

QSYS

# Administrator Manual

Sabine Grabner

Version 1.0 beta

Last Change: June 29th 2009

## **Abstract**

The administrator manual should assist in getting an instance of QSYS up and running. Furthermore it gives insight into QSYS' data model and access control system. The document also explains how one can perform stress tests with jMeter and summarizes some tests that were made. QSYS' model supports various features that have not been implemented in QSYS' view yet. A brief introduction of these features and configuration instructions are given below. For further information it is advised to study the user and developer manual.

## Table of Contents

<b>TECHNOLOGY</b>	<b>5</b>
<b>DEPLOYMENT</b>	<b>5</b>
DOWNLOAD	5
SERVLET ENGINE	5
<b>CONFIGURATION</b>	<b>5</b>
TOMCAT CONFIGURATION	5
ENVIRONMENT SETTINGS	6
APPLICATION CONFIGURATION	6
<b>SECURITY</b>	<b>6</b>
ADMINISTRATOR AUTHENTICATION	6
AUTHORING AUTHENTICATION	6
QSYS START PAGE	7
SECURITY PROPERTIES	7
<b>APPLICATION PROPERTIES</b>	<b>9</b>
<b>STORAGE</b>	<b>9</b>
RESOURCES	9
DATA MODEL	10
STORAGE OPTIONS	10
FILE SYSTEM	11
ORACLE DATABASE	12
EXIST DATABASE	12
<b>ADMINISTRATOR TOOL</b>	<b>12</b>
<b>AUTHORIZATION</b>	<b>12</b>
ADMINISTRATOR POLICY	12
<b>NONE-GUI FEATURES</b>	<b>12</b>
INTERNATIONALIZATION	13
GRADING	13
ANALYSIS	13
<b>MISCELLANEOUS</b>	<b>14</b>
USING GOOGLE ANALYTICS	14
<b>PERFORMANCE TESTING</b>	<b>15</b>
TOOL	15
TEST DESCRIPTIONS AND RESULTS	15



## Technology

QSYS was implemented using Java SDK 1.5 and was extensively tested on Tomcat 5.5.26. The web interface reliably works on most recent versions of Safari and Firefox, while tests on Internet Explorer have not been as exhaustive. More details on the technical background of QSYS can be found in the developer manual.

## Deployment

### Download

Download qsys.war from <http://sourceforge.net/projects/qsys/>. If you prefer to build qsys from source follow the instructions of the QSYS developer manual.

### Servlet Engine

QSYS was extensively tested on Tomcat 5.5.26 which is recommended to use. The following deployment steps are described for Tomcat 5.5.26.

- Put qsys.war into TOMCAT\_HOME/webapps
- In order to compile and run translets QSYS uses a JAXP 1.3 compliant XML parser, namely Xerces 2.9. To make those classes accessible to QSYS considering Tomcat's classloader precedence put the following jars into TOMCAT\_HOME/common/endorsed: xalan.jar, xercesImpl.jar, serializer.jar

The jars can be extracted from qsys.war by running the following commands:  
jar xf qsys.war WEB-INF/lib/serializer.jar  
jar xf qsys.war WEB-INF/lib/xalan.jar  
jar xf qsys.war WEB-INF/lib/xercesImpl.jar

More information on this subject can be found here:

[http://xml.apache.org/xalan-j/xsltc\\_usage.html](http://xml.apache.org/xalan-j/xsltc_usage.html)

<http://tomcat.apache.org/tomcat-5.5-doc/class-loader-howto.html> under 'XML Parsers and J2SE 1.4'

- Start Tomcat: TOMCAT\_HOME/bin/startup.sh
- Shut down Tomcat for configuration as described in the next section: TOMCAT\_HOME/bin/shutdown.sh
- Start Tomcat again
- Access QSYS at: <http://yourDomain/qsys>

## Configuration

### Tomcat Configuration

TOMCAT\_HOME/conf/server.xml:

Connector element: add the attribute **URIEncoding="UTF-8"** to read form fields with UTF-8 encoding.

## Environment Settings

TOMCAT\_HOME/bin/env.sh:

```
export CATALINA_OPTS="-server -Xms512M -Xmx512M -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Dencoding=UTF-8"
```

## Application Configuration

Below is a list of files that need be configured. Details on the properties are found in the respective sections further below.

- *TOMCAT\_HOME/webapps/qsys/WEB-INF/web.xml*: the deployment descriptor file needs to be adjusted according to your choice of authentication mechanism (leave default settings if deciding for QSYS' native authentication, otherwise follow instructions of the security section)
- *TOMCAT\_HOME/webapps/qsys/WEB-INF/classes/security.properties*: properties related to user authentication and authorization (leave default settings if deciding for QSYS' native authentication, otherwise follow instructions of the security section)
- *TOMCAT\_HOME/webapps/qsys/WEB-INF/classes/qsys.properties*: QSYS application properties

Storage related properties:

- *TOMCAT\_HOME/webapps/qsys/WEB-INF/classes/fsdb.properties*: properties related to data storage on the file system (recommended)
- *TOMCAT\_HOME/webapps/qsys/WEB-INF/classes/exist.properties*: properties related to data storage in an eXist database (only if used)
- *TOMCAT\_HOME/webapps/qsys/WEB-INF/classes/oracle.properties*: properties related to Oracle data storage (only if used)

**Updating configuration files requires a restart of tomcat!**

## Security

QSYS distinguishes three kinds of users, namely the administrator, authors, and interviewees. For authentication of authors and interviewees several mechanisms are supported and the list can be extended with little effort. For more information on adding a new mechanism see the developer manual. Supported mechanisms include QSYS' native authentication mechanism, LDAP, and Shibboleth.

### Administrator Authentication

The URL to the administrator login page is adminlogin.qsys and the password is set in qsys.properties. The administrator tool is described further down.

### Authoring Authentication

QSYS' native authentication system requires the admin to create authoring accounts upon requests. Therefore using any of the external authentication mechanisms

decreases administration tasks significantly. The URL for authoring login is `authoringlogin.qsys`.

## QSYS Start Page

QSYS' default start page is `index.html` that only shows a redirect-information. To omit this page on QSYS-access filter it out. `intex.html` redirects the user to `startpage.qsys`. `startpage.qsys` offers two links, one for the administrator login and one for the authoring login. If you would like to save the majority of your users opting for the authoring login page, filter out `startpage.qsys` requiring the administrator to visit `adminlogin.qsys` explicitly.

### web.xml

```
<filter-mapping>
  <filter-name>AuthenticationFilter</filter-name>
  <url-pattern>/index.html</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>AuthenticationFilter</filter-name>
  <url-pattern>/startpage.qsys</url-pattern>
</filter-mapping>
```

## Security Properties

Security properties are configured in `security.properties`. The `AuthenticationFilter` is responsible for initializing those settings upon server startup and therefore must not be commented out in `web.xml`. For invoking changes to the `security.properties` file the server must be restarted.

### web.xml

```
<filter>
  <filter-name>AuthenticationFilter</filter-name>
  <filter-class>at.fhv.security.authentication.AuthenticationFilter</filter-class>
</filter>
```

### security.properties

#### *Common Properties*

For using QSYS' native authentication mechanism copy the exact content of the following box, otherwise copy the content and adjust `LOGIN_MECHANISM` and `SINGLE_SIGN_ON` accordingly.

```
#####
##          SET UP YOUR AUTHENTICATION MECHANISM          ##
#####

# Chose between your application's native security mechanism or an external mechanism
# such as LDAP or SHIBBOLETH. You may add security mechanisms that suit your procedures.
# For off-line development you may want to use the DeveloperAuthenticaton Servlet.

# allowed values: application, shibboleth, ldap
LOGIN_MECHANISM=application

# SINGLE_SIGN_ON
# true: if you use a SSO authentication mechanism such as Shibboleth
#       and want your user to get a notification upon logout
#       that it affected the application session only but not the IDP session
#       set this property to true
# false: warning will be suppressed
SINGLE_SIGN_ON=false

# Where do you want the user to be redirected to after successful authentication?
POST_AUTHENTICATION_FILTER_URL_TO_GO=/PostAuthoringAuthentication
#####
```

```
## APPLICATION's NATIVE AUTHENTICATION
#####

# don't use the full URL, only the part after the context path
APPLICATION_AUTHENTICATION_URL=startpage.qsys
```

### *LDAP Specific Properties*

```
#####
## LDAP AUTHENTICATION
#####

# specify the UI where the user enters his/her credentials
# from there call the LDAPAuthentication Servlet and hand over the credentials
LDAP_AUTHENTICATION_URL=authoringloginv.qsys

### properties needed to connect to LDAP
LDAP_VERSION=3
LDAP_authenticate=true
LDAP_security-authentication=simple
LDAP_HOST=ldap.domain.at
LDAP_PORT=389

### properties needed to authenticate a user
LDAP_BIND_FILTER=uid={0},ou=People,dc=domain,dc=at

### what attributes would you like from LDAP?

## attributes to map username, roles, and groups to the User object's user, roles, groups
## if this level of information is not required, groups and roles attributes might as well
## be stored in LDAP_ATTRIBUTES

#which ldap attribute do you use as username in your application. This must not appear twice here!
LDAP_USERID=uid
# which ldap attribute do you use as groups for your application
LDAP_GROUPS=uid
# which ldap attribute do you use as roles for your application
#LDAP_ROLES=

## attributes that should also be used for policy evaluation
LDAP_ATTRIBUTES=mail, cn
```

### *Shibboleth Specific Properties*

```
#####
## SHIBBOLETH AUTHENTICATION
#####

#SHIB_USERID=cn
#SHIB_ROLES=affiliation
SHIB_GROUPS=unscoped-affiliation
#SHIB_IDP_ID=Shib-Identity-Provider
```

### *Attribute Mapping for external authentication mechanisms*

The authentication layer allows mapping between attribute names at their source (LDAP, Shibboleth) and in QSYS' access policies.

```
####
# ATTRIBUTES OBTAINED FROM THE AUTHENTICATION MECHANISM
# MAY NOT REFLECT THE ATTRIBUTE NAME USED IN POLICIES.
# TO MAP FROM THE OBTAINED ATTRIBUTE NAME TO THE
# ATTRIBUTE NAME USED IN POLICIES MAP THEM HERE:
# ON THE LEFT HAND SIDE USE YOUR ATTRIBUTE NAME FROM ABOVE
# IN UPPER CASE LETTERS CONCATENATED TO AM_. ON THE RIGHT
# HAND SIDE SPECIFY THE ATTRIBUTE NAME USED IN YOUR POLICIES.
# e.g. AM_UID = uniqueID
# ONE ATTRIBUTE FROM THE AUTHENTICATION MECHANISM MAY BE
# USED TO MATCH MULTIPLE ATTRIBUTES IN YOUR POLICY
# e.g. AM_NAME = CN, NICKNAME
####

AM_UID=urn:oasis:names:tc:xacml:1.0:subject:subject-id
AM_MAIL=email
```

Depending on your organization's network requirements, establish connections between QSYS' hosting server and the authentication server (e.g. VPN).

# Application Properties

Create the file and configure properties accordingly. Make sure both the policy directories and the data export directory exist on your file system.

```
qsys.properties

# *** ADMINISTRATOR ***
ADMIN_PW=qsys
# EMAIL ADDRESS OF ADMINISTRATOR
MAIL_ADDRESS=user@domain.at
# IF DEBUG IS TRUE, THE ADMINISTRATOR GETS TO SEE METADATA ON EACH PAGE ON THE WEB
DEBUG=false

# *** STORAGE SYSTEM ***
# pick between FILESYSTEM (recommended), EXIST, ORACLE
DB=FILESYSTEM

# *** POLICY ***
# QSYS distinguishes between resource policies and application policies
# While a resource policy rules over a certain resource
# application policies rule globally.
RESOURCE_POLICY_DIR=/Users/admin/Documents/qsys/data/policies/resourcepolicies
APPLICATION_POLICY_DIR=/Users/admin/Documents/qsys/data/policies/applicationpolicies

# *** EMAIL ***
# consider your organizations email policies regarding email relaying
# your applications server may need a VPN connection, or its IP may need be registered
# with the SMTP server to allow relaying
MAIL_ADDRESS_OUTGOING=no-reply@domain.at
#MAIL_PASSWORD_OUTGOING=
SMTP_SERVER=smtp.domain.at
NOTIFY_ADMIN=true

# *** DATA EXPORT ***
DATA_EXPORT_DIR=/Users/admin/Documents/qsys/data/export
```

## Storage

To understand the storage system one has to understand QSYS' distinction of the two kinds of resources first, namely questionnaires and templates, and the data model behind the concept.

## Resources

Templates are questionnaires with the limitation that they cannot be filled out and their instances cannot be modified.

So far QSYS supports templates only when using the file system for data storage. Another limitation is, that there is no GUI to create templates yet. Instead, an author creates a regular questionnaire and the administrator turns it into a template by following these steps:

1. move the resource folder from qdata/questionnaires/resoruceID to data/templates/resourceID
2. remove parts that are not relevant for a template (answers folder!)
3. update the policy to reflect a template policy
4. remove entry from data/questionnaires/questionnaires.xml and add it to data/templates/questionnaires.xml

The following table highlights the major differences between questionnaires and templates:

	Participate	Edit	Other operations	
Template	Never	As long as no instance has been created.		
Template instance	Yes	Never	Template owner should specify who can instantiate and what instantiator can do.	
Questionnaire	yes	As long as no answers have been given.		

## Data Model

A resource whether it being a template or questionnaire consists of general components. Additionally, specific components apply to templates and questionnaires respectively.

1. *questionnaire* – holds questions of the questionnaire. It can exist in various languages.
2. *questexplanation* – holds welcome and good bye text for the interviewee. It can exist in various languages.
3. *basicsettings* – holds general settings about the appearance
4. *answers* – folder containing all answer documents

The following components are added if the related modules are being used:

5. *analysis* – holds analysis configuration
6. *grading* – holds grading configuration such as the correct answers and scores to be obtained, and grade ranges
7. *notification* – holds notification configuration such as email addresses of people to be notified upon participations

A questionnaire has the following additional components:

8. *interviewees* – holds interviewee mode

A template has the following additional components:

9. *instances* – a list of all questionnaire IDs that are instances of the template

## Storage options

QSYS' data are stored using the XML data format, where the XML content can be stored in either of the following systems:

1. File system
2. Oracle database
3. eXist database

## File System

Create the file and configure the PATH property accordingly.

*fsdb.properties*

PATH=/Users/admin/Documents/qsys/data

The file hierarchy of the resources is as follows:

PATH/

- archive/ (deleted questionnaires are moved here)
- conf/
  - users.xml (this file holds all user and their information)
- policies/
  - applicationpolicies/
    - administrator.xml
  - resourcepolicies/
    - myFirstQuestionnaire.xml
    - myFirstTemplate.xml
- questionnaires/
  - myFirstQuestionnaire/
    - analysis.xml
    - basicSettings.xml
    - grading.xml
    - interviewees.xml
    - notification.xml
    - questexplanation.xml
    - questexplanation\_de.xml
    - questexplanation\_en.xml
    - questionnaire.xml
    - questionnaire\_de.xml
    - questionnaire\_en.xml
- templates/
  - myFirstTemplate/
    - analysis.xml
    - basicSettings.xml
    - grading.xml
    - instances.xml
    - interviewees.xml
    - notification.xml
    - questexplanation.xml
    - questexplanation\_de.xml
    - questexplanation\_en.xml
    - questionnaire.xml
    - questionnaire\_de.xml
    - questionnaire\_en.xml

## Oracle Database

## Exist Database

# Administrator Tool

The administrator tool can be accessed at `qsys/adminloginv.qsys`. When using QSYS' native authentication mechanism the start page shows two links, one pointing to the admin login page mentioned above and one pointing to the authoring login page. However, when using the security middleware, that page is skipped and the authoring login page is shown as start page. Either way the admin login page can be accessed directly at `qsys/adminloginv.qsys`. The admin password is stored in `qsys.properties`.

The admin tool allows adding user accounts and directly accessing those. It also lists application properties, system properties, and session variables.

## Authorization

Permissions on resources are written to XACML policies. The owner of a resource can perform any action. To accommodate for a growing number of actions the owner's XACML action match allows "any action" as opposed to an explicit set of actions. As new actions get added the owner can perform them instantaneously.

### Administrator policy

Placing the following content into `data/policies/applicationpolicies/administrator.xml` grants the administrator access to all available resources when logging in as author. The string 'adminName' needs be adjusted accordingly.

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy PolicyId="administrator" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-
overrides">
  <Description>
    Administrator policy.
  </Description>
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">adminName</AttributeValue>
          <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
            DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </SubjectMatch>
      </Subject>
    </Subjects>
  </Target>
  <Resources>
    <AnyResource/>
  </Resources>
  <Actions>
    <AnyAction/>
  </Actions>
  <Rule RuleId="1" Effect="Permit"/>
</Policy>
```

## None-GUI features

This chapter deals with features added to QSYS for a small user group. As there was too little need of the features so far, no GUI was implemented and using them requires help by the administrator.

## Internationalization

To provide an interview in multiple languages, the following steps are advised:

Author:

1. Create the questionnaire in one language
2. Copy the first questionnaire and translate the questionnaire elements, do not change the structure of the questionnaire, such as adding more questions, or changing the order.

Administrator:

1. go to the directory of the original resource and copy questionnaire.xml and questexplanation.xml to reflect the language of the original resource (e.g. questionnaire\_de.xml and questexplanation\_de.xml)
2. go to the directory of the translated resource
3. doublecheck whether questexplanation.xml was translated too
4. copy questionnaire.xml and questexplanation.xml of the translation to the original version of the questionnaire by applying the same language convention as described above (questionnaire\_en.xml and questexplanation\_en.xml).

## Grading

## Analysis

	CLSD	DICH	PICT	CLSD_R	CLSD_M	CLSD_MM	INT	INT_M	SEMDIFF	OPEN	RANK_O	OPEN_M	IMG_MAP
BoxAndWhisker	✓												
BoxAndWhiskerWithBareHistogram	✓												
BoxAndWhiskerWithBareHistogram	✓												
MeanAndStdWithBareHistogram	✓												
MeanAndStdWithBareBar	✓												
MeanAndStdWithBareHistogramWithBareBar	✓												
ComplementaryBar		✓											
Histogram	✓	✓	✓										
BareBar													
NumericTextBinned									✓				
Text									✓	✓	✓		

*NumericTextBinned* erlaubt das Gruppieren von numerischen Antworten in gleichgrosse Gruppen. Hierzu muss das Minimum, Maximum und die Schrittweite angegeben werden. Derzeit funktioniert das Gruppieren nur, wenn das Minimum 1 ist. Ein Offset soll in einer späteren Version möglich sein.

## Miscellaneous

### Using Google Analytics

Add the code generated by Google into TOMCAT\_HOME/webapps/qsys/WEB-INF/js/google.js and include the new javascript analog to others in TOMCAT\_HOME/webapps/qsys/xsl/head.xsl.

## Update

Before running updates on your server back up the following resources and restore them afterwards:

- TOMCAT\_HOME/webapps/qsys/images/logos
- .properties files in TOMCAT\_HOME/webapps/qsys/WEB-INF/classes
- TOMCAT\_HOME/webapps/qsys/WEB-INF/web.xml

To update QSYS remove

- TOMCAT\_HOME/webapps/qsys.war
- TOMCAT\_HOME/webapps/qsys

Put the new version of qsys.war to TOMCAT\_HOME/webapps. After starting the server restore the resources from above and restart the server.

# Performance Testing

## Tool

For performance testing the Apache JMeter application is being used. Below is a summary of various tests that have been conducted and a documentation of each to help reproducing tests for additional questionnaires, or as starting point for testing other use cases. The respective test plans are part of the QSYS source code distribution. A brief introduction how to get started with JMeter follows. For further information on JMeter it is suggested to study the manual pages.

JMeter:

Download JMeter and start the application by double clicking the .jar-file.

Download the JMeter test plans.

Menu: *File > Open* > choose an existing test plan.

Do adjustments for the questionnaire to be tested as described below.

Add additional listeners for test results.

Menu: *Run > Start* or *Ctrl+R* for running a test.

Look at the listener output for test results.

## CSV Data

To test the application with real parameters, JMeter allows putting them into a CSV data file.

Explain how to generate:

1. data into data.csv
2. parameters into CSV
- 3.

## Test Descriptions and Results

For testing QSYS' interviewing performance allowing a somewhat simple test plan, it is recommended to adjust the questionnaire settings to not require authentication of the interviewee.

The table below summarizes load test results. Note that these results heavily depend on both, the test machine's resources and the server's resources and configurations. (infos on server settings here!)

	Questions	Authentication	Start Text	1 Q/P	1 Page	Sep.	End Text	Threads	Ramp-Up Period [sec]	No of Samples	Latest Sample [ms]	Average [ms]	Deviation [ms]	Throughput [/min]	Median [ms]
Test 1	13	no	no		yes		no								
Test 2	13							40	60	80	321	495	148	80	600
								50	60	100	357	506	158	100	607
								60	60	120	385	586	185	120	608
								70	60	140	372	704	200	139	762
								80	60	160	630	3524	1387	150	3808
								80	60	160	416	1401	503	157	1398
								80	60	160	327	910	290	158	896
								100	60	200	285	595	214	198	667
								100	60	200	286	570	197	198	634

### **Test 1: Load Questionnaire**

### **Test 2: Loading and Submitting a Questionnaire**

Generating test data:

1. Fill out your questionnaire just like an interviewee would do.
2. Submit the questionnaire
3. Open the log file and grab the data that are logged for this purpose, looking like this:

Parameters to be used for stress testing:

\_q11\_type,\_q11\_alternatives\_2,\_q11\_alternatives\_1,

Values to be used for stress testing:

openendedMatrixQ,bbb,aaa,

Note, if your interview requires multiple submits, the submitted key/value pairs are spread over multiple log entries.

To perform the test as described in this example you need to change the basic settings, saying to display all questions on one page. Additionally, you need to change the participation mode, saying that anybody can participate. However, the test plan can be extended to accept other settings.

### **Data file:**

For this example place a file named data.csv in the same directory as the Test Plan below is in. Get the list of values as described above and paste it into data.csv. When creating your own data, be aware that whitespaces between items are not removed so you may want to omit them. There must not be a line break at the end of the last line.

Note that when using a single line of data as described above, all submissions carry the same information.

### Test Plan: (interviewee.jmx)

Open the jMeter GUI and adjust the following:

- Test Plan/Interviewee  
adjust the number of repetitions
- Test Plan/Interviewee/CSV Answers: Variable Names  
Get the parameter list as described above and paste it here or create your own one (the list must not contain whitespaces between the items, at least on a Mac).
- Test Plan/Interviewee/Load Interview: Send Parameter  
use the questionnaire ID as the value
- Test Plan/Interviewee/Submit: adjust all parameters

Note: when working on a Mac the GUI does not support the copy/paste, so you may prefer editing the xml file directly.

```
<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="2.1">
  <hashTree>
    <TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="Test Plan" enabled="true">
      <stringProp name="TestPlan.comments"></stringProp>
      <boolProp name="TestPlan.functional_mode">false</boolProp>
      <boolProp name="TestPlan.serialize_threadgroups">false</boolProp>
      <elementProp name="TestPlan.user_defined_variables" elementType="Arguments" guiclass="ArgumentsPanel"
testclass="Arguments" testname="User Defined Variables" enabled="true">
        <collectionProp name="Arguments.arguments"/>
      </elementProp>
      <stringProp name="TestPlan.user_define_classpath"></stringProp>
    </TestPlan>
    <hashTree>
      <ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="interviewee" enabled="true">
        <elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass="LoopControlPanel"
testclass="LoopController" testname="Loop Controller" enabled="true">
          <boolProp name="LoopController.continue_forever">false</boolProp>
          <stringProp name="LoopController.loops">1</stringProp>
        </elementProp>
        <stringProp name="ThreadGroup.num_threads">4</stringProp>
        <stringProp name="ThreadGroup.ramp_time">1</stringProp>
        <longProp name="ThreadGroup.start_time">1239012650000</longProp>
        <longProp name="ThreadGroup.end_time">1239012650000</longProp>
        <boolProp name="ThreadGroup.scheduler">false</boolProp>
        <stringProp name="ThreadGroup.on_sample_error">continue</stringProp>
        <stringProp name="ThreadGroup.duration"></stringProp>
        <stringProp name="ThreadGroup.delay"></stringProp>
      </ThreadGroup>
      <hashTree>
        <CSVDataSet guiclass="TestBeanGUI" testclass="CSVDataSet" testname="CSV Answers" enabled="true">
          <stringProp name="delimiter">,</stringProp>
          <stringProp name="fileEncoding"></stringProp>
          <stringProp name="filename">data.csv</stringProp>
          <boolProp name="quotedData">false</boolProp>
          <boolProp name="recycle">true</boolProp>
          <stringProp name="shareMode">All threads</stringProp>
          <boolProp name="stopThread">false</boolProp>
          <stringProp name="variableNames">_q11_alternatives_1,_q11_alternatives_2</stringProp>
        </CSVDataSet>
      </hashTree>
      <HTTPSampler guiclass="HttpTestSampleGui" testclass="HTTPSampler" testname="Load Interview" enabled="true">
        <elementProp name="HTTPSampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel"
testclass="Arguments" testname="User Defined Variables" enabled="true">
          <collectionProp name="Arguments.arguments">
            <elementProp name="questId" elementType="HTTPArgument">
              <boolProp name="HTTPArgument.always_encode">false</boolProp>
              <stringProp name="Argument.value">StressTest</stringProp>
              <stringProp name="Argument.metadata">=</stringProp>
              <boolProp name="HTTPArgument.use_equals">true</boolProp>
              <stringProp name="Argument.name">questId</stringProp>
            </elementProp>
          </collectionProp>
        </elementProp>
      </HTTPSampler>
    </hashTree>
  </jmeterTestPlan>
</xml>
```

```

        </elementProp>
    </collectionProp>
</elementProp>
<stringProp name="HTTPSampler.domain">localhost</stringProp>
<stringProp name="HTTPSampler.port">8080</stringProp>
<stringProp name="HTTPSampler.protocol"></stringProp>
<stringProp name="HTTPSampler.contentEncoding">UTF-8</stringProp>
<stringProp name="HTTPSampler.path">qsys/answerLogin.qsys</stringProp>
<stringProp name="HTTPSampler.method">GET</stringProp>
<boolProp name="HTTPSampler.follow_redirects">false</boolProp>
<boolProp name="HTTPSampler.auto_redirects">true</boolProp>
<boolProp name="HTTPSampler.use_keepalive">true</boolProp>
<boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
<stringProp name="HTTPSampler.FILE_NAME"></stringProp>
<stringProp name="HTTPSampler.FILE_FIELD"></stringProp>
<stringProp name="HTTPSampler.mimetype"></stringProp>
<stringProp name="HTTPSampler.monitor">false</stringProp>
<stringProp name="HTTPSampler.embedded_url_re"></stringProp>
</HTTPSampler>
<hashTree/>
<HTTPSampler guiclass="HttpTestSampleGui" testclass="HTTPSampler" testname="Submit" enabled="true">
    <elementProp name="HTTPSampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel"
testclass="Arguments" testname="User Defined Variables" enabled="true">
        <collectionProp name="Arguments.arguments">
            <elementProp name="_q11_alternatives_1" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.value">${_q11_alternatives_1}</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
                <stringProp name="Argument.name">_q11_alternatives_1</stringProp>
            </elementProp>
            <elementProp name="_q11_alternatives_2" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.value">${_q11_alternatives_2}</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
                <stringProp name="Argument.name">_q11_alternatives_2</stringProp>
            </elementProp>
            <elementProp name="_q11_type" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.value">openendedMatrixQ</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
                <stringProp name="Argument.name">_q11_type</stringProp>
            </elementProp>
        </collectionProp>
    </elementProp>
<stringProp name="HTTPSampler.domain">localhost</stringProp>
<stringProp name="HTTPSampler.port">8080</stringProp>
<stringProp name="HTTPSampler.protocol"></stringProp>
<stringProp name="HTTPSampler.contentEncoding"></stringProp>
<stringProp name="HTTPSampler.path">qsys/addfilledout.qsys</stringProp>
<stringProp name="HTTPSampler.method">POST</stringProp>
<boolProp name="HTTPSampler.follow_redirects">false</boolProp>
<boolProp name="HTTPSampler.auto_redirects">true</boolProp>
<boolProp name="HTTPSampler.use_keepalive">true</boolProp>
<boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
<stringProp name="HTTPSampler.FILE_NAME"></stringProp>
<stringProp name="HTTPSampler.FILE_FIELD"></stringProp>
<stringProp name="HTTPSampler.mimetype"></stringProp>
<stringProp name="HTTPSampler.monitor">false</stringProp>
<stringProp name="HTTPSampler.embedded_url_re"></stringProp>
</HTTPSampler>
<hashTree/>
<URLRewritingModifier guiclass="URLRewritingModifierGui" testclass="URLRewritingModifier" testname="HTTP
URL Re-writing Modifier" enabled="true">
    <stringProp name="argument_name">jsessionid</stringProp>
    <boolProp name="path_extension">true</boolProp>
    <boolProp name="path_extension_no_equals">false</boolProp>
    <boolProp name="path_extension_no_questionmark">false</boolProp>
    <boolProp name="cache_value">false</boolProp>
</URLRewritingModifier>
<hashTree/>
<ResultCollector guiclass="ViewResultsFullVisualizer" testclass="ResultCollector" testname="View Results
Tree" enabled="true">
    <boolProp name="ResultCollector.error_logging">false</boolProp>
    <objProp>
        <name>saveConfig</name>
        <value class="SampleSaveConfiguration">
            <time>true</time>
            <latency>true</latency>
            <timestamp>true</timestamp>
            <success>true</success>
            <label>true</label>
            <code>true</code>
            <message>true</message>
            <threadName>true</threadName>
        </value>
    </objProp>
</ResultCollector>

```

```
<dataType>true</dataType>
<encoding>false</encoding>
<assertions>true</assertions>
<subresults>true</subresults>
<responseData>false</responseData>
<samplerData>false</samplerData>
<xml>true</xml>
<fieldNames>true</fieldNames>
<responseHeaders>true</responseHeaders>
<requestHeaders>true</requestHeaders>
<responseDataOnError>false</responseDataOnError>
<saveAssertionResultsFailureMessage>false</saveAssertionResultsFailureMessage>
<assertionsResultsToSave>0</assertionsResultsToSave>
<bytes>true</bytes>
<url>true</url>
</value>
</objProp>
<stringProp name="filename"></stringProp>
</ResultCollector>
<hashTree/>
</hashTree>
</hashTree>
</hashTree>
</jmeterTestPlan>
```