MarketPlace

Contents

[Introduction 1](#_Toc448325189)

[Requirements 2](#_Toc448325190)

[Instructions 3](#_Toc448325191)

[Creating a Course and a Student 1](#_Toc448325192)

[Setting up The Environment 2](#_Toc448325193)

[Building and Deploying Module 3](#_Toc448325194)

# Introduction

The MarketPlace is a Blackboard Module created to motivate students to do their work and help bring a game-like environment to the class. This project idea was created by [Dr. David Thornton](http://www.jsu.edu/mcis/people/thornton.html); work began with a Masters student (What was his name?), and continued by Sari Sabouh, an Undergraduate student studying Computer Science.

The MarketPlace contains 3 kinds of items: Instant, Continuous, and Passive Items. Instant Items are expired once they have been activated, like adding 5 points to a test grade. Continuous Items stay as long as the item specifies, like adding a specified percentage to everything submitted in that period. Passive Items are more like a contract between the teacher and the student. For example, if the Please Reply Item is bought and activated, then the teacher is required to reply to that student’s email at any time. It also has different views for a student and an instructor.

To get credentials, please ask Dr. Thornton.

# Requirements

Blackboard is very specific about what it takes and what it accepts. This documentation will cover everything that you have to do or have, starting with general requirements, like the following:

1. Eclipse Java EE IDE for Web Developers.
2. Java 1.7 (I had to use 1.6 for a bit and then it accepted 1.7 by itself).
3. Bb-cms-admin.jar, bb-platform.jar and bb-taglibs.jar. These jars are included in the project repository.
4. SQL and Databases (MySQL and Oracle or SQL Server), JAVA, JSP, JS, jQuery, JUnit, and basic Maven.

# Instructions

## Creating a Course and a Student

To create a course, please follow these steps:

1. Open <205.174.62.96> with **ssabouh** as username and password.
2. Click on System Admin, Courses, Create Course

To create a student, please follow the same steps as the steps before, but instead of Courses go to Users.

## Setting up The Environment

To set up the project on your local machine, please use GitHub and pull this [repository](https://github.com/SariSabouh/marketplace) to your preferred location. After you have the project and all the requirements on your computer, please follow these steps:

1. Open Eclipse Java EE IDE.
2. Click File.
3. Click Import -> Existing Maven Projects.
4. Browse for the directory of your project.
5. Click Finish.

## Building and Deploying Module

To Build the project and deploy it to Blackboard, please follow these steps (If it is not your first time running the project, skip step 1 and step 2):

1. Right-Click project and choose Run As -> Run Configurations
2. Inside Goals put “clean package”, check “Update Snapshots” and, if you want it to build fast, then check “Skip Tests” and click Apply. Make sure to run unit tests after adding or changing anything.
3. Click Run (Or the PLAY button on the top of the window).
4. Go to <205.174.62.96>, log in as **ssabouh** with password **ssabouh.**
5. Go to System Admin – Building Blocks – Installed Tools – Upload Building Blocks (Remove it if it already exists).
6. Then go back to your course and Add Course Module – Other and you will see MarketPlace, click Add.

Following these steps you will have the module ready to be used for that course.

# Things to be Aware of

Working with Blackboard is not really fun, until you understand how picky it is. There are multiple things to be aware of when working with it. Like the following:

1. In the end of creating any object in the database, whether it was a primary key, a foreign key, or a table, it has to start with the vendor name, which is jsu for us.
2. When trying to create a new Id of a Blackboard object, use the newId method like this: ***Id.newId(GradableItem.DATA\_TYPE)***, GradableItem is a Blackboard object.
3. Not all browsers work the same. “Click” functions in jQuery would work almost anywhere in FireFox, like in a dropdown box. But in Chrome, it has to be “onChange” rather than click.
4. MySQL is used for testing only. You have to set up a local MySQL database server and set up all the tables, which can be found in the schema.xml file in the project, to be able to unit test.
5. Some methods were created inside of BlackboardHandler and MarketPlaceDAO just to help running test cases. Blackboard is very picky about using its objects if you are not on the website, so a lot of if statements with the variable testing had to be created.
6. To check the logs of the module, go to System Admin – Logs – System Logs - tomcat/stdout-stderr-xxxx.log

This [website](http://www.brucephillips.name/blog/index.cfm/2012/10/12/Tutorial-On-Using-Building-Block-Feature-That-Creates-Database-Tables-In-the-Blackboard-Database) and this [one](https://blog.alltheducks.com/post/introduction_to_blackboard_schema_xml/) will talk about Blackboard Database requirements.

# Features

This project has been under development since September 2015. I stopped working on it in April and I was able to add a lot of features and make it fully functional. The main features are:

1. Adding gold to all students at once, if the person logged in is an instructor.
2. Resetting project to its original value with preset items.
3. Settings having only Name and Value that can be set as default and will be applied forever or until it’s changed again or Database is reset.
4. Adding, Editing and Prefilling items.
5. Community Items that can be purchased by one student by submitting a down payment. The rest of the class can participate on the purchase to activate this item. Preferably it should affect DUEDATE or Number of Attempts.
6. If you click on item, description appears in bottom.

# Future Features

A module like this is ready for expansion and integration with other JSU modules. Like the Leaderboard, Avatar and, of course, the QuestPath. All of them combined will create the perfect game-like class. The MarketPlace is still missing the logic to integrate them together. Giving the teacher more view of the database, or what he needs to see in the database, like who bought what and how many times. Rewarding students for completing an assignment, test or project by granting them gold. A method called passesCondition already exists in the project but it is not implemented and not tested. The idea of conditions being assigned to columns is as simple as 1st, 2nd and 3rd place. If you get 100% in this grade then you receive 100 gold. If you get 75% or above you will receive 50 gold. Anything below 75% will not be rewarded. Of course, the numbers are arbitrary and flexible to the requirements of the instructor.