

pum2pum

Test cases

2012-05-15

Contents

1	System quality tests	4
1.1	Usability	4
1.1.1	4
1.2	Performance	4
1.2.1	4
2	Testing the server	4
2.1	Argument handling	5
2.1.1	5
2.1.2	5
2.2	App configuration	5
2.3	Networking	5
2.3.1	5
2.3.2	5
2.3.3	6
3	Testing database abstraction layer	6
3.1	Users	6
3.1.1	6
3.1.2	6
3.2	Categories	6
3.2.1	6
3.2.2	6
3.3	Subforum	7
3.3.1	7
3.3.2	7
3.4	Threads	7
3.4.1	7
3.4.2	7
3.5	Posts	7
3.5.1	7
3.5.2	7
3.6	Answers	7
3.6.1	7
3.6.2	7
3.7	Overall	8
3.7.1	Callbacks on new data	8

4	Testing Enyo-kinds	8
4.1	Utilities	8
4.1.1	Date formatting	8
5	Integration testing	8
5.1	Post a new thread	8
5.1.1	8
5.2	Reply to a thread	9
5.2.1	9

Versions

Version	Date	Changes	By
0.4	2012-05-15	Corrections from feedback.	nicol271
0.3	2012-05-02	More integration tests.	nicol271
0.2	2012-04-19	Database abstraction tests.	nicol271
0.1	2012-04-10	First version.	nicol271

Introduction

This document contains a description for test cases of the system. For information on running test scripts, please consult the *Installation and operation manual*. Information about test runs and test results may be found in the *Test log* document.

1 System quality tests

Tests overall, system-wide qualities.

1.1 Usability

1.1.1

- | | |
|-----------------------|---|
| How to measure | A tester is given the following tasks: read a thread, reply to the thread, and create a new thread. |
| Requirement | 3 out of 4 randomly chosen testers must be able to fulfill the tasks, see R.1.1.0.1 |

1.2 Performance

1.2.1

- | | |
|-----------------------|---|
| How to measure | By navigating through the system, measuring the response time using the net-panel in the browser web developer tools. The time is measured using a standard PC client with a broadband network connection to a web-server. The response time is measured as the total time between two interactions with the application. The initial time to start the application (load the web page) is not included as response time. |
| Requirement | Response time less than 2 seconds, see R.1.2.0.1 |

2 Testing the server

Test scripts for these tests can be found in `test/node/server.test.js`. For information on running test scripts, please consult the *Installation and operation manual*.

2.1 Argument handling

Tests if the server handles incorrect command-line arguments.

2.1.1

Input Server is called without giving an appname to run

Output A message describing the usage is shown

2.1.2

Input An incorrect appname is given

Output A message saying that the app could not be found

2.2 App configuration

Tests if handling of the app-definitions is correct.

This part of the testing has not been written, since the purpose of the server in the project only is to run the forum application. This kind of tests would only have been useful if the server was more generic and actually was supposed to run many different applications.

2.3 Networking

Tests if the server really can serve requests correctly.

Prerequisites: The server is running an existing app which returns known data.

2.3.1

Input A plain HTTP GET-request without data is sent to the port that the server is listening on

Output A response with HTTP-status code 200

2.3.2

Input A plain HTTP-request is sent to the port that the server is listening on

Output The body of the HTTP response consists of the known data declared in the app

2.3.3

- Input** A HTTP-request with a path to a non-existing file is sent to the server
- Output** A response with HTTP-status code 404

3 Testing database abstraction layer

Unit-testing of the part of the system which interacts directly with the database.

Test scripts for this tests can be found in `src/testdriver/tests/`. For information on running test scripts, please consult the *Installation and operation manual*.

3.1 Users

3.1.1

- Input** Log into the database with a name and a callback-function
- Output** The user is logged in and the callback-function is called

3.1.2

- Input** Get list of users from database
- Output** The logged-in users are recieved

3.2 Categories

3.2.1

- Input** Create a new category with a name
- Output** The category is created, no error-callback is recieved

3.2.2

- Input** Get categories from database
- Output** The supplied callback-function is called with a list of all category-objects in the database

3.3 Subforum

3.3.1

Input Create a new subforum in a category
Output The subforum is created, no error-callback is recieved

3.3.2

Input Get all subforums for a specified category
Output A list of all subforum-objects for the specified category

3.4 Threads

3.4.1

Input Create a new thread in a subforum
Output The thread is created, no error-callback is recieved

3.4.2

Input Get all threads for a specified subforum
Output A list of all thread-objects for the specified subforum

3.5 Posts

3.5.1

Input Create a new post in a thread
Output The post is created in the thread, no error-callback is recieved

3.5.2

Input Get all posts for a specified thread
Output A list of all post-objects for the specified thread

3.6 Answers

3.6.1

Input Create a new answer to a thread
Output The answer is created, no error-callback is recieved

3.6.2

Input Get all answers for a specified thread
Output A list of all answer-objects for the specified thread

3.7 Overall

3.7.1 Callbacks on new data

- Input** Gets a list of objects and then inserts a new object into the database.
- Output** The callback-function is called twice: once with the first query result, and then again after inserting the new object.

4 Testing Enyo-kinds

Unit- and integration testing of the modules which the application consists of.

4.1 Utilities

4.1.1 Date formatting

- Input** Javascript date objects, such as `Date(2011, 03, 01, 01, 02, 30)`. Remember that JS months are zero-based.
- Output** The date formatted on YYYY-MM-DD HH:MM:SS format.
In the case above: 2011-04-01 01:02:30

5 Integration testing

Manual testing of the application as a whole.

5.1 Post a new thread

5.1.1

- Prerequisites** The server is running and the browser is located at the forum start page.
- Input** "Test user" is entered as login, and the login-button is clicked. A forum name is clicked. The "Create new thread"-button is clicked, a title and text is entered, and the post button is pressed.
- Output** A thread with the entered name is created and shows up immediately in the list of all threads. "Test user" is shown as the creator of the thread. The thread is shown when its title is clicked.

5.2 Reply to a thread

5.2.1

Prerequisites	The server is running and the browser is located at a thread.
Input	The "Post"-button is pressed. A text field and a send-button shows up. A text is entered in the field and the button is pressed.
Output	The answer shows up correctly and without reloading the page.