közül választhatnánk), a harmadik oldalon pedig adjunk nevet a projektünknek. Ezzel máris létrejött a projekt.

Vizsgáljuk meg az új Maven projekt felépítését, és nézzük meg a pom.xml-t is. Látható, hogy bekerült függőségként a JUnit, de például a Selenium nem. Ráadásul a JUnit-ból is van már frissebb verzió, így az XML fájl függőségek részét írjuk felül az alábbi kódrésszel:

```
<dependencies>
<dependency>
<groupId>org.seleniumhq.selenium</groupId>
<artifactId>selenium-java</artifactId>
<version>2.53.0</version>
</dependency>
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.10</version>
<scope>test</scope>
</dependency>
</dependencies>
```

Futtassa le a teszteseteket néhány alkalommal, próbálja módosítani a forrásfájlokat a WebDriver API segítségével. Például vezesse be változók használatát, próbáljon ki egyéb elérési utakat is (objektum név, XPath kifejezések, stb.). (Itt figyeljünk arra, hogy a JUnit által ajánlott nevezési konvenciót a Maven ki is kényszeríti, vagyis csak akkor fog lefutni egy teszteset, ha **„Test” szóra végződik az osztály neve!**)

A Maven futtatásához pedig válasszuk a „Run as/Maven install” parancsot. Ez több célt is magába foglal, így a tesztelés is le fog futni.

Az elkészült teszteseteket illetve a Maven projekt készítése során szerzett tapasztalatait, valamint a futtatás eredményét dokumentálja a jegyzőkönyvbe!

3. feladat: Folytonos integrációt biztosító szerver konfigurálása

A teljes automatizálástól így már csak egy lépés választ el minket, a folytonos integrációt támogató szerver konfigurációja. A böngészőben navigáljon át a Jenkins webes adminisztrációs felületére (<http://localhost:8082>), és ismerkedjen meg a képernyőjével.

Első lépésben a Jenkins számára meg kell adni a Maven elérési útját. Ezt a <http://localhost:8082/configure> oldalon tehetjük meg, ahol a Maven szekcióban ki kell kapcsolni az "Install automatically" kapcsolót, majd a MAVEN_HOME mezőbe a "/usr/share/maven" könyvtárat megadni, névnek pedig pl. azt, hogy "Maven".

A feladat ezután az, hogy létrehozzon egy új Jenkins jobot, mely az előző feladatban elkészült tesztek lefuttatását fogja végrehajtani. Ehhez először ossza meg a korábban létrehozott Maven projektet a tárgy keretében használt saját GitHub repositoryjukon keresztül, melyet a Jenkinsnek a repository URL hivatkozásával lehet majd átadni. Ezen felül be kell kapcsolni a GitHub projekt kapcsolót is, és megadni a repository elérési útját.

Hozza létre az új jobot. Konfigurálja úgy, hogy a build akkor fut le, ha a GitHubra pushsal felkerült egy változás a repositoryba.

Az feladat során szerzett tapasztalatait, valamint a Jenkins job lefutásának eredményét dokumentálja a jegyzőkönyvbe!