Test rail Integration document with Selenium Java framework

Table of Contents

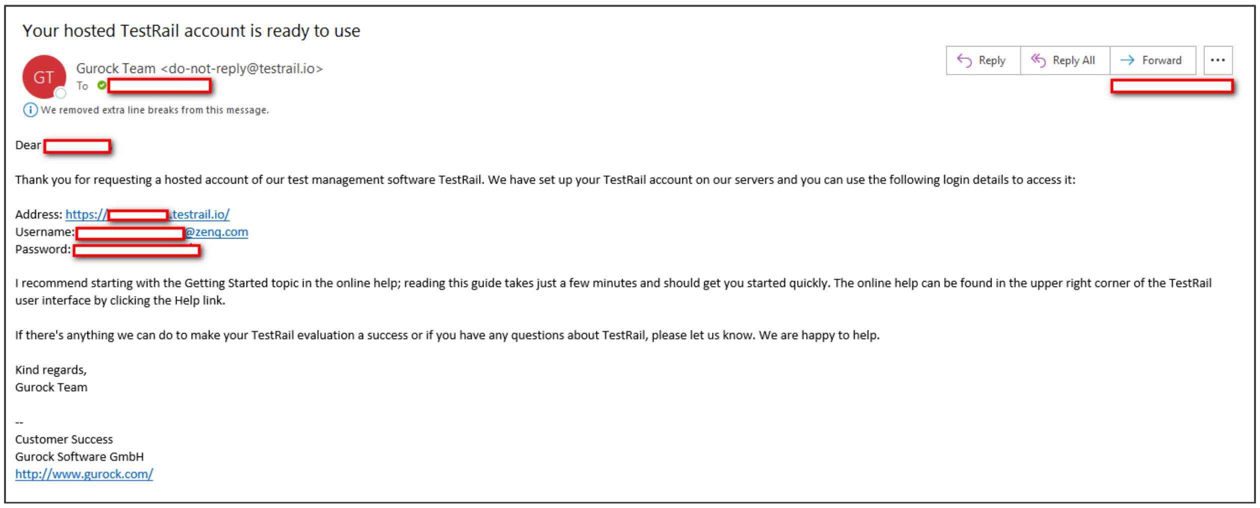
1. Overview

This document provides explanation about how to integrate the results of your selenium tests with corresponding test in Test rail test management tool

2. Preconditions and Steps To Create Project in Test Rail

1. One should create a Testrail account at "https://www.gurock.com/testrail/" (if already exists, please use it)

2. After creating an account the Testrail details are generated and sent to our registered email account as below:



Your hosted TestRail account is ready to use

Gurock Team <do-not-reply@testrail.io>
To [redacted]
[redacted]

We removed extra line breaks from this message.

Dear [redacted]

Thank you for requesting a hosted account of our test management software TestRail. We have set up your TestRail account on our servers and you can use the following login details to access it:

Address: https://[redacted].testrail.io/
Username: [redacted]@zenq.com
Password: [redacted]

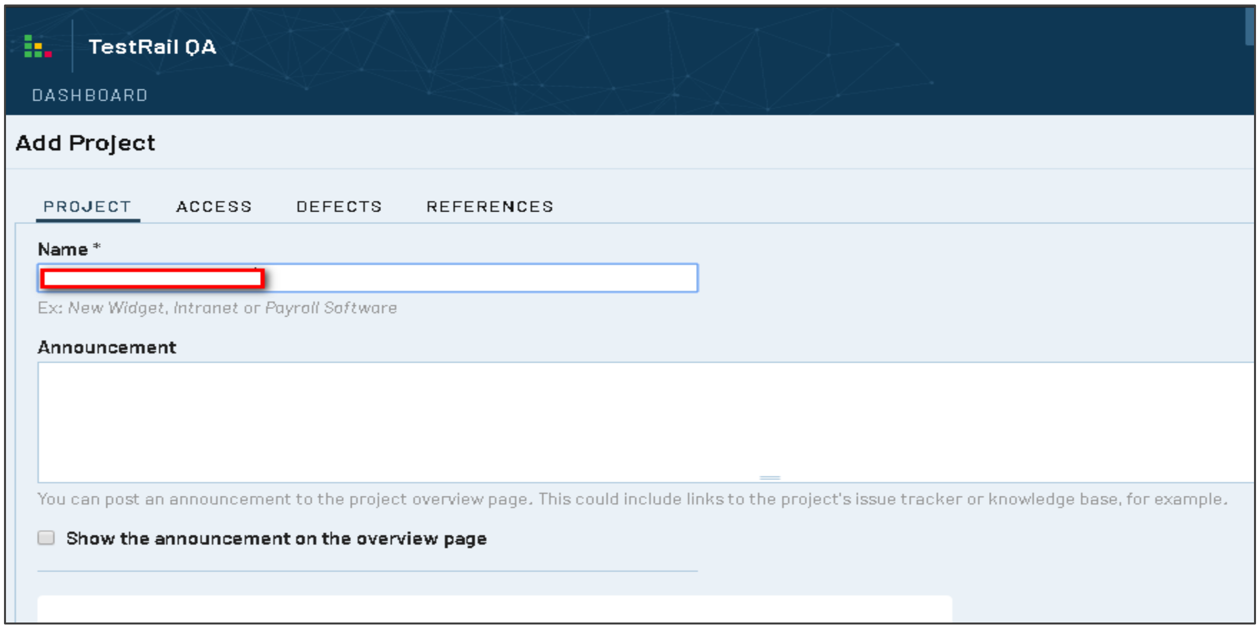
I recommend starting with the Getting Started topic in the online help; reading this guide takes just a few minutes and should get you started quickly. The online help can be found in the upper right corner of the TestRail user interface by clicking the Help link.

If there's anything we can do to make your TestRail evaluation a success or if you have any questions about TestRail, please let us know. We are happy to help.

Kind regards,
Gurock Team

--
Customer Success
Gurock Software GmbH
http://www.gurock.com/

3. Before integrating test cases follow the steps as below:
   - Create a project in Testrail tool



Add Project

Name *

[redacted]

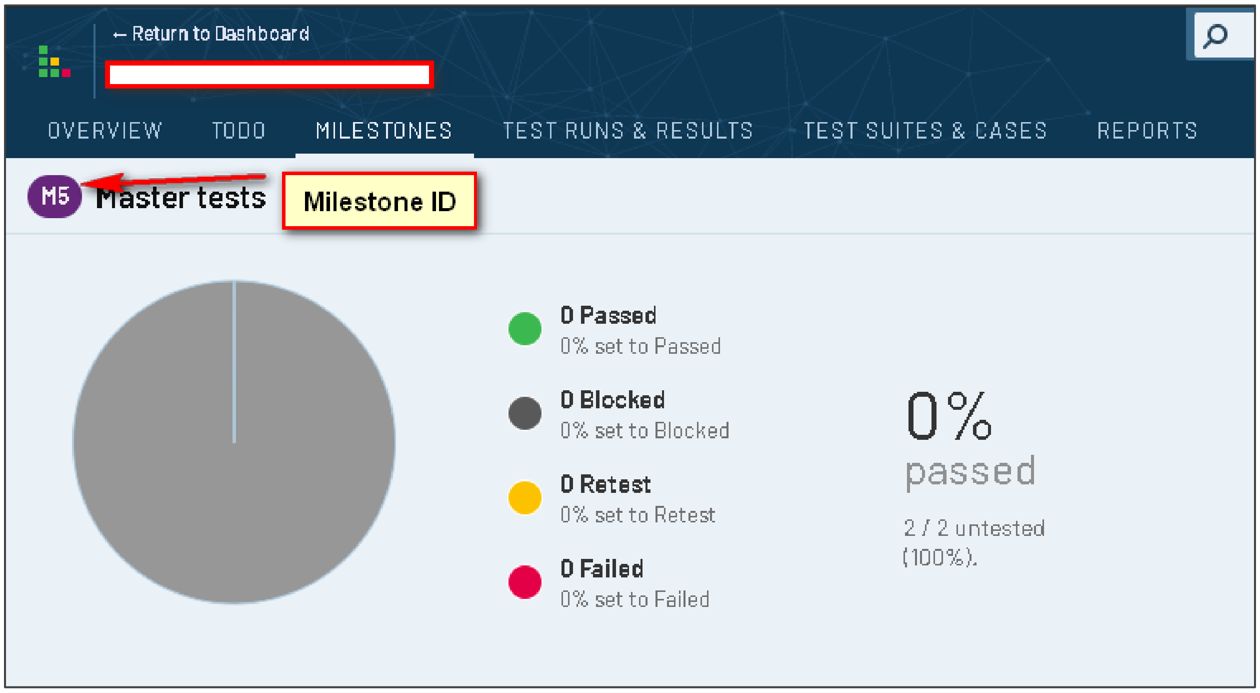
Ex: New Widget, Intranet or Payroll Software

Announcement

You can post an announcement to the project overview page. This could include links to the project's issue tracker or knowledge base, for example.

Show the announcement on the overview page

- Add Milestone and Testrun in the project before adding the test cases



[redacted]

● 0 Passed
0% set to Passed

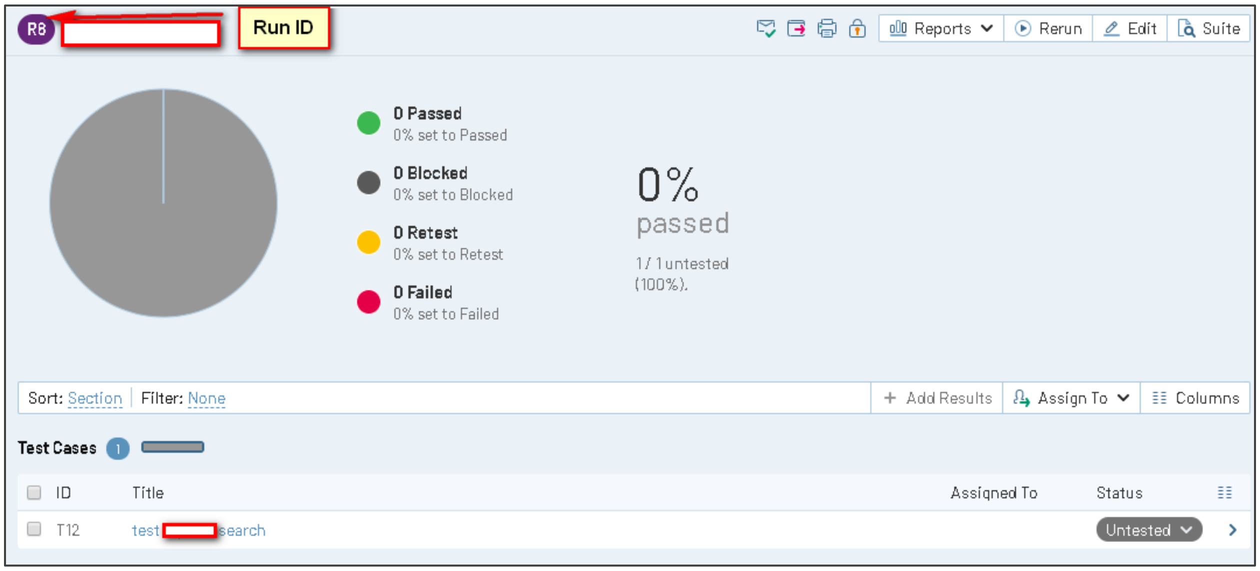
● 0 Blocked
0% set to Blocked

● 0 Retest
0% set to Retest

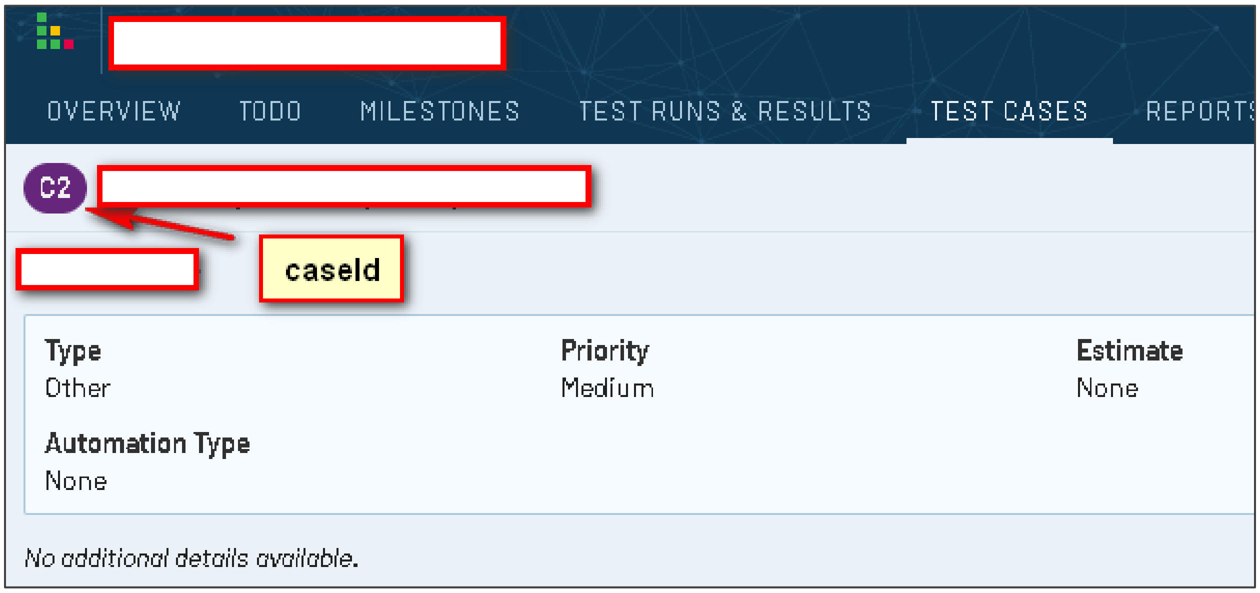
● 0 Failed
0% set to Failed

0%
passed

2 / 2 untested
(100%).



4. Create test case in tool. After Creating Testcase, TestCaseID will be generated as below screenshot-



Type: Other
Priority: Medium
Estimate: None
Automation Type: None

No additional details available.

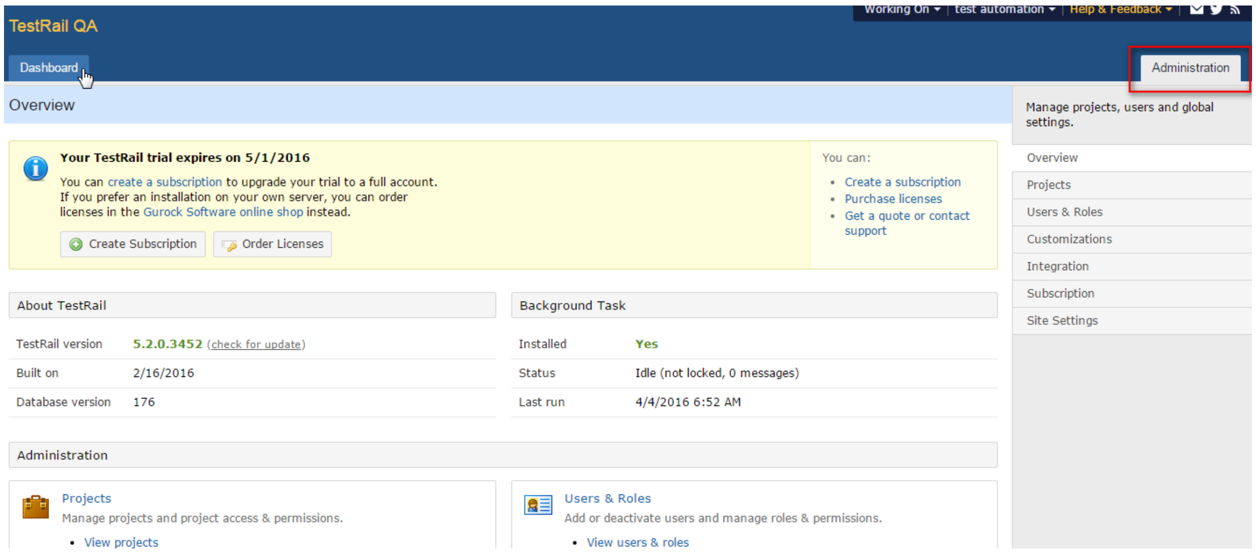
Below changes are needed in selenium framework for updating results in test rail.

3. Enable API in the administration area in Test Rail

User should enable API in the administration area in Test Rail under “Administration > Site Settings > API” before using test rail API in the framework. It can be used to integrate with test automation tools, for UI customizations or for the initial import of projects, test cases, etc.

Please look into below screenshots for more details about how to enable API in Test Rail UI

1. Login into Test Rail account. Click on “Administration” tab in Test rail.



Manage projects, users and global settings.

- Overview
- Projects
- Users & Roles
- Customizations
- Integration
- Subscription
- Site Settings

Your TestRail trial expires on 5/1/2016

You can create a subscription to upgrade your trial to a full account. If you prefer an installation on your own server, you can order licenses in the Gurock Software online shop instead.

Create Subscription | Order Licenses

You can:

- Create a subscription
- Purchase licenses
- Get a quote or contact support

About TestRail

TestRail version	5.2.0.3452 (check for update)
Built on	2/16/2016
Database version	176

Background Task

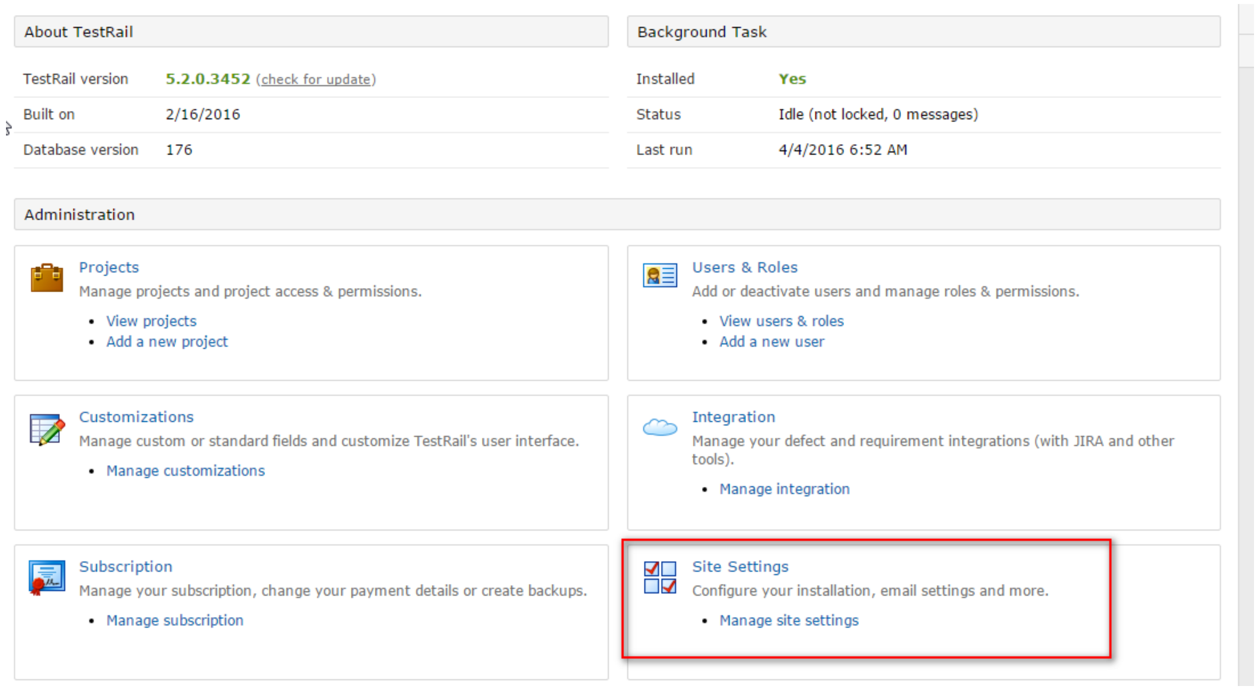
Installed	Yes
Status	Idle (not locked, 0 messages)
Last run	4/4/2016 6:52 AM

Administration

Projects
Manage projects and project access & permissions.
- View projects

Users & Roles
Add or deactivate users and manage roles & permissions.
- View users & roles

2. Click on “Manage site settings” link under “Site Settings”



About TestRail

TestRail version	5.2.0.3452 (check for update)
Built on	2/16/2016
Database version	176

Background Task

Installed	Yes
Status	Idle (not locked, 0 messages)
Last run	4/4/2016 6:52 AM

Administration

Projects
Manage projects and project access & permissions.
- View projects
- Add a new project

Users & Roles
Add or deactivate users and manage roles & permissions.
- View users & roles
- Add a new user

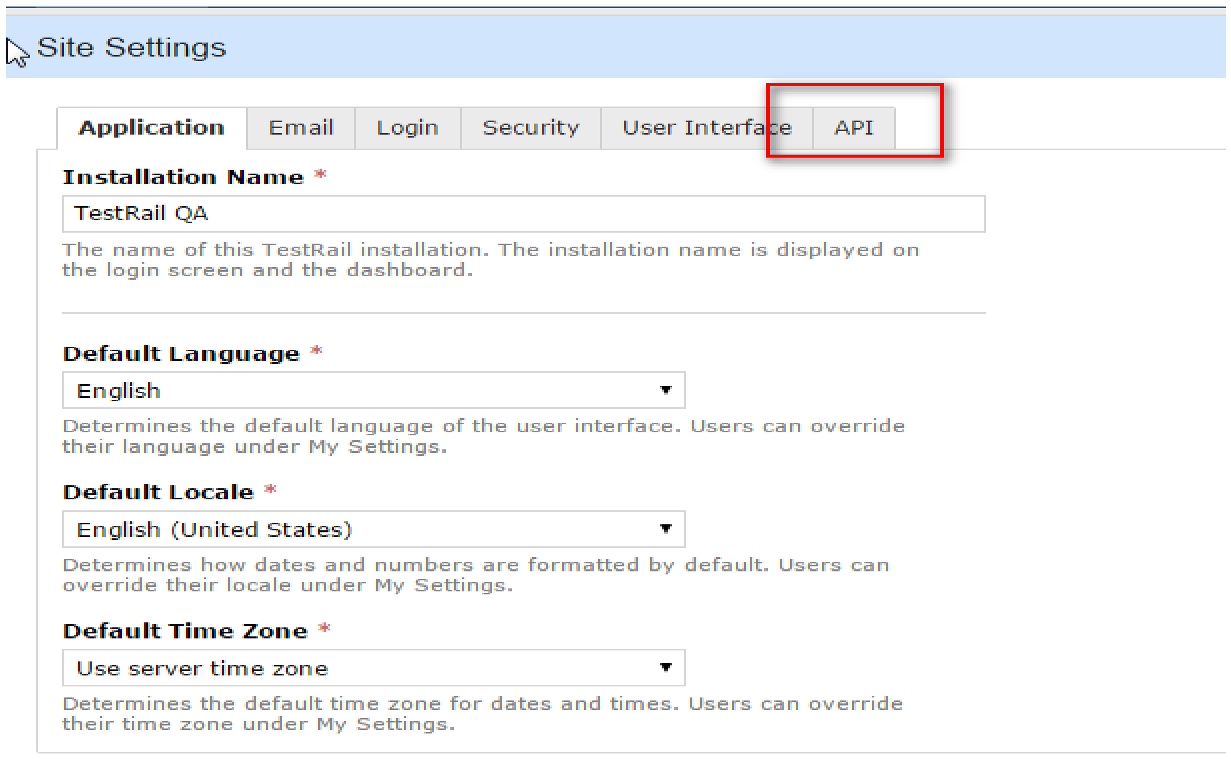
Customizations
Manage custom or standard fields and customize TestRail's user interface.
- Manage customizations

Integration
Manage your defect and requirement integrations (with JIRA and other tools).
- Manage integration

Subscription
Manage your subscription, change your payment details or create backups.
- Manage subscription

Site Settings
Configure your installation, email settings and more.
- Manage site settings

3. Click on “API” tab under “Site Settings” page



Site Settings

Installation Name *

TestRail QA

The name of this TestRail installation. The installation name is displayed on the login screen and the dashboard.

Default Language *

English

Determines the default language of the user interface. Users can override their language under My Settings.

Default Locale *

English (United States)

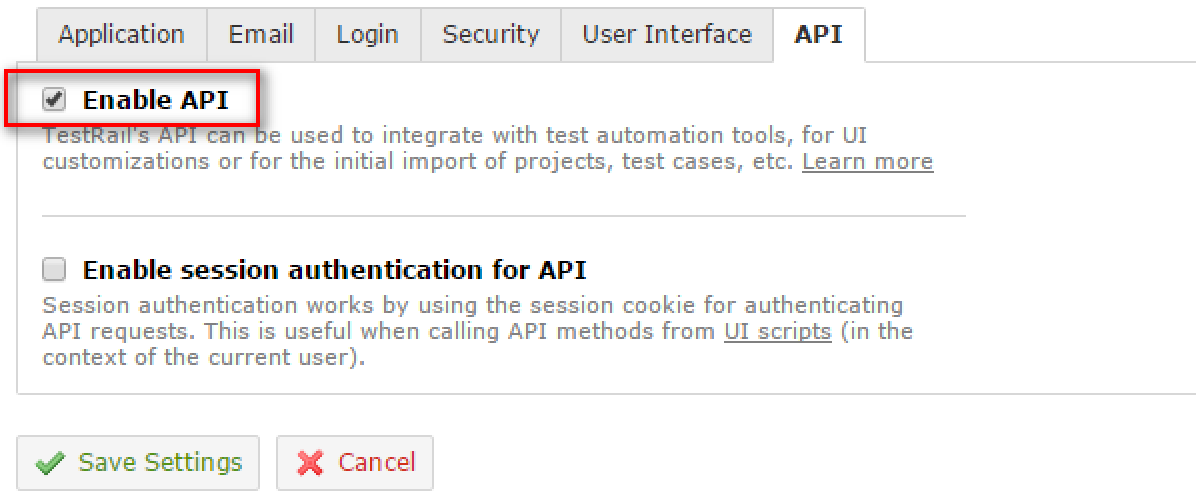
Determines how dates and numbers are formatted by default. Users can override their locale under My Settings.

Default Time Zone *

Use server time zone

Determines the default time zone for dates and times. Users can override their time zone under My Settings.

4. Check the “Enable API” checkbox under “API” tab



[✓] **Enable API**

TestRail's API can be used to integrate with test automation tools, for UI customizations or for the initial import of projects, test cases, etc. Learn more

[ ] **Enable session authentication for API**

Session authentication works by using the session cookie for authenticating API requests. This is useful when calling API methods from UI scripts (in the context of the current user).

4. Changes needed in Sys.properties file

Set appropriate values for below properties in ‘/ConfigFiles/Sys.properties’ file of project.

- ✓ Set ‘true’ if you want to add results to test rail, otherwise set as false.

```
# Set true or false, To update test results to TestRail
UpdateResultsToTestRail = true
```

✓ Set valid test rail account details for below properties.

```
# Testrail account details
# Testrail account URL
TestrailServerUrl = https://[redacted]
# Testrail account user name
TestrailUsername = [redacted]
# Testrail account password
TestrailPassword = [redacted]
```

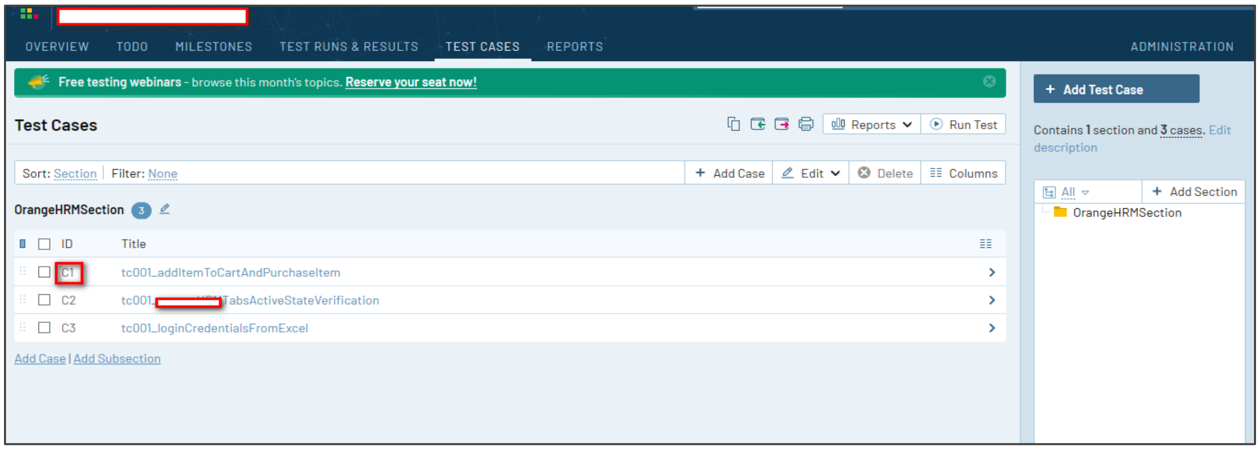
✓ Set test rail configuration details for below properties

```
# Testrail Configuration details
# Name of the project in test rail consists of test suites
ProjectName = [redacted]
# Existed milestone id, if it is blank new milestone will be created
MileStoneID =
# Existed run id, if it is blank new test run will be created
RunID =
```

5. Changes needed in test suite

Add Test rail test case id in "@UseAsTestRailId" annotation above selenium test method name in test suite. Please see below example for more details.

Open test rail account in web browser manually, find test case id for this test method under test cases tab.


[redacted]

OVERVIEW TODO MILESTONES TEST RUNS & RESULTS TEST CASES REPORTS ADMINISTRATION

Free testing webinars - browse this month's topics. Reserve your seat now!

+ Add Test Case

Test Cases

Reports Run Test

Contains 1 section and 3 cases. Edit description

Sort: Section | Filter: None

+ Add Case Edit Delete Columns

All + Add Section

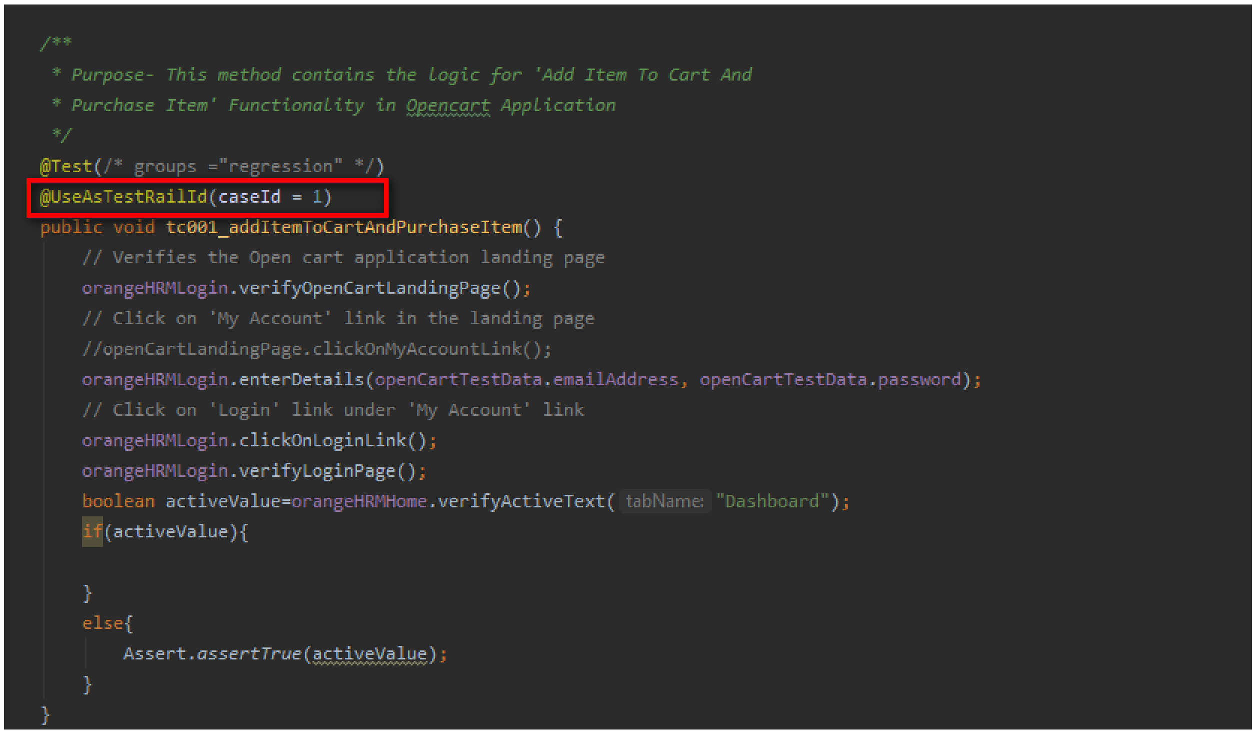
OrangeHRMSection 3

OrangeHRMSection

ID	Title
C1	tc001_addItemToCartAndPurchaseItem
C2	tc001_[redacted]TabsActiveStateVerification
C3	tc001_loginCredentialsFromExcel

Add Case | Add Subsection

Here ‘C1’ is the test case id for test case **tc001_addItemToCartAndPurchaseItem.**You should add ‘caseId = 1’ in “@UseAsTestRailId” annotation above test method in test suite. i.e. **@UseAsTestRailId(caseId = 1).**



```
/**
 * Purpose- This method contains the logic for 'Add Item To Cart And
 * Purchase Item' Functionality in Opencart Application
 */
@Test(/* groups ="regression" */)
@UseAsTestRailId(caseId = 1)
public void tc001_addItemToCartAndPurchaseItem() {
    // Verifies the Open cart application landing page
    orangeHRMLogin.verifyOpenCartLandingPage();
    // Click on 'My Account' link in the landing page
    //openCartLandingPage.clickOnMyAccountLink();
    orangeHRMLogin.enterDetails(openCartTestData.emailAddress, openCartTestData.password);
    // Click on 'Login' link under 'My Account' link
    orangeHRMLogin.clickOnLoginLink();
    orangeHRMLogin.verifyLoginPage();
    boolean activeValue=orangeHRMHome.verifyActiveText( tabName: "Dashboard");
    if(activeValue){

    }
    else{
        Assert.assertTrue(activeValue);
    }
}
```

In the same way you need to add test case id for every test method in your selenium test suite for which you want test result integration with test rail.

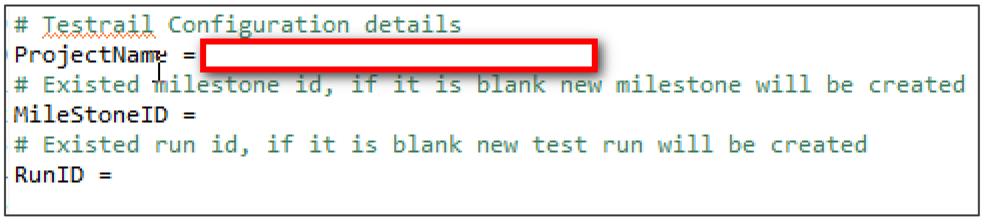
6. Update results to test rail using MileStoneID and RunID

6.1 Update results to test rail with new milestone and test run

In this case **new milestone** will be created with **new test run**. Milestone name created with current date and time. Test run name created with suite name. All existed test cases will be added to this test run. Test results for executed scripts will be updated to test run.

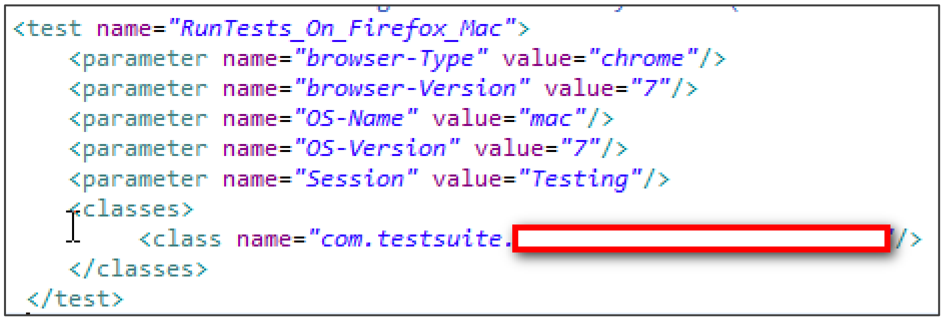
Please see below example for more details.

1. Set MileStoneID and RunID properties values as blank



```
# Testrail Configuration details
ProjectName = [redacted]
# Existed milestone id, if it is blank new milestone will be created
MileStoneID =
# Existed run id, if it is blank new test run will be created
RunID =
```

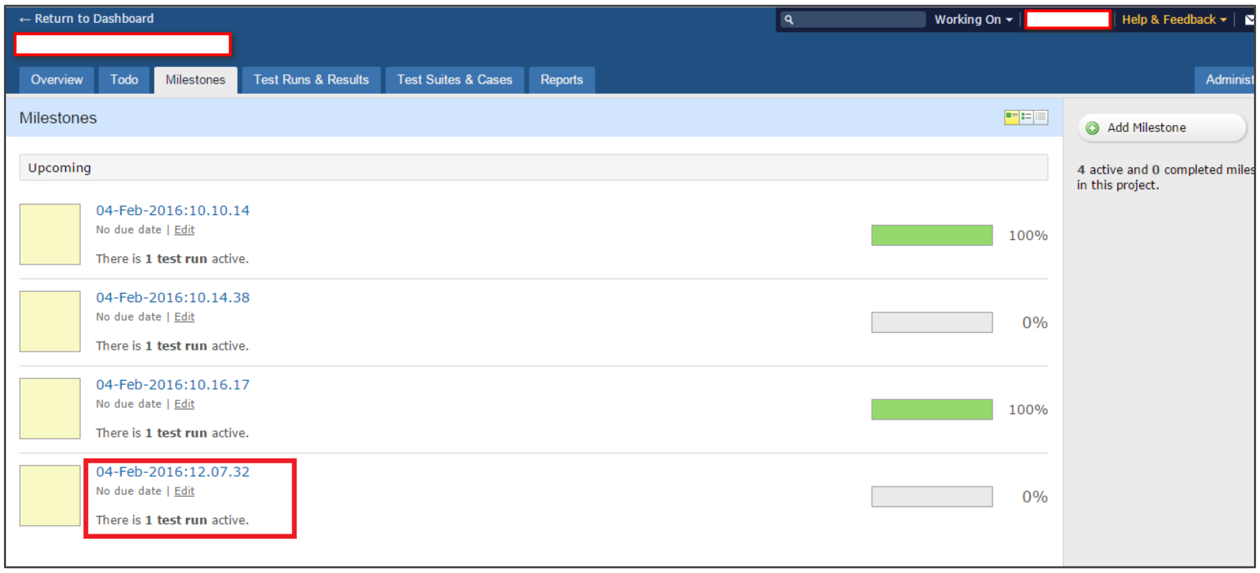
2. Add this test suite to TestNG.xml file.



```
<test name="RunTests_On_Firefox_Mac">
    <parameter name="browser-Type" value="chrome"/>
    <parameter name="browser-Version" value="7"/>
    <parameter name="OS-Name" value="mac"/>
    <parameter name="OS-Version" value="7"/>
    <parameter name="Session" value="Testing"/>
    <classes>
        <class name="com.testsuite.[redacted]"/>
    </classes>
 </test>
```

3. Double click on ‘Run Tests.bat’ file from project folder to kick-off the automated tests setup.

4. New Milestone will be created with new test run before script execution of test suite. Please see below screenshots for more details.



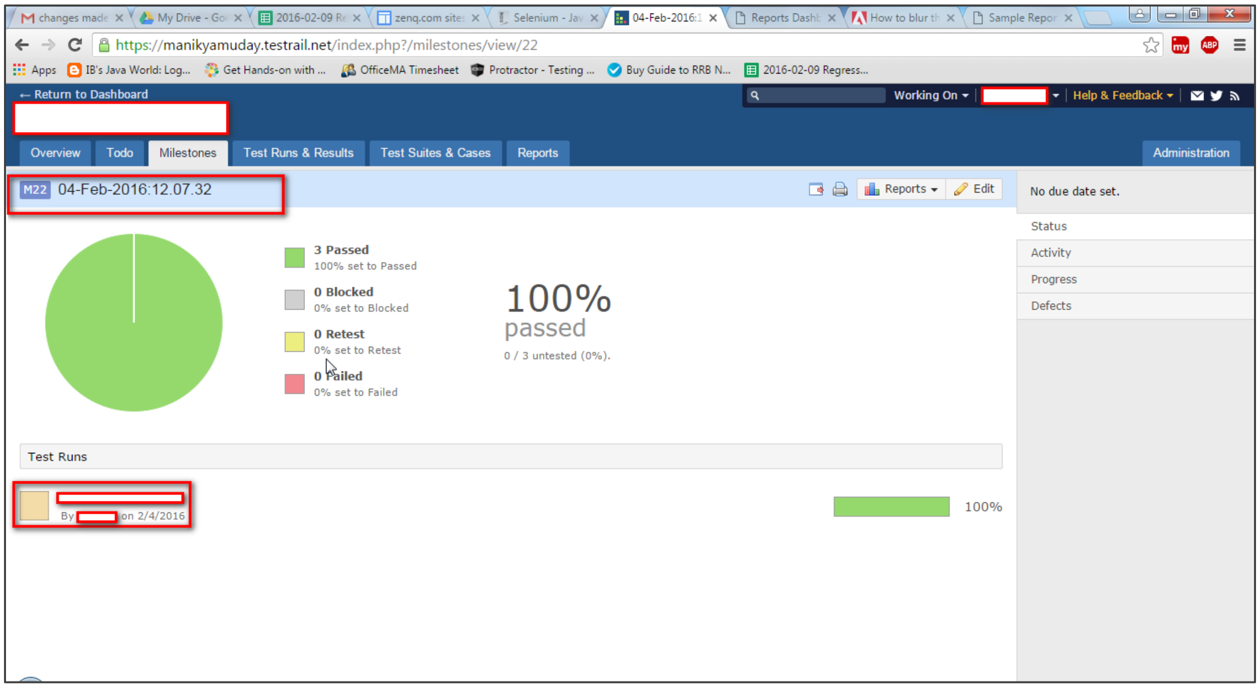
4 active and 0 completed miles in this project.

04-Feb-2016:10.10.14 No due date | Edit There is 1 test run active.	100%
04-Feb-2016:10.14.38 No due date | Edit There is 1 test run active.	0%
04-Feb-2016:10.16.17 No due date | Edit There is 1 test run active.	100%
04-Feb-2016:12.07.32 No due date | Edit There is 1 test run active.	0%

Milestone is created with current date and time as shown in above screenshot.

ii. New milestone created with mile stone id as shown in below screenshot(Mile StoneID will be followed by ‘M’). New test run is added to mile stone.

Under Test Runs testrun is created as below screenshot.



M22 04-Feb-2016:12.07.32

No due date set.

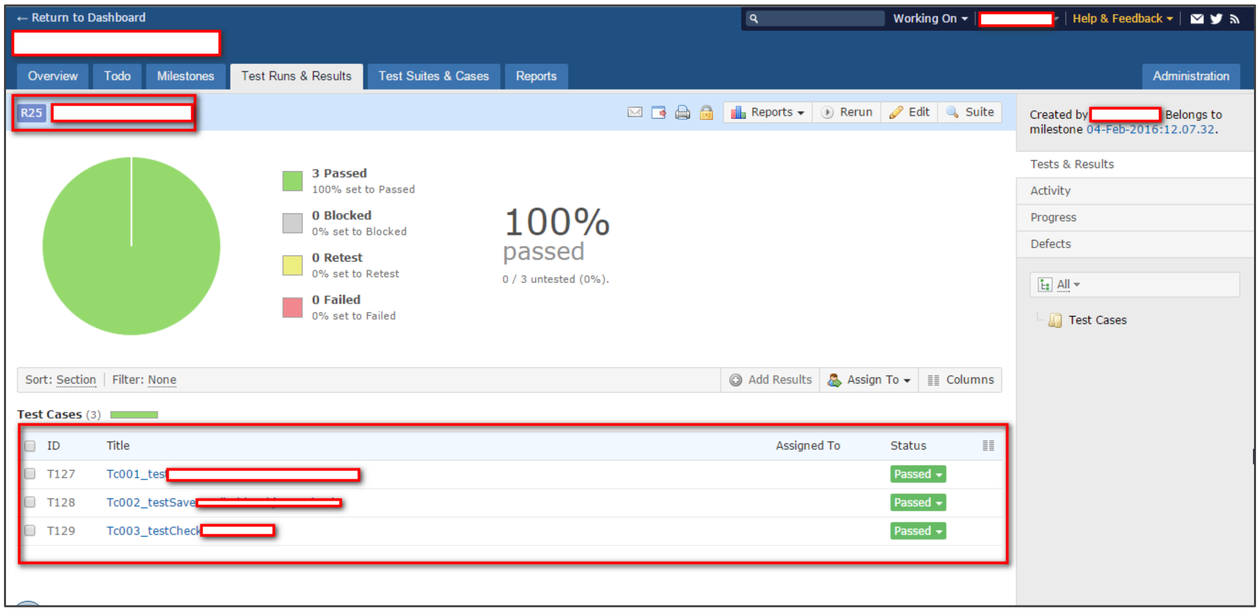
🟩	3 Passed 100% set to Passed
▨	0 Blocked 0% set to Blocked
🟨	0 Retest 0% set to Retest
🟥	0 Failed 0% set to Failed

100% passed
0 / 3 untested (0%).

Test Runs

[redacted]
By [redacted] on 2/4/2016 — 100%

iii. Click on created folder under test run, you can see Run ID as shown in below screenshot (RunID followed by 'R'). After completion of scripts execution, results for all test methods in suite file are added to tests as passed.



R25 [redacted]

Created by [redacted] Belongs to milestone 04-Feb-2016:12.07.32.

🟩	3 Passed 100% set to Passed
▨	0 Blocked 0% set to Blocked
🟨	0 Retest 0% set to Retest
🟥	0 Failed 0% set to Failed

100% passed
0 / 3 untested (0%).

Sort: Section | Filter: None

Test Cases (3)

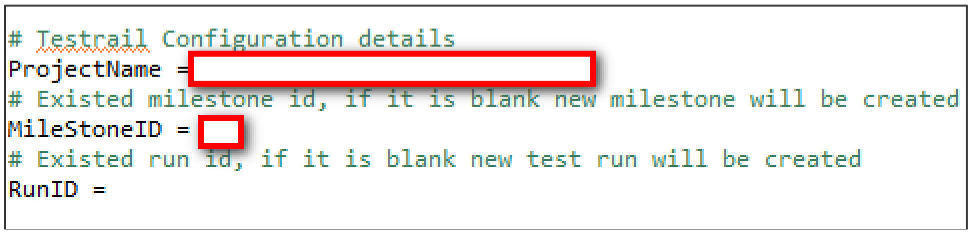
ID	Title	Assigned To	Status
T127	Tc001_tes[redacted]		Passed
T128	Tc002_testSave[redacted]		Passed
T129	Tc003_testCheck[redacted]		Passed

6.2 Update results to test rail with existed milestone and with new test run

In this case **new test run** will be created under **existed milestone**. Test run name created with suite name. All existed test cases will be added to this test run. Test results for executed scripts will be updated to test run.

Please see below example for more details.

1. Set existed milestone id for MileStoneID and blank for RunID in '/ConfigFiles/Sys.properties' file.

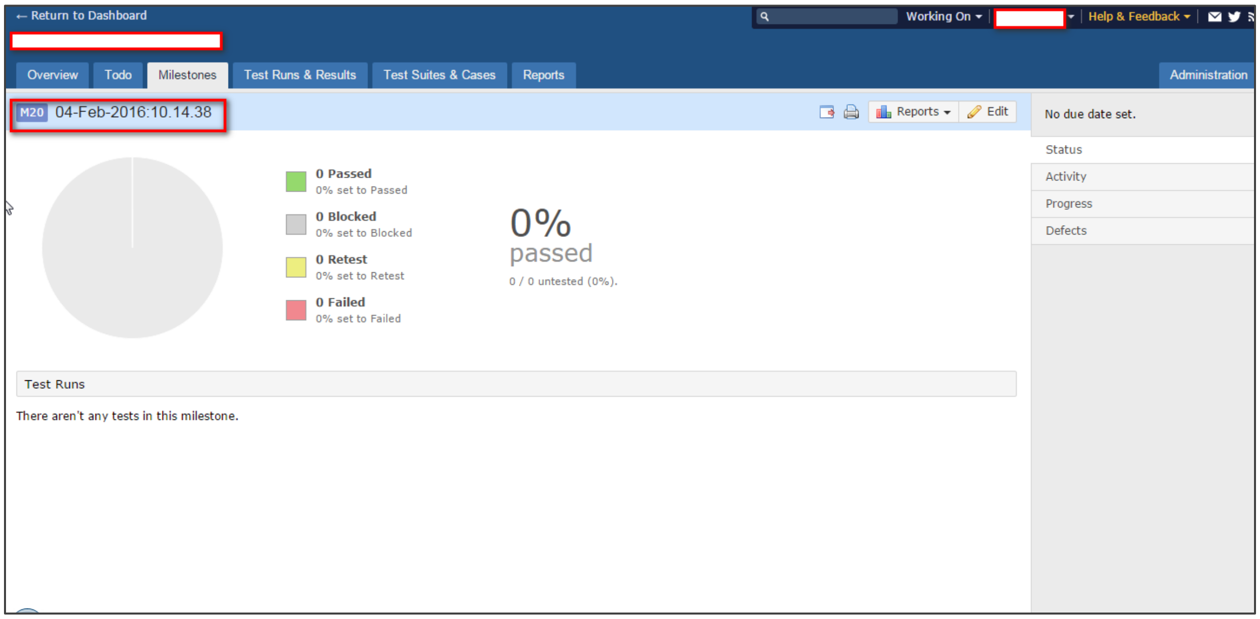


```
# Testrail Configuration details
ProjectName = [redacted]
# Existed milestone id, if it is blank new milestone will be created
MileStoneID = [redacted]
# Existed run id, if it is blank new test run will be created
RunID =
```

Repeat 2, 3 from section 5.1

2. New test run will be created under Milestone id (MileStoneId as given in step 1 of section 6.2) before execution of test suite. Please see below screenshots for more details.

i. See existed milestone with id (MileStoneId as given in step 1 of section 6.2). No test runs are available in this milestone.



[redacted]

[redacted]

M20 04-Feb-2016:10.14.38

No due date set.

0 Passed
0% set to Passed

0 Blocked
0% set to Blocked

0 Retest
0% set to Retest

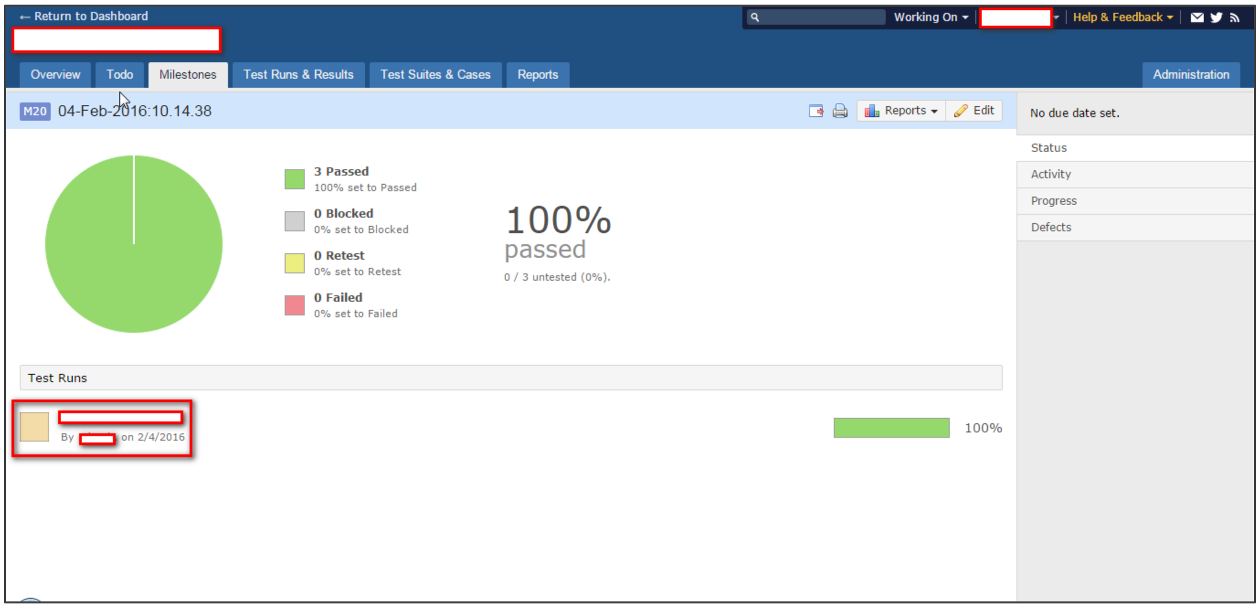
0 Failed
0% set to Failed

0% passed
0 / 0 untested (0%).

Test Runs

There aren't any tests in this milestone.

ii. After Execution is completed, Go to test rail account and click on milestone id (MileStoneId as given in step 1 of section 6.2). You can see new test run has been created with Executed className



[redacted]

M20 04-Feb-2016:10.14.38

No due date set.

3 Passed
100% set to Passed

0 Blocked
0% set to Blocked

0 Retest
0% set to Retest

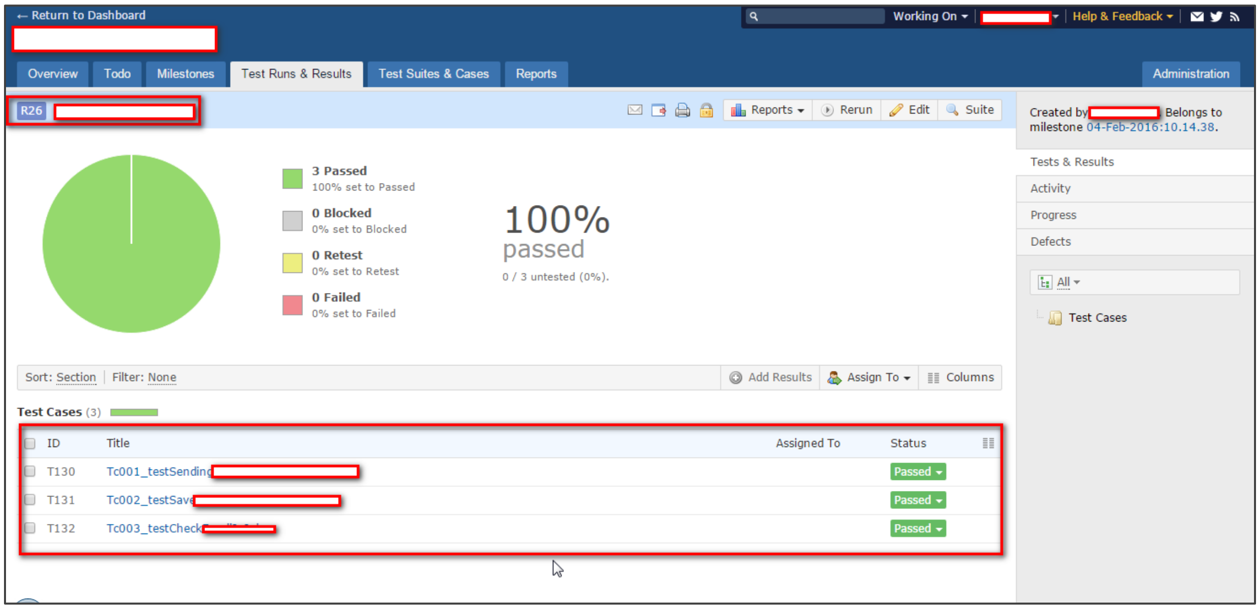
0 Failed
0% set to Failed

100%
passed
0 / 3 untested (0%).

Test Runs

[redacted]
By [redacted] on 2/4/2016 100%

iii. Click on new test run created. Here is the new test run id (RunID followed by R). Observe that results for all test methods in suite file are added to newly created test run i.e. passed



[redacted]

R26 [redacted]

Created by [redacted] Belongs to milestone 04-Feb-2016:10.14.38.

3 Passed
100% set to Passed

0 Blocked
0% set to Blocked

0 Retest
0% set to Retest

0 Failed
0% set to Failed

100%
passed
0 / 3 untested (0%).

Sort: Section | Filter: None

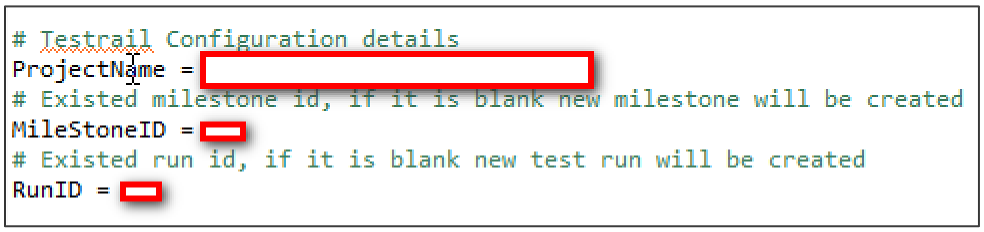
Test Cases (3)

ID	Title	Assigned To	Status
T130	Tc001_testSending[redacted]		Passed
T131	Tc002_testSave[redacted]		Passed
T132	Tc003_testCheck[redacted]		Passed

6.3 Update results to test rail with existed milestone and test run

In this case neither milestone nor test run will be created. Adding Test results for executed scripts will be updated to test run using **existed milestone and test run**.

1. Set existed milestone id for MileStoneID and RunID in '/ConfigFiles/Sys.properties' file.

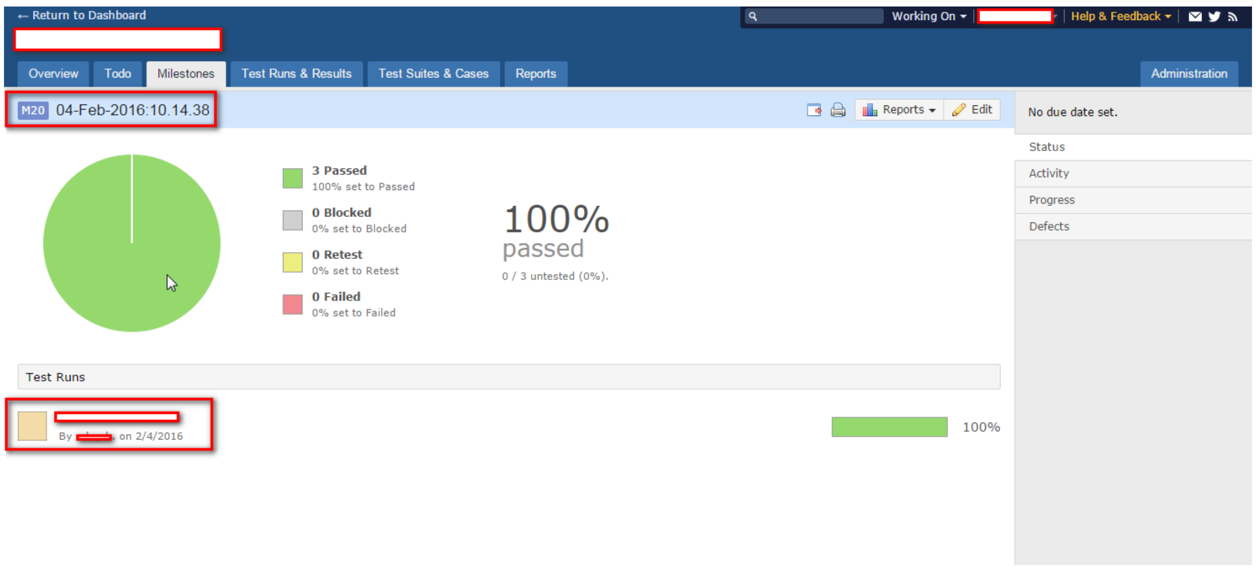


```
# Testrail Configuration details
ProjectName = [redacted]
# Existed milestone id, if it is blank new milestone will be created
MileStoneID = [redacted]
# Existed run id, if it is blank new test run will be created
RunID = [redacted]
```

Repeat 2, 3 from section 5.1

2. Using existed milestone and test runs for adding results into testrail.

i. See existed milestone with id (MileStoneID given in sys.properties) and with existed test run



M20 04-Feb-2016:10.14.38

No due date set.

🟩 3 Passed
100% set to Passed

▨ 0 Blocked
0% set to Blocked

🟨 0 Retest
0% set to Retest

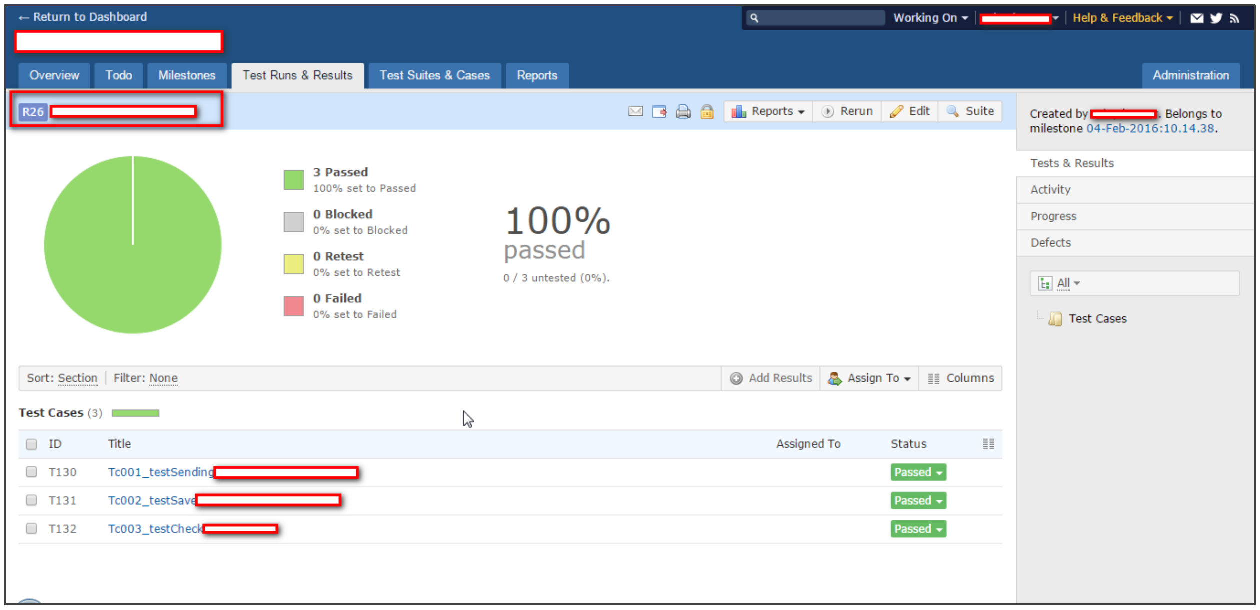
🟥 0 Failed
0% set to Failed

100% passed
0 / 3 untested (0%).

Test Runs

[redacted]
By [redacted] on 2/4/2016 — 100%

ii. Click on existed test run (Available under Test Runs). Here in the existing test run id, observe that results for all test methods in suite file are added to existed test run i.e. passed



[redacted]

R26 [redacted]

Created by [redacted]. Belongs to milestone 04-Feb-2016:10.14.38.

🟩	3 Passed 100% set to Passed
▨	0 Blocked 0% set to Blocked
🟨	0 Retest 0% set to Retest
🟥	0 Failed 0% set to Failed

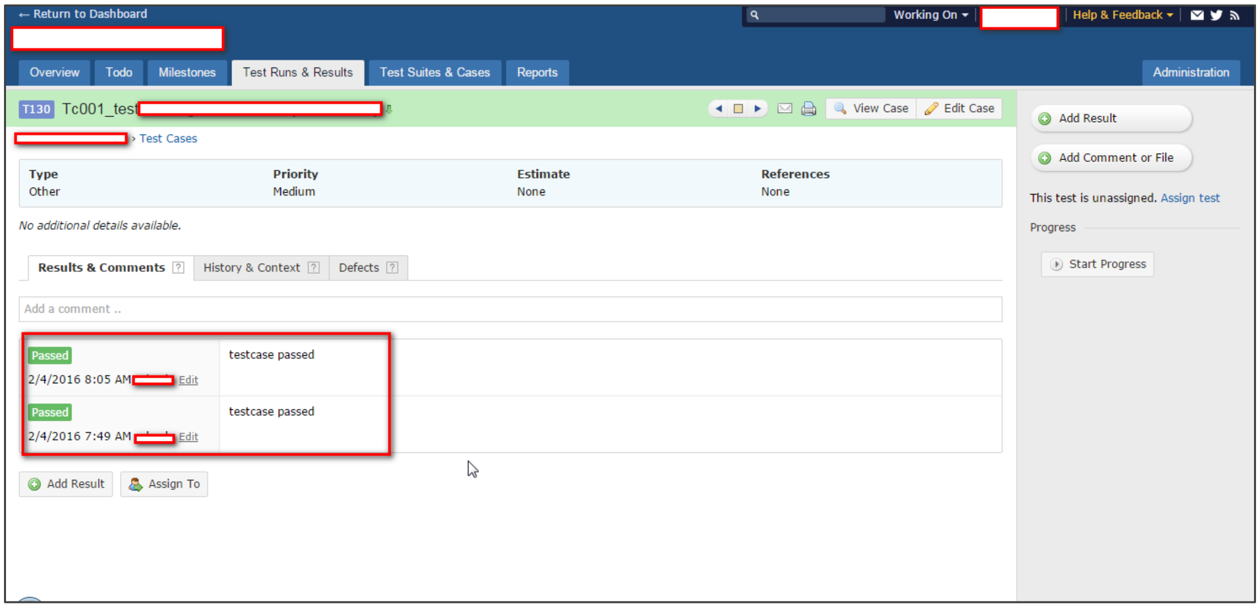
100%
passed
0 / 3 untested (0%).

Sort: Section | Filter: None

Test Cases (3)

ID	Title	Assigned To	Status
T130	Tc001_testSending[redacted]		Passed ▾
T131	Tc002_testSave[redacted]		Passed ▾
T132	Tc003_testCheck[redacted]		Passed ▾

Here results are updated and displayed as passed and override the previous results. If you want to see previous test run results, click on individual test case



[redacted]

T130 Tc001_test[redacted]

[redacted] › Test Cases

Type	Priority	Estimate	References
Other	Medium	None	None

No additional details available.

This test is unassigned. Assign test

Progress

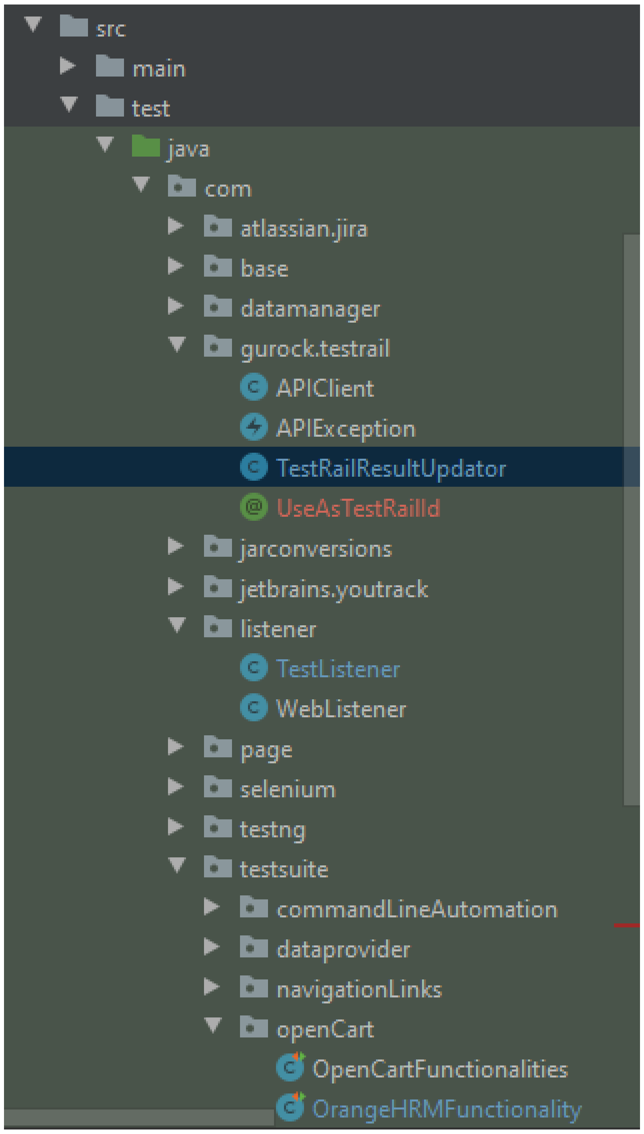
Passed 2/4/2016 8:05 AM [redacted] Edit	testcase passed
Passed 2/4/2016 7:49 AM [redacted] Edit	testcase passed

7. Code for customization of test rail API methods and integration of test results to test rail

This section provides information about location of test rail API code, implementation of customized methods using test rail API and integration code for updating test results into test rail in selenium java framework.

7.1 Implementation code for customizing test rail API methods

Test rail API code exists inside package '**com.gurock.testrail**' and consists of files shown in below screenshot.



```
src
  main
  test
    java
      com
        atlassian.jira
        base
        datamanager
        gurock.testrail
          APIClient
          APIException
          TestRailResultUpdator
          UseAsTestRailId
        jarconversions
        jetbrains.youtrack
        listener
          TestListener
          WebListener
        page
        selenium
        testng
        testsuite
          commandLineAutomation
          dataprovider
          navigationLinks
          openCart
            OpenCartFunctionalities
            OrangeHRMFunctionality
```

1. ApIClient.Java

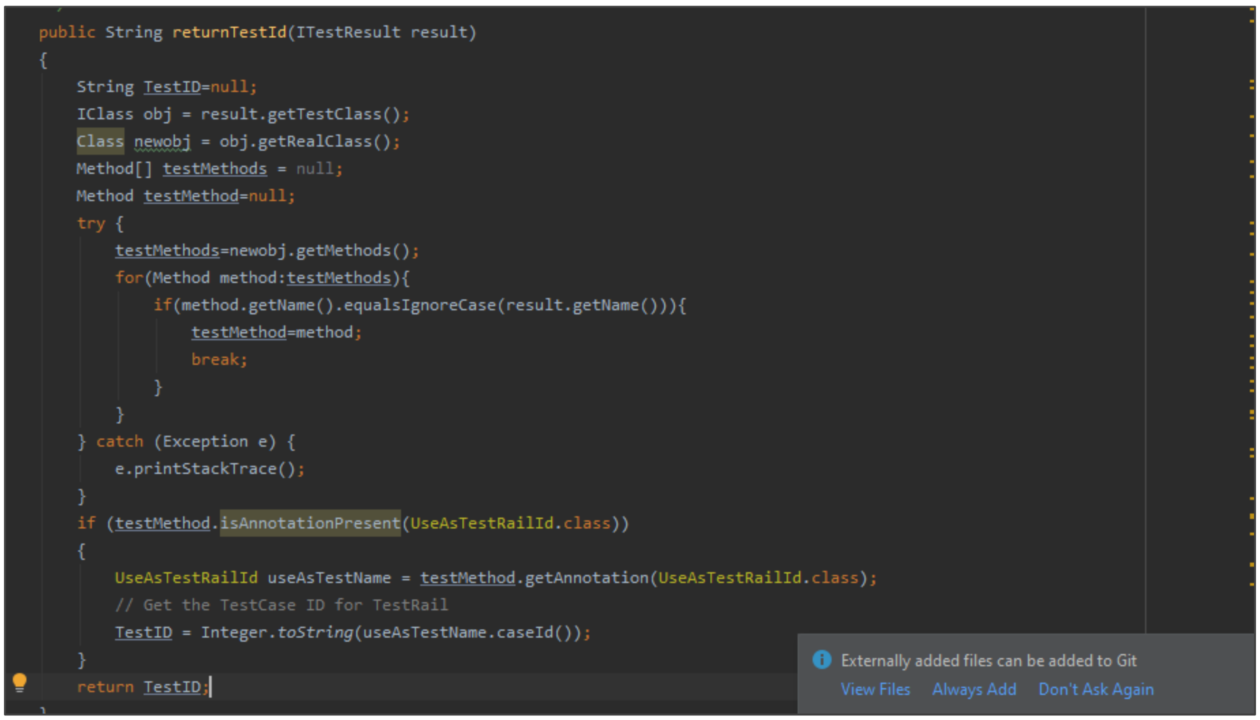
This file contains code of Test Rail API binding for Java. It provides the basic functionality to authenticate API requests, provides seamless JSON and UTF-8 encoding/decoding and has generic support for read and write requests. Download the API from this link http://docs.gurock.com/testrail-api2/start

2. APIException.Java

This file contains code for exception of type TestRailAPIException may be thrown in case of an error (connection issues, failed authentication, missing API arguments, etc.) and your code should be prepared to handle this.

3. TestRailResultUpdator.Java

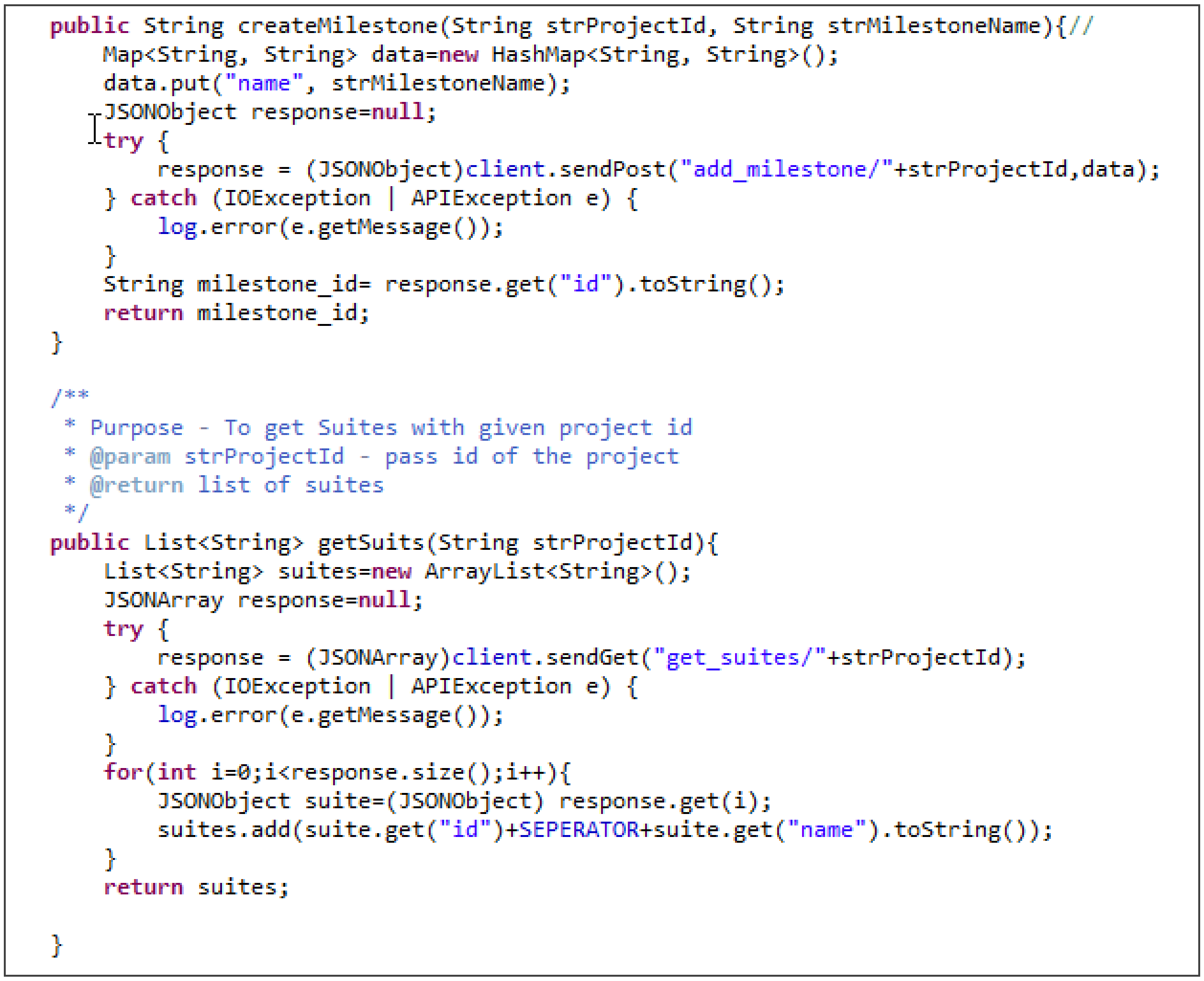
This file contains implementation of customized code i.e. to get testcaseID from test case name, get projects, get cases, create milestones, add test runs, and add test results to test rail using test rail API method. Please see some of the implemented methods shown in below screenshots.



```
public String returnTestId(ITestResult result)
{
    String TestID=null;
    IClass obj = result.getTestClass();
    Class newobj = obj.getRealClass();
    Method[] testMethods = null;
    Method testMethod=null;
    try {
        testMethods=newobj.getMethods();
        for(Method method:testMethods){
            if(method.getName().equalsIgnoreCase(result.getName())){
                testMethod=method;
                break;
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    if (testMethod.isAnnotationPresent(UseAsTestRailId.class))
    {
        UseAsTestRailId useAsTestName = testMethod.getAnnotation(UseAsTestRailId.class);
        // Get the TestCase ID for TestRail
        TestID = Integer.toString(useAsTestName.caseId());
    }
    return TestID;
```

Externally added files can be added to Git

```
public void addResultToTestRail(int status, int case_id)
{
    Map<String, Object> data=new HashMap<String, Object>();
    //1=Passed, 5-Failed, 6-Skipped
    if(status==1){
        data.put("status_id", 1);
        data.put("comment", "testcase passed");
    }else if(status==2){
        data.put("status_id", 2);
        data.put("comment", "testcase blocked");
    }else if(status==5){
        data.put("status_id", 5);
        data.put("comment", "testcase failed");
    }else if(status==6){
        data.put("status_id", 6);
        data.put("comment", "testcase skipped");
    }
    try {
        boolean flag = false;
        JSONObject response;
        for(String runID : runIds){
            if(getCaseIdsForRun(runID).contains(String.valueOf(case_id))){
                response = (JSONObject)client.sendPost("add_result_for_case/"+runID+"/"+case_id, data);
                if(response.get("id")!=null){
                    log.info("Result Added Successfully into TestRail for case ID : "+case_id);
                    flag = true;
                }
                if(!flag){
                    log.error("Failed to add the result into TestRail");
                }
            }
        }
    }
    catch (IOException | APIException e) {
        log.error("Failed to add result to testrail, Here is the issue: " + e.getMessage());
    }
}
```

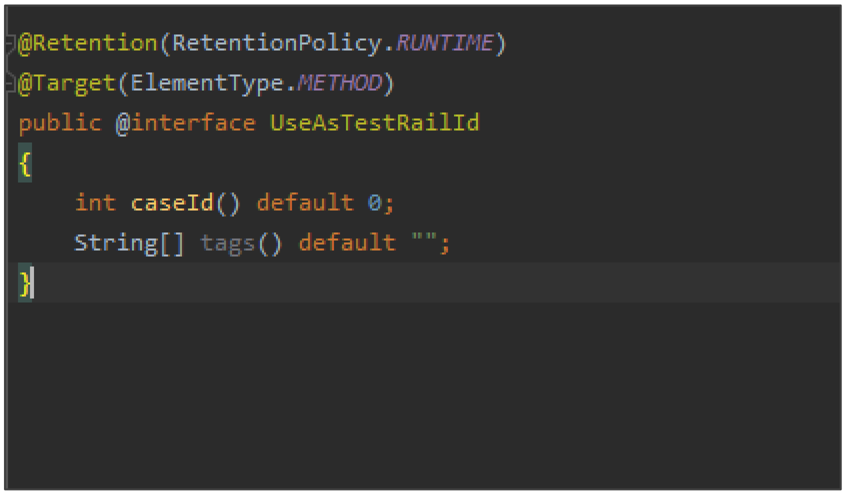


```
public String createMilestone(String strProjectId, String strMilestoneName){//
    Map<String, String> data=new HashMap<String, String>();
    data.put("name", strMilestoneName);
    JSONObject response=null;
    try {
        response = (JSONObject)client.sendPost("add_milestone/"+strProjectId,data);
    } catch (IOException | APIException e) {
        log.error(e.getMessage());
    }
    String milestone_id= response.get("id").toString();
    return milestone_id;
}

/**
 * Purpose - To get Suites with given project id
 * @param strProjectId - pass id of the project
 * @return list of suites
 */
public List<String> getSuits(String strProjectId){
    List<String> suites=new ArrayList<String>();
    JSONArray response=null;
    try {
        response = (JSONArray)client.sendGet("get_suites/"+strProjectId);
    } catch (IOException | APIException e) {
        log.error(e.getMessage());
    }
    for(int i=0;i<response.size();i++){
        JSONObject suite=(JSONObject) response.get(i);
        suites.add(suite.get("id")+SEPERATOR+suite.get("name").toString());
    }
    return suites;

}
```

Note: This file is useful for any future enhancements like e.g. delete milestones through code.

UseAsTestRailId.Java:

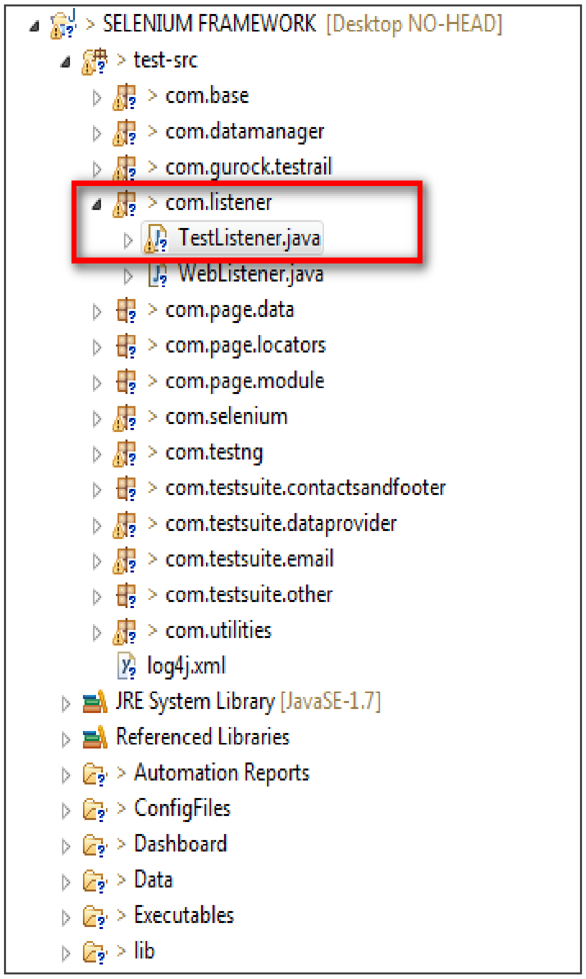
This file creates annotation which can be used above Test case and which accepts the TestCaseId as parameter.



```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface UseAsTestRailId
{
    int caseId() default 0;
    String[] tags() default "";
}
```

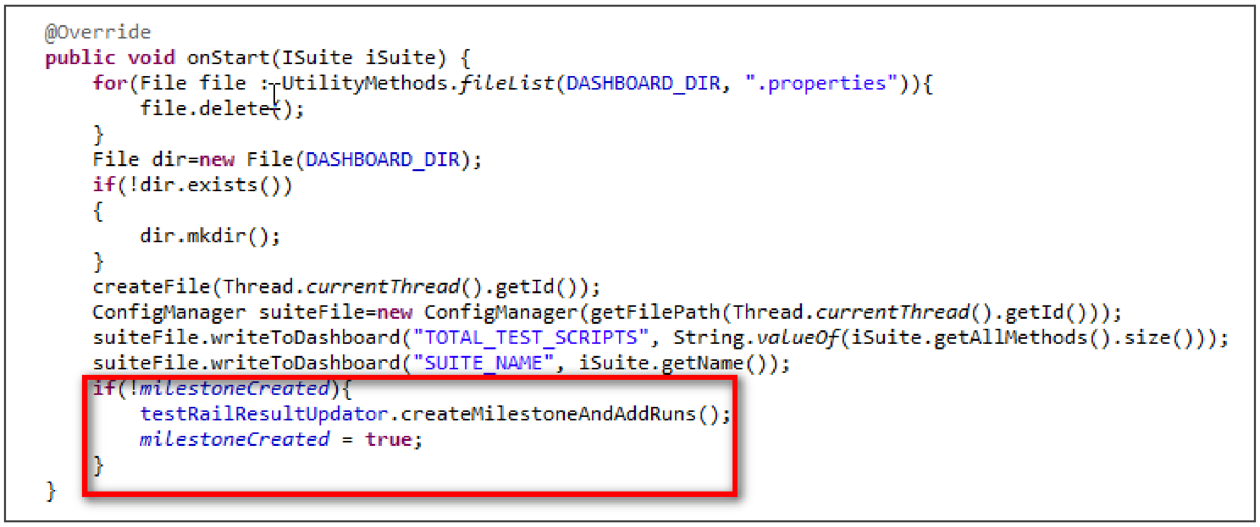
7.2 Integration code for updating test results to test rail

Integration code for updating test results to test rail is implemented in ‘**TestListener.java’** file. Please see location of file is shown in below screenshot.



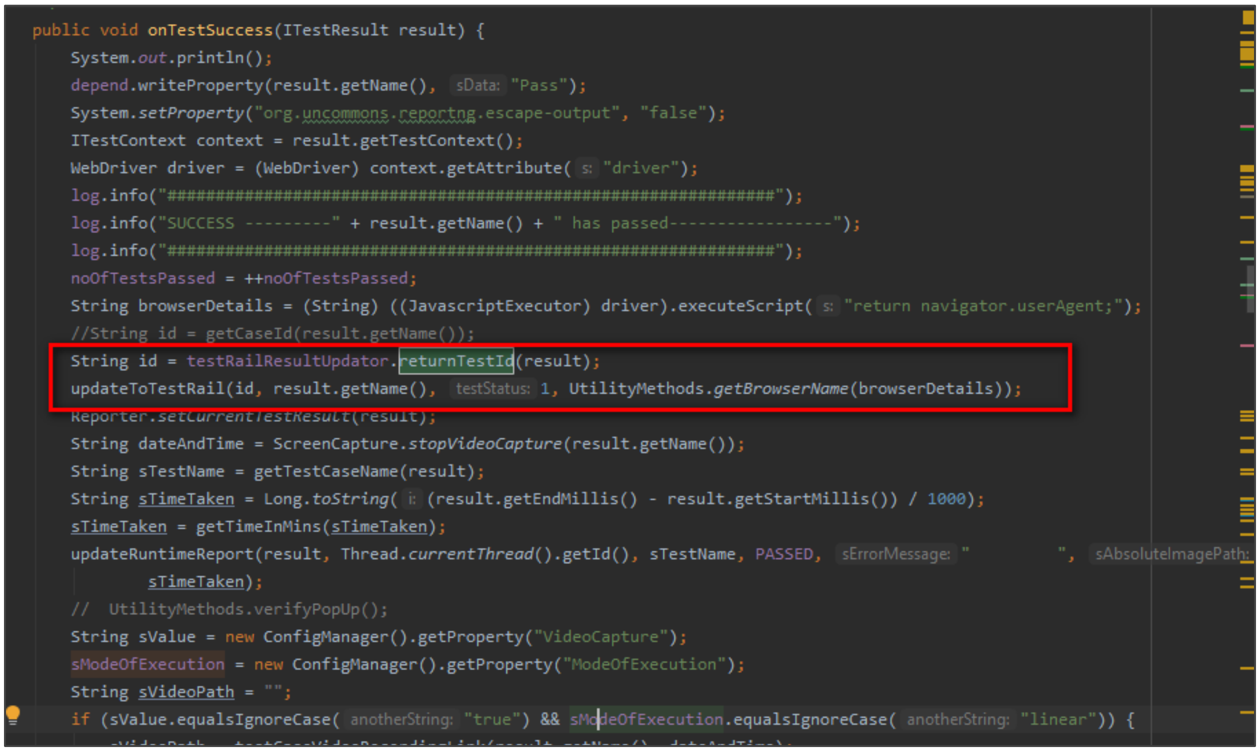
```
> SELENIUM FRAMEWORK [Desktop NO-HEAD]
  > test-src
    > com.base
    > com.datamanager
    > com.gurock.testrail
    > com.listener
        TestListener.java
        WebListener.java
    > com.page.data
    > com.page.locators
    > com.page.module
    > com.selenium
    > com.testng
    > com.testsuite.contactsandfooter
    > com.testsuite.dataprovider
    > com.testsuite.email
    > com.testsuite.other
    > com.utilities
      log4j.xml
  JRE System Library [JavaSE-1.7]
  Referenced Libraries
  > Automation Reports
  > ConfigFiles
  > Dashboard
  > Data
  > Executables
  > lib
```

Creating milestones and runs before starting execution of test method. This is implemented in **onStart()** method. Please see below screenshot for more detail.



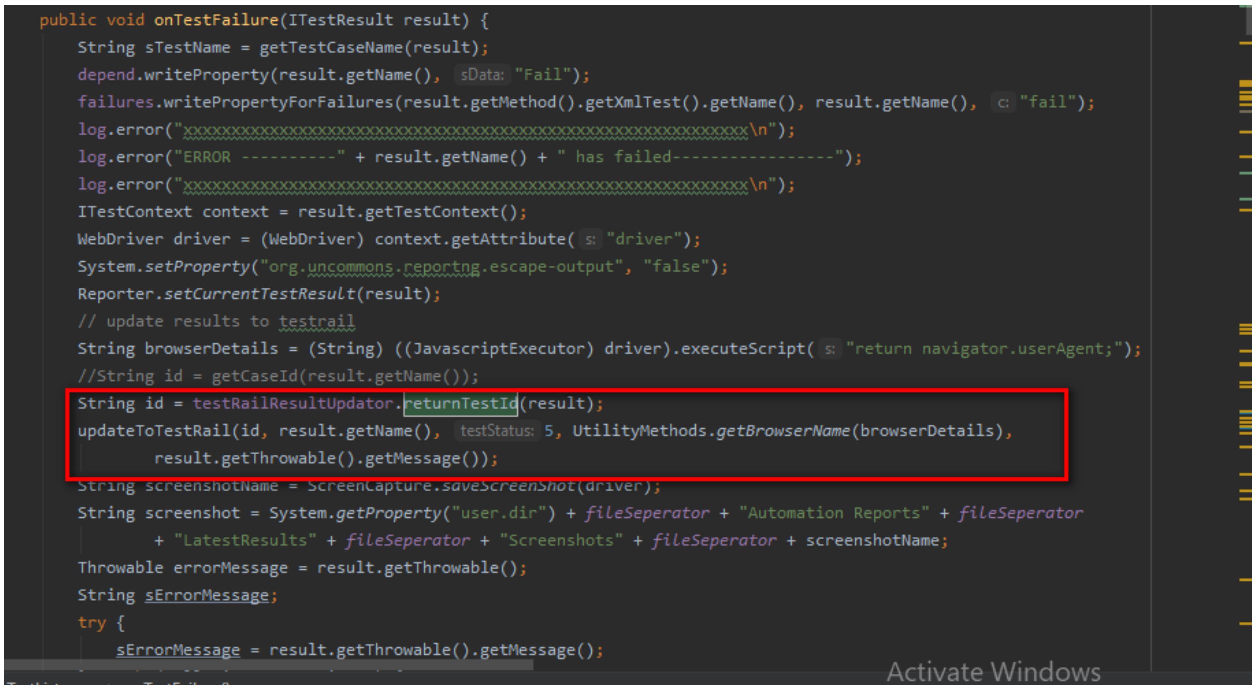
```java
@Override
public void onStart(ISuite iSuite) {
    for(File file : UtilityMethods.fileList(DASHBOARD_DIR, ".properties")){
        file.delete();
    }
    File dir=new File(DASHBOARD_DIR);
    if(!dir.exists())
    {
        dir.mkdir();
    }
    createFile(Thread.currentThread().getId());
    ConfigManager suiteFile=new ConfigManager(getFilePath(Thread.currentThread().getId()));
    suiteFile.writeToDashboard("TOTAL_TEST_SCRIPTS", String.valueOf(iSuite.getAllMethods().size()));
    suiteFile.writeToDashboard("SUITE_NAME", iSuite.getName());
    if(!milestoneCreated){
        testRailResultUpdator.createMilestoneAndAddRuns();
        milestoneCreated = true;
    }
}
```

After test method executed, if test is passed **onTestSuccess ()** method will be called and result will be updated as passed to test rail for this test case by getting testcaseId.



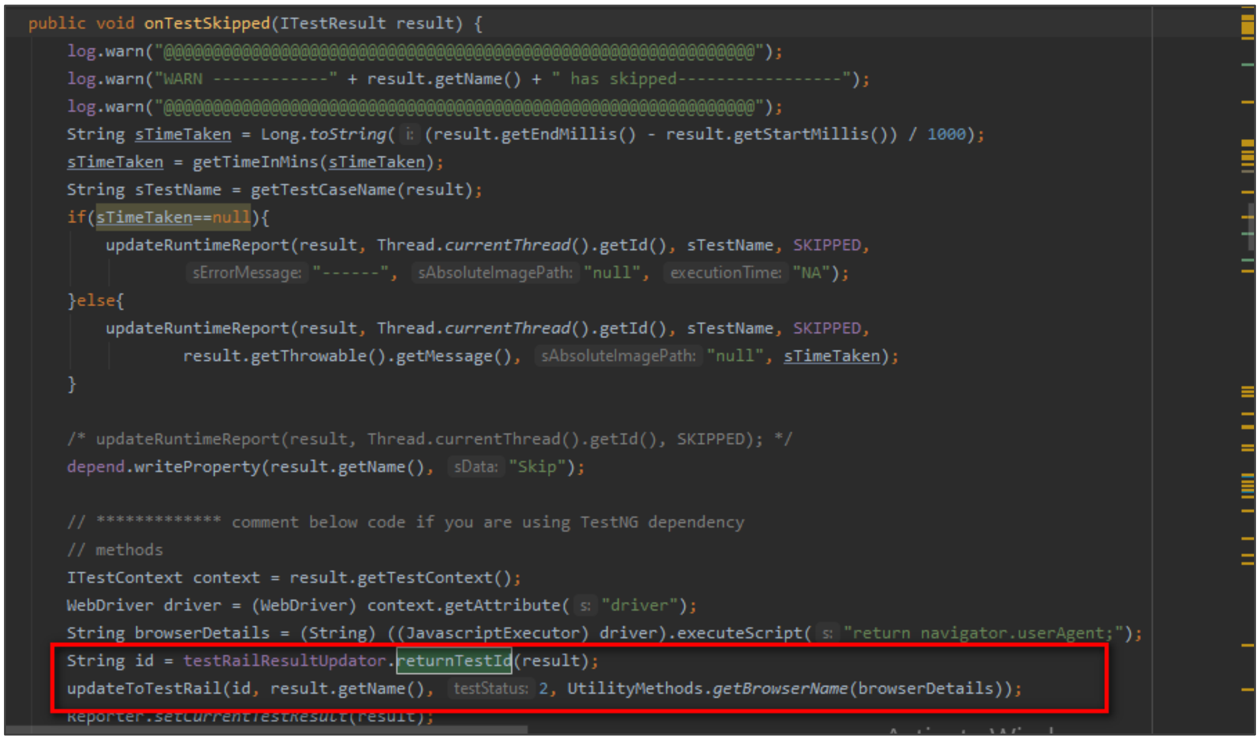
```java
public void onTestSuccess(ITestResult result) {
    System.out.println();
    depend.writeProperty(result.getName(),  sData: "Pass");
    System.setProperty("org.uncommons.reportng.escape-output", "false");
    ITestContext context = result.getTestContext();
    WebDriver driver = (WebDriver) context.getAttribute( s: "driver");
    log.info("##############################################################");
    log.info("SUCCESS ---------" + result.getName() + " has passed----------------");
    log.info("##############################################################");
    noOfTestsPassed = ++noOfTestsPassed;
    String browserDetails = (String) ((JavascriptExecutor) driver).executeScript( s: "return navigator.userAgent;");
    //String id = getCaseId(result.getName());
    String id = testRailResultUpdator.returnTestId(result);
    updateToTestRail(id, result.getName(),  testStatus: 1, UtilityMethods.getBrowserName(browserDetails));
    Reporter.setCurrentTestResult(result);
    String dateAndTime = ScreenCapture.stopVideoCapture(result.getName());
    String sTestName = getTestCaseName(result);
    String sTimeTaken = Long.toString( i: (result.getEndMillis() - result.getStartMillis()) / 1000);
    sTimeTaken = getTimeInMins(sTimeTaken);
    updateRuntimeReport(result, Thread.currentThread().getId(), sTestName, PASSED,  sErrorMessage: "         ",  sAbsoluteImagePath:
            sTimeTaken);
//  UtilityMethods.verifyPopUp();
    String sValue = new ConfigManager().getProperty("VideoCapture");
    sModeOfExecution = new ConfigManager().getProperty("ModeOfExecution");
    String sVideoPath = "";
    if (sValue.equalsIgnoreCase( anotherString: "true") && sModeOfExecution.equalsIgnoreCase( anotherString: "linear")) {
```

In the same way, failed result will be updated to test rail when test case failed by getting testcaseId. Please see code implemented in **onTestFailure()** method.



```java
public void onTestFailure(ITestResult result) {
    String sTestName = getTestCaseName(result);
    depend.writeProperty(result.getName(),  sData: "Fail");
    failures.writePropertyForFailures(result.getMethod().getXmlTest().getName(), result.getName(),  c: "fail");
    log.error("xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n");
    log.error("ERROR ----------" + result.getName() + " has failed-----------------");
    log.error("xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n");
    ITestContext context = result.getTestContext();
    WebDriver driver = (WebDriver) context.getAttribute( s: "driver");
    System.setProperty("org.uncommons.reportng.escape-output", "false");
    Reporter.setCurrentTestResult(result);
    // update results to testrail
    String browserDetails = (String) ((JavascriptExecutor) driver).executeScript( s: "return navigator.userAgent;");
    //String id = getCaseId(result.getName());
    String id = testRailResultUpdator.returnTestId(result);
    updateToTestRail(id, result.getName(),  testStatus: 5, UtilityMethods.getBrowserName(browserDetails),
            result.getThrowable().getMessage());
    String screenshotName = ScreenCapture.saveScreenShot(driver);
    String screenshot = System.getProperty("user.dir") + fileSeperator + "Automation Reports" + fileSeperator
            + "LatestResults" + fileSeperator + "Screenshots" + fileSeperator + screenshotName;
    Throwable errorMessage = result.getThrowable();
    String sErrorMessage;
    try {
        sErrorMessage = result.getThrowable().getMessage();
```

If any test is skipped due to configuration failed then test result will be updated as skipped in test rail by getting testcaseId. Please see below code in **onTestSkipped()** method.



```java
public void onTestSkipped(ITestResult result) {
    log.warn("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@");
    log.warn("WARN ------------" + result.getName() + " has skipped-----------------");
    log.warn("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@");
    String sTimeTaken = Long.toString( i: (result.getEndMillis() - result.getStartMillis()) / 1000);
    sTimeTaken = getTimeInMins(sTimeTaken);
    String sTestName = getTestCaseName(result);
    if(sTimeTaken==null){
        updateRuntimeReport(result, Thread.currentThread().getId(), sTestName, SKIPPED,
                sErrorMessage: "------",  sAbsoluteImagePath: "null",  executionTime: "NA");
    }else{
        updateRuntimeReport(result, Thread.currentThread().getId(), sTestName, SKIPPED,
                result.getThrowable().getMessage(),  sAbsoluteImagePath: "null", sTimeTaken);
    }

    /* updateRuntimeReport(result, Thread.currentThread().getId(), SKIPPED); */
    depend.writeProperty(result.getName(),  sData: "Skip");

    // ************ comment below code if you are using TestNG dependency
    // methods
    ITestContext context = result.getTestContext();
    WebDriver driver = (WebDriver) context.getAttribute( s: "driver");
    String browserDetails = (String) ((JavascriptExecutor) driver).executeScript( s: "return navigator.userAgent;");
    String id = testRailResultUpdator.returnTestId(result);
    updateToTestRail(id, result.getName(),  testStatus: 2, UtilityMethods.getBrowserName(browserDetails));
    Reporter.setCurrentTestResult(result);
```