# Telerik QA Academy

# Sample Intermediate Exam #3

### Performance and Load Testing with Visual Studio

Go to <http://forums.academy.telerik.com/>.

Your task is to perform the following tests:

A) Create a performance test for testing the application that should include loading three separate pages of the site.

B) Create a load test (based on the performance test) with 20 users (call the test scenario “20UsersTest”). The test should start with 1 user and the count of users should grow with 1 user in every 5 seconds until it reaches 20 users. The test duration should be 2 minutes.

C) Perform two separate runs of the load test and generate a report in MS Excel that shows the differences in collected metrics between the two runs. The report should contain the following metrics: average page time, average test time, average response time, requests/sec., user load and total transactions.

D) Create a data-driven performance test that performs login and logout of the forum with two different users. Create a load test with constant load of 10 users and duration of 2 minutes. Make one run of this test and create a report for it. Use the same metrics as for sub problem C).

For the purpose of the last task:

username: qa-academy-1 / pass: qa-academy

username: qa-academy-2 / pass: qa-academy

As a result of this exam problem you should submit your Visual Studio project and Excel reports.

Make the task with Test Studio. Consider which metrics to use.

### Performance and Load Testing with Visual Studio

You are given the source code of a small Web application (**CarService-Example.rar**). Your task is to deploy the application in IIS on your local machine and perform the following tests:

A) Create performance tests for testing the application.

1. Create one performance test for ONE of the pages;

2. Create second performance tests with THREE pages in it.

B) Create load tests based on the performance tests with the following parameters:

1. Crete load test with 5 users for both performance tests.

2. Create a load test with 100 users (again for both performance tests). The test should start with 10 users and the count of users should grow with 5 users in every 5 seconds until it reaches 100 users.

As a result of this exam problem you should submit your project, created with Visual Studio.

### Testing for Cross Site Scripting (XSS) Vulnerabilities

You are given the source code of a small Web application (**BugTrackingSystem.rar**). Your task is to run the application on your local machine and perform the following security testing:

1. Find which part of the site is vulnerable to Cross Site Scripting (XSS). Note that the site might accept an XSS attack through a particular element, but the result of the attack might be manifested in another part of the site so search carefully.
2. Once you have found the XSS vulnerability, exploit it by inserting a script that causes a hyperlink "Hack" to appear somewhere on the page pointing to [www.blog.com](http://www.blog.com).

Notes: you may need to attach the database in SQL Server and use Visual Studio to compile, inspect and run the Web application.

Submit your solution to this problem as screenshots where the vulnerable elements and the results of your XSS attack/s should be pointed. Use labels in the screenshots or a separate text file to submit the exact attack code you have used.

### Testing for SQL Injection Vulnerabilities

Use the same Web application given as source code for the previous problem but this time look for a vulnerability to SQL Injection. The problem has three parts:

1. Find the vulnerable element in the application.
2. Exploit the vulnerability to delete all bugs from the database.

Submit your solution of this problem as screenshots where the vulnerable elements and the results of your SQL Injection attack/s should be pointed. Use labels in the screenshots or a separate text file to submit the exact input used for the SQL Injection attack.

### Testing for URL Manipulation Vulnerabilities

Use the same Web application given as source code for the previous problem but this time look for a vulnerability to URL manipulation. The problem has two parts:

1. Find the vulnerable element in the application.
2. Exploit the vulnerability to modify private information which is not intended to be accessible by the users without administrative privileges.

Submit your solution of this problem as screenshots where the vulnerable elements and the results of your URL Manipulation attack/s should be pointed. Use labels in the screenshots or a separate text file to submit the exact input used for the URL Manipulation attack.

### WebService Testing

You are given the sample web service application: <http://webservices.daehosting.com/services/TemperatureConversions.wso?WSDL>

Your task is to use SoapUI and create tests for testing the web service:

1. Create requests for testing the methods **CelciusToFahrenheit** and **FahrenheitToCelcius**. Create appropriate valid and invalid cases.
2. Create a test suite with the tests from **CelciusToFahrenheit** and **FahrenheitToCelcius** and add appropriate assertions to them. (For content assertions – make sure your assertions validate the specific element wanted – not just presence of the value in the response in general.)
3. Make a property transfer (in the created test suite) of the field ***CelciusToFahrenheitResult*** from the response of a valid request for method **CelciusToFahrenheit** and use it in the field ***nFahrenheit*** for a valid request for the method **FahrenheitToCelcius.**
4. Create a load test for one of the methods with 4 users and duration of 90 seconds.

As a result of this exam problem you should submit your project created with SoapUI.