

Hook Test Suite

Client Server Test Suite

Scrubbing Script Scenario

Case 1: Submitting a Correctly Prepared Package

Steps	Expected
Create a main folder to create the package	Main folder is created
Inside the main folder, create 3 folders: Current Year, Previous Years, and Exceptions	The three sub-folders are created
In the Current Year and Previous Years folder enter subfolders of student submission folders with the name in the form FirstName_LastName_StudentNumber	Student Submission Folders are created.
Have java, c, and/or c++ files in the student submission folders (these files may be in subdirectories of the student submission folder mentioned above)	The student submission folders contain source files
Open the command line	Command line is opened
In the terminal, go to the script location	The terminal is set to the script location
Run the command <code>python scrub.py package_location destination_location database_username database_password</code>	The command is run
No errors should appear	No errors appeared
Navigate to the destination folder path included in the python command above	The student submission folders should be scrubbed

Steps	Expected
Open the scrubbed source files	The first name, last name, and student number should be hashed wherever it appears in the source code

Case 2: Submitting a Prepared Package with Incorrectly Named Student Submission Folder

Steps	Expected
Create a main folder to create the package	Main folder is created
Inside the main folder, create 3 folders: Current Year, Previous Years, and Exceptions	The three sub-folders are created
In the Current Year and Previous Years folder enter subfolders of student submission folders with the name in the form FirstName_LastName_StudentNumber	Student Submission Folders are created.
Have one student submission folder that does not comply with the above format	A student submission folder does not comply with the above stated format
Have java, c, and/or c++ files in the student submission folders (these files may be in subdirectories of the student submission folder mentioned above)	The student submission folders contain source files
Open the command line	Command line is opened
In the terminal, go to the script location	The terminal is set to the script location
Run the command <code>python scrub.py package_location destination_location database_username database_password</code>	The command is run

Steps	Expected
Returns an error that one folder was unable to be scrubbed.	An error appeared
Navigate to the destination folder path included in the python command above	Only the student submission folders that followed the naming format above should be present
Open the scrubbed source files	The first name, last name, and student number should be hashed wherever it appears in the source code

Case 3: Submit A Package with non-source code files in the Student Submission Folder

Steps	Expected
Create a main folder to create the package	Main folder is created
Inside the main folder, create 3 folders: Current Year, Previous Years, and Exceptions	The three sub-folders are created
In the Current Year and Previous Years folder enter subfolders of student submission folders with the name in the form FirstName_LastName_StudentNumber	Student Submission Folders are created.
Have java, c, and/or c++ files in the student submission folders (these files may be in subdirectories of the student submission folder mentioned above)	The student submission folders contain source files
Have a .txt / image file in the student submission folder (either directly in the folder or in a sub-directory)	A non-source code file is saved in at least one student submission folder
Open the command line	Command line is opened

Steps	Expected
In the terminal, go to the script location	The terminal is set to the script location
Run the command <code>python scrub.py package_location destination_location database_username database_password</code>	The command is run
No errors should appear	No errors appeared
Navigate to the destination folder path included in the python command above	The student submission folders should be scrubbed
Open the scrubbed source files	The first name, last name, and student number should be hashed wherever it appears in the source code
Looking at the scrubbed files in the scrubbed student submission folder, the .txt or image files should not be present	No non-source files are present in the scrubbed student submission folder

Case 4: Submitting a Package with incorrect database credentials

Steps	Expected
Create a main folder to create the package	Main folder is created
Inside the main folder, create 3 folders: Current Year, Previous Years, and Exceptions	The three sub-folders are created
In the Current Year and Previous Years folder enter subfolders of student submission folders with the name in the form FirstName_LastName_StudentNumber	Student Submission Folders are created.

Steps	Expected
Have java, c, and/or c++ files in the student submission folders (these files may be in subdirectories of the student submission folder mentioned above)	The student submission folders contain source files
Open the command line	Command line is opened
In the terminal, go to the script location	The terminal is set to the script location
Run the command <code>python scrub.py package_location destination_location database_username database_password</code> with an incorrect database username and / or password	The command is run
An error should appear that the script could not connect to the database	Error appeared
Navigate to the destination folder path included in the python command above	There should be no scrubbed data files created.

Case 5: Submitting a Package that is not in the expected hierarchy

Steps	Expected
Create a main folder to create the package	Main folder is created
Inside the main folder, create 3 folders: Something, Something Else, and Exceptions (Note: the titles of the folders are not what the script is expecting)	The three sub-folders are created

Steps	Expected
In the Something and Something Else folder enter subfolders of student submission folders with the name in the form FirstName_LastName_StudentNumber	Student Submission Folders are created.
Have java, c, and/or c++ files in the student submission folders (these files may be in subdirectories of the student submission folder mentioned above)	The student submission folders contain source files
Open the command line	Command line is opened
In the terminal, go to the script location	The terminal is set to the script location
Run the command <code>python scrub.py package_location destination_location database_username database_password</code>	The command is run
An error saying the package is not in the correct format is displayed	Error appeared
Navigate to the destination folder path included in the python command above	There should be no scrubbed data folders created

Client Portal Test Suite

Login Scenario

Case 1: Login Sucessfully

Steps	Expected Results
Navigate to the login page	The login page is displayed
Enter a valid username	Username is entered

Steps	Expected Results
Enter a valid password	Password is entered
Click 'Login' or press the Enter key	The user is logged in successfully

Case 2: Click Login without entering credentials

Steps	Expected Results
Navigate to the login page	The login page is displayed
Click 'Login' or press the Enter key	Errors show that a username and password are needed

Case 3: Click Login without entering the username

Steps	Expected Results
Navigate to the login page	The login page is displayed
Enter a valid password	Password is entered
Click 'Login' or press the Enter key	Errors show that a username is needed

Case 4: Click Login without entering the password

Steps	Expected Results
Navigate to the login page	The login page is displayed
Enter a valid username	Username is entered
Click 'Login' or press the Enter key	Errors show that a password is needed

Case 5: Click Login without entering the password

Steps	Expected Results
Navigate to the login page	The login page is displayed

Steps	Expected Results
Enter a non-created username	Username is entered
Enter a non-created password	Password is entered
Click 'Login' or press the Enter key	Errors show that the account does not exist

Case 6: Account is locked out

Steps	Expected Results
Navigate to the login page	The login page is displayed
Enter a valid username	Username is entered
Enter the incorrect password	Password is entered
Click 'Login' or press the Enter key	Errors show that the password is incorrect
Enter a valid username	Username is entered
Enter the incorrect password	Password is entered
Click 'Login' or press the Enter key	Errors show that the password is incorrect
Enter a valid username	Username is entered
Enter the incorrect password	Password is entered
Click 'Login' or press the Enter key	Error shows that the password has been entered incorrectly three times and the account is now locked out. The user will have to contact the system administrator to reset the password and unlock the account

User Upload Scenario

Case: User Uploads Student Submissions Successfully For an Active Course

Steps	Expected
User logs in to portal	Login successful dashboard is displayed
User clicks on “Upload” under the “Submission” menu section in the side nav	Interface for adding submissions displays on the screen
User selects desired course name from drop down	Course is selected
User selects desired assignment number	Assignment number is selected
User clicks on ‘Browse’ to upload files	File explorer is opened and appears on screen
User selects and confirms the assignments	Assignments are queued in the system for submission
User clicks ‘Launch’	Assignments are uploaded onto the client server.

User View Results Scenario

Case 1: View Detection Processed Unsuccessfully

Steps	Expected
User logs in to portal	Login successful
User clicks on “View Resules” under the “Hook Job Management” menu section in the side nav	Interface for viewing results renders on the screen, indicates failed detection
User clicks on detection results	Pop up containing error message.

Case 2: View Detection Results Processed Successfully

Steps	Expected
User logs in to portal	Login successful

Steps	Expected
User clicks on “View Results” under the “Hook Job Management” menu section in the side nav	Interface for viewing results renders on the screen
User clicks on detection results	Interface loads showing user a list of matches found in the detection procedure
User selects a match to view	View expands a card with two code views side by side
User selects two students to compare code view	Expanded code views display the selected student’s plagiarised code

User View Assignments Scenario

Case: Viewing various student’s code for COSC1P02 from a previous semester

Steps	Expected
User logs in to portal	Login successful
User clicks on “View Submissions” under the “Submission” menu section in the side nav	Interface for adding submissions renders on the screen
User selects previous semester from drop down	Semester is selected
User selects course code from drop down	Course code is selected
User selects desired assignment number	Assignment number is selected and students and assignment submissions from that course are populated for viewing
User selects student from the list of students	The selected student’s submission contents will be populated under ‘Files’ for viewing

Steps	Expected
User selects *.java file	The student's code for the selected *.java file will be generated for viewing

User Course/Assignment Creation Scenario

Case 1: Add a course without an assignment

Steps	Expected Result
Login to the portal	Login successful
Click on the 'Create Course' link under the 'Course Management' header in the navigation bar	The Create Course page is displayed
Enter the Program Code	Program Code is entered
Enter the Course Code	Course Code is entered
Enter the Academic Year the course will be active	The year is entered
Enter the semester that the course will be active	The semester is entered
Click the Create button	The button is clicked, the form is reset
Click the Search Courses link on the Navigation bar	The Search Courses page is displayed
Search for the course that you just added	The course should be present

Case 2: Add a course with an assignment

Steps	Expected Result
Login to the portal	Login successful
Click on the 'Create Course' link under the 'Course Management' header in the navigation bar	The Create Course page is displayed

Steps	Expected Result
Enter the Program Code	Program Code is entered
Enter the Course Code	Course Code is entered
Enter the Academic Year the course will be active	The year is entered
Enter the semester that the course will be active	The semester is entered
Enter the name of an assignment	The assignment name is created
Enter the open date for the assignment	Open date is entered
Enter the close date for the assignment	Close date is entered
Click the '+' button	Another assignment row will appear
Fill in this assignment row with name, open date, and close date	All the information is entered
Click the Create button	The button is clicked, the form is reset
Click the Search Courses link on the Navigation bar	The Search Courses page is displayed
Search for the course that you just added	The course should be present, and you will see two assignments within the course

Case 3: Add a course with missing information

Steps	Expected Result
Login to the portal	Login successful
Click on the 'Create Course' link under the 'Course Management' header in the navigation bar	The Create Course page is displayed

Steps	Expected Result
Do not enter the Program Code	Program Code is not entered
Enter the Course Code	Course Code is entered
Enter the Academic Year the course will be active	The year is entered
Enter the semester that the course will be active	The semester is entered
Enter the name of an assignment	The assignment name is created
Enter the open date for the assignment	Open date is entered
Enter the close date for the assignment	Close date is entered
Click the '+' button	Another assignment row will appear
Fill in this assignment row with name, open date, and close date	All the information is entered
Click the Create button	An error is displayed that the Program Code is needed to create the course
Repeat this test case for each data field	An error should display for any data field that isn't populated

Case 4: Delete an added assignment row

Steps	Expected Result
Login to the portal	Login successful
Click on the 'Create Course' link under the 'Course Management' header in the navigation bar	The Create Course page is displayed
Enter the Program Code	Program Code is entered
Enter the Course Code	Course Code is entered

Steps	Expected Result
Enter the Academic Year the course will be active	The year is entered
Enter the semester that the course will be active	The semester is entered
Enter the name of an assignment	The assignment name is created
Enter the open date for the assignment	Open date is entered
Enter the close date for the assignment	Close date is entered
Click the '+' button	Another assignment row will appear
Click the '-' button next to the assignment row	The assignment row will disappear
Click the Create button	The button is clicked, the form is reset
Click the Search Courses link on the Navigation bar	The Search Courses page is displayed
Search for the course that you just added	The course should be present, and you will see one assignment within the course

Edit Course Scenario

Case 1: Editing a course

Steps	Expected Result
Login to the portal	Login successful
Navigate to any course and click the pencil icon to direct to the 'Edit Course' page	The 'Edit Course' page is displayed

Steps	Expected Result
The 'Edit Course' page will show the course information (program code, course code, year, semester, assignments) that was previously entered	The previously entered information is being populated
Change the year the course is offered	The year is changed
Change the semester the course is offered	The semester the course is offered is changed
Change the program code of the course	The program code is changed
Change the course code of the course	The course code is changed
Change the name of the assignment	The name is changed
Click the '+' button	A new assignment row appears
Enter information for another assignment	Information for another assignment is entered
Click the 'Save' button	The course is saved and the site redirects to the dashboard
Click the Search Courses link on the Navigation bar	The Search Courses page is displayed
Search for the course that you just added	The course should be present, and you will see two assignments within the course

Case 2: Editing a course with missing information

Steps	Expected Result
Login to the portal	Login successful
Navigate to any course and click the pencil icon to direct to the 'Edit Course' page	The 'Edit Course' page is displayed

Steps	Expected Result
The 'Edit Course' page will show the course information (program code, course code, year, semester, assignments) that was previously entered	The previously entered information is being populated
Change the year the course is offered	The year is changed
Change the semester the course is offered	The semester the course is offered is changed
Change the program code of the course	The program code is changed
Change the course code of the course	The course code is changed
Delete the name of the assignment	The name is deleted
Click the 'Save' button	An error appears that the name of the assignment is needed to save the course
Repeat this test case for each data field	An error should display for any data field that isn't populated

Case 3: Delete an added assignment row

Steps	Expected Result
Login to the portal	Login successful
Navigate to any course and click the pencil icon to direct to the 'Edit Course' page	The 'Edit Course' page is displayed
The 'Edit Course' page will show the course information (program code, course code, year, semester, assignments) that was previously entered	The previously entered information is being populated
Change the year the course is offered	The year is changed

Steps	Expected Result
Change the semester the course is offered	The semester the course is offered is changed
Change the program code of the course	The program code is changed
Change the course code of the course	The course code is changed
Change the name of the assignment	The name is changed
Click the '+' button	A new assignment row appears
Click the '-' button next to the assignment row	The assignment row will disappear
Click the 'Save' button	The course is saved and the site redirects to the dashboard
Click the Search Courses link on the Navigation bar	The Search Courses page is displayed
Search for the course that you just added	The course should be present, and you will see one assignment within the course

Case 4: Deleting a course

Steps	Expected Result
Login to the portal	Login successful
Navigate to any course and click the pencil icon to direct to the 'Edit Course' page	The 'Edit Course' page is displayed
The 'Edit Course' page will show the course information (program code, course code, year, semester, assignments) that was previously entered	The previously entered information is being populated
Click the 'Delete' button	The course is deleted and the site redirects to the dashboard

Steps	Expected Result
Click the Search Courses link on the Navigation bar	The Search Courses page is displayed
Search for the course that you just deleted	The course should be not appear in the search

Case 5: Delete an assignment from a course

Steps	Expected Result
Login to the portal	Login successful
Navigate to any course and click the pencil icon to direct to the 'Edit Course' page	The 'Edit Course' page is displayed
The 'Edit Course' page will show the course information (program code, course code, year, semester, assignments) that was previously entered	The previously entered information is being populated
Click the '-' button next to the assignment row	The assignment row will disappear
Click the 'Save' button	The course is saved and the site redirects to the dashboard
Click the Search Courses link on the Navigation bar	The Search Courses page is displayed
Search for the course that you just added	The course should be present, the assignment that you clicked the '-' button for is not displayed with the course

Dashboard Scenario

Case: Current Courses Functionality

Steps	Expected Result
Login to the portal	Login successful

Steps	Expected Result
The dashboard should be loaded automatically	Dashboard is displayed
The Semester being shown in the title should be the current semester	The Semester and Year shown is the current semester and year
All courses being shown under the heading should match departments (I.e If the heading is COSC, all courses underneath that heading should be COSC courses)	All the courses match the heading
If the pencil in the corner of the course is clicked then the site is redirected to the 'Edit Course' page	The site was re-directed to the 'Edit Course' page
Click on the 'Current Courses' link under the 'Course Management' header in the navigation bar	The Dashboard page should be displayed
Click on the title of any course	The site should redirect to a page showing the courses information

Search Courses Scenario

Case: Search Courses Functionality

Steps	Expected Result
Login to the application	Login was successful
Click the 'Search Courses' link in the navigation bar under the 'Course Management' header	The Search Courses Page should be displayed with only the year selector being enabled and no courses showing
Select a Year	The year is selected and the semester drop down is now enabled
Press Search	All courses that were offered in the selected year should appear

Steps	Expected Result
Select a semester	The semester is selected and the program code dropdown is now enabled
Press Search	The list of courses is now filtered so that only the courses that were offered in the selected year and selected semester should appear
Select a program code	The program code is selected and the course code is now enabled
Press Search	The list of courses is now filtered so that only the courses of that specific program code (ex. COSC) that were offered in the selected year and selected semester should appear
Click the pencil icon in any of the courses	The site should redirect to the 'Edit Course' page
Click the 'Search Courses' link in the navigation bar under the 'Course Management' header	The Search Courses Page should be displayed with only the year selector being enabled and no courses showing
Click on the title of any course	The site should redirect to a page showing the courses information

Processing Server Test Suite

Submit Test Scenario

Case 1: Correct parameters and data formation: 2 files

Steps	Expected Result
Send postman 'submit' request to server with expected parameters	- Receive 200 response -Receive estimated processing time - Receive email within 2 minutes to posted email address notifying that results are ready

Case 2: Correct parameters and data formation: large number of files (>500)

Steps	Expected Result
Send postman 'submit' request to server with expected parameters	- Receive 200 response -Receive estimated processing time - Receive email within 6 hours to posted email address notifying that results are ready

Case 3: Correct parameters and data formation: 1 file

Steps	Expected Result
Send postman 'submit' request to server with expected parameters	- Receive 400 response

Case4: Erroneous file structure being submitted

Steps	Expected Result
Send postman 'submit' request to server with a file in a tarball that is not one of the following folder formats: “./CurrentYear/###_###_###/file.ext” or “./PreviousYear/###_###_###/file.ext” or “./Exclusions/file.ext”	- Receive 400 response

Case5: Incorrect User ID and license string

Steps	Expected Result
Send postman 'submit' request to server with incorrect User ID and/or license string	- Receive 403 response

Request Test Scenario

Case1: Results are ready

Steps	Expected Result
Send postman 'results' request to server with expected parameters	- Receive 200 response with results

Case2: Incorrect User ID and license string

Steps	Expected Result
Send postman 'results' request to server with incorrect User ID and/or license string	- Receive 403 response

Case3: Results aren't ready yet

Steps	Expected Result
Send postman 'results' request to server with correct User ID and/or license string and JobID	- Receive 403 response

Case4: JobID doesn't exist

Steps	Expected Result
Send postman 'results' request to server with correct User ID and license string but a job ID that doesn't exist	- Receive 403 response

Case5: Results exist but don't belong to user

Steps	Expected Result
Send postman 'results' request to server with incorrect User ID for the given job ID	- Receive 403 response

Case6: Erroneous job ID

Steps	Expected Result
Send postman 'results' request to server with correct User ID and license string but a job ID that doesn't conform to the 128 bit uuid number	- Receive 403 response

Tests to Date and Proposed Testing ===

We are currently utilizing the Travis Continuous Integration tool to automatically run a customized script on every change that is pushed to the master branch of our github repo. The tests currently build the tables defined for the client server.

We aspire to implement test test suite defined above with pytest for python files and protractor and unit tests for C# files. Once these tests are defined, it will be relatively straightforward to add them to the existing travis script and automate our testing. This will ensure that the hook team is made aware within a very short period of time, when a proposed change will break the system and we can fix it before it is merged.