



Project Details

Usage Document - iAutomate



Contents

1. Objective	3
2. Introduction.....	3
3. Prerequisites.....	3
4. Reference Implementation	3
5. Implementation of the iAutomate Jar.....	4
6. Javadoc	6
7. Troubleshooting Tips.....	7
8. MongoDB CRUD testcases and dynamic web table handling.....	12

1. Objective

Purpose of this document is to make Audience understand guidelines to use iAutomate libraries while using the framework for UI and API automation.

2. Introduction

This document will guide automation engineers, to use iAutomate libraries, by using the Reference Implementation. iAutomate also has elaborate Javadocs for utility methods, like control actions for UI automation, Excel Reader for handling test data files, API Utility methods for enabling UI and API Automation.

3. Prerequisites

- JDK 1.7 or above
- Maven
- Eclipse or any other IDE which follows Maven Structure
- TestNG
- iAutomate JAR
- SVN Access of iAutomate library location (iAutomate.jar):
<https://192.168.77.176/svn/AX360/trunk/iAutomate-Jars>
- SVN Access of Reference Implementation location:
<https://192.168.77.176/svn/AX360/trunk/iAutomate-Ref-Implementation>
- SVn Access of Javadocs location:
<iAutomate-Ref implementation\doc\index.html>

4. Reference Implementation

Please take reference of iAutomate-Ref-Implementation project for building UI and API Automation. TestNG.xml in this project has the testclasses that need to be executed as a sample. Test Classes have 1 testcase automated for UI and 1 for API.

Constants.properties – Contains a list of constants required for UI and API Automation.

Resources folder – Contains chromeDriver.exe and config.properties

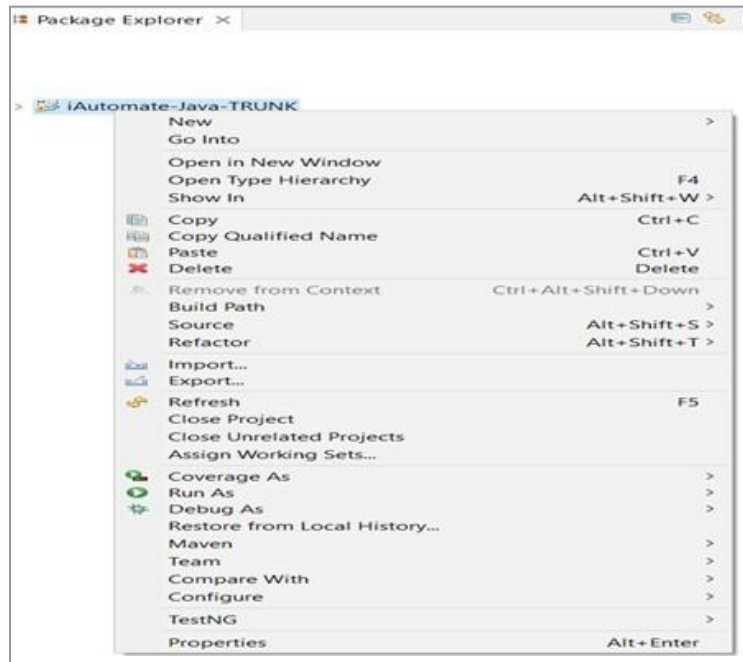
Config.properties contains browser setting can be done in (browser=[chrome](#) / browser=[firefox](#)) and your application url

Jar contains methods to launch Browser, generating extent reports and logs. Also contains custom methods for control Actions, API Utility methods, miscellaneous support like excel Integration, etc.

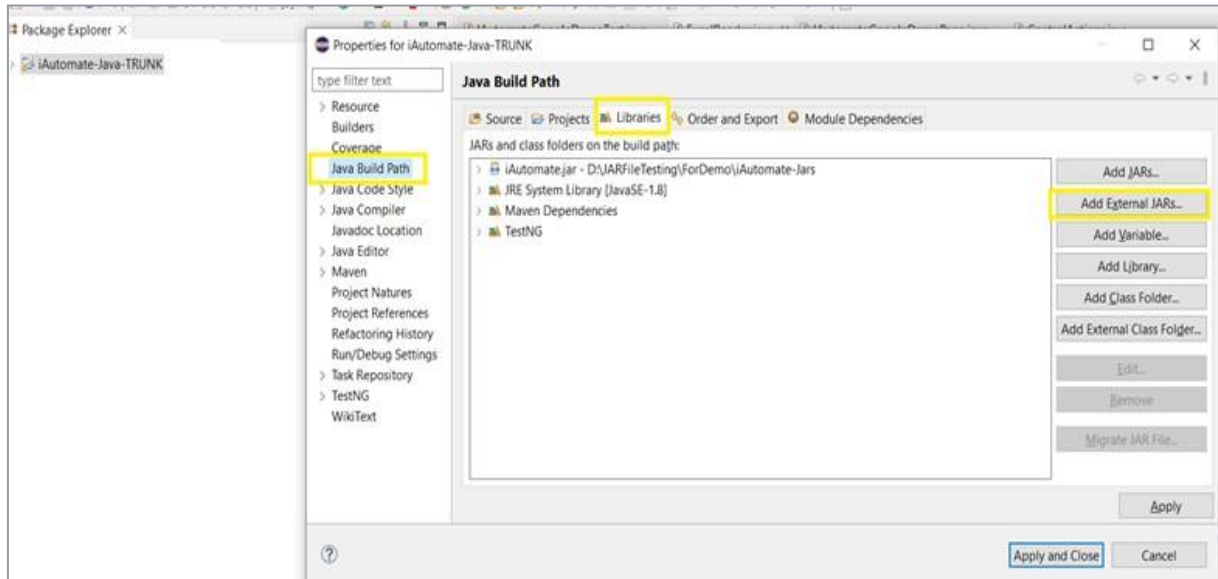
Details of these methods can be found in JavaDocs.

5. Implementation of the iAutomate Jar

Please follow the following steps to use iAutomate.JAR



1. Go to package explorer and select root folder of project into eclipse
2. Click on properties and go to JAVA Build Path Property option.
3. Go to “Libraries” tab.
4. Add external JAR with name as iAutomate. JAR



6. Javadoc

iAutomate Java Docs have a list of all predefined methods available for use. It is a detailed documentation of parameters, return types and purpose for all methods from following classes:

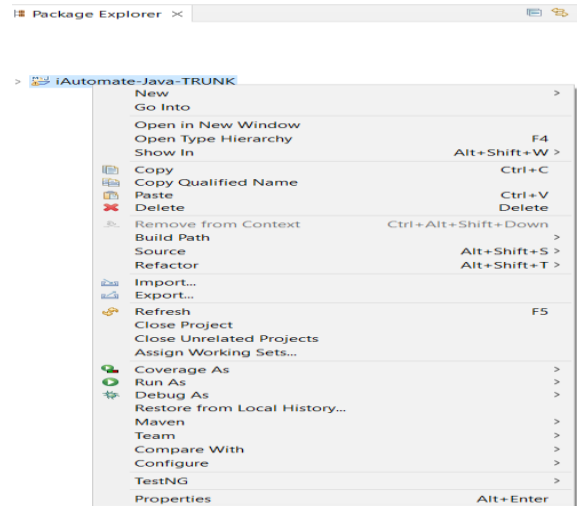
All Classes

- [ApiUtils](#)
- [Constants](#)
- [ControlActions](#)
- [ControlActions.WINDOW_TAB](#)
- [DBHelper](#)
- [DynamicTestNGFile](#)
- [ExcelFileReader](#)
- [ExcelReader](#)
- [GenericUtils](#)
- [JSONUtils](#)
- [Listener](#)
- [MongoConnection](#)
- [MongoDBGenericUtils](#)
- [Retry](#)
- [TestBase](#)
- [WebTable](#)

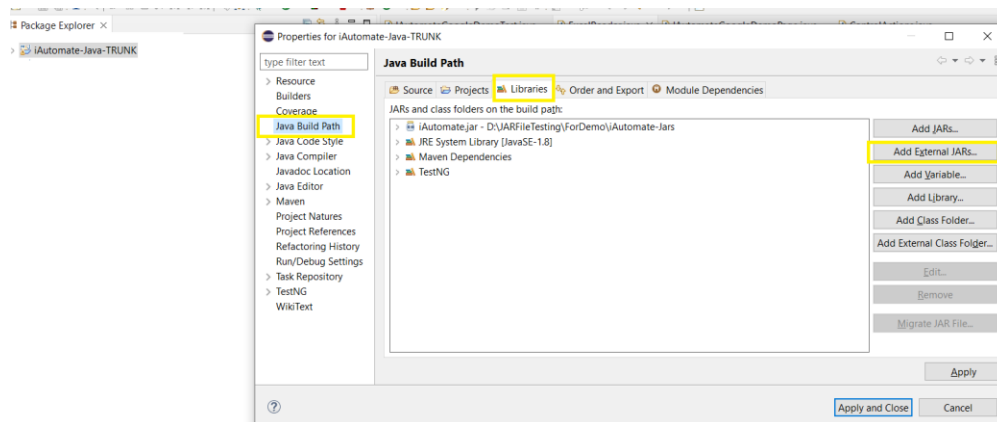
7. Troubleshooting Tips

iAutomate.jar is already bundled with iRef implementation in src/test/resources/Jars.
For adding newer versions of iAutomate jar follow below process

- I. Go to package explorer and select root folder of project into eclipse



- II. Click on properties and go to JAVA Build Path Property option.
- III. Go to “Libraries” tab.
- IV. Add external JAR with name as iAutomate. JAR



1. Javadoc

Please take reference of JAVA Doc for new projects who would use iAutomate, it is the documentation in a set of source file describing usage of following classes

All Classes

- ApiUtils
- Constants
- ControlActions
- ControlActions.WINDOW_TAB
- DBHelper
- DynamicTestNGFile
- ExcelFileReader
- ExcelReader
- GenericUtils
- JSONUtils
- Listener
- MongoConnection
- MongoDBGenericUtils
- Retry
- TestBase
- WebTable

Troubleshooting tips

1:mvn test errors

Issue 1:While executing tests as a TestNG.xml or as 'mvn test' if you face following error:

```
[INFO] -----  
[INFO] BUILD FAILURE  
[INFO] -----  
[INFO] Total time: 15.775 s  
[INFO] Finished at: 2022-02-21T14:30:43+05:30  
[INFO] -----  
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-surefire-plugin:2.17:test (default-test) on project iAutomate-JavaRef: Execution default-test of goal org.apache.maven.plugins:maven-surefire-plugin:2.17:test failed: There was an error in the forked process  
[ERROR] java.lang.NoClassDefFoundError: com/project/testbase/TestBase  
1. Ensure that iAutomate.jar is referenced correctly.  
2. If you are still unable to utilize iAutomate.jar in your project and you get above error: Add a dependency in your pom.xml
```

```
<dependency>  
  <groupId>iAutomate</groupId>  
  <artifactId>Java</artifactId>  
  <version>1.0</version>  
  <scope>system</scope>  
  <systemPath>PATH WHERE JAR IS LOCATED ON LOCAL MACHINE\iAutomate.jar</systemPath>  
</dependency>
```

If replacing existing iAutomate jar with newest version

Issue 2: If you get error running on JRE instead of JDK while running mvn test then set runtime env to jdk .

[ERROR] No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK?

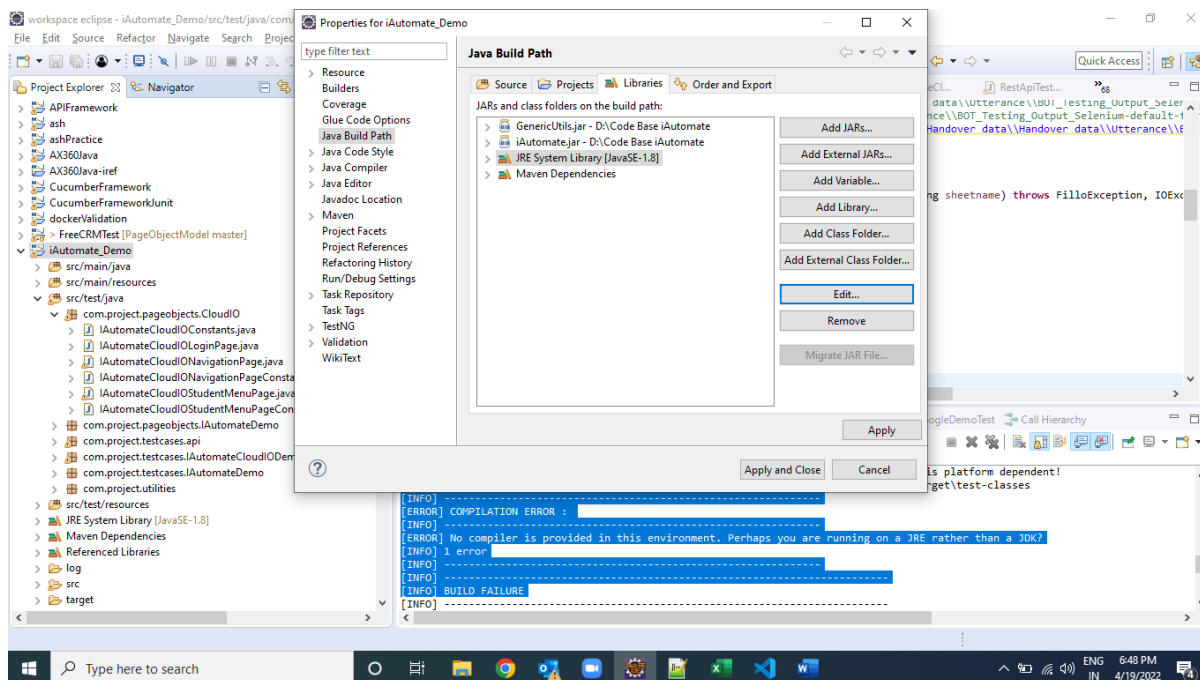
[INFO] 1 error

[INFO] -----

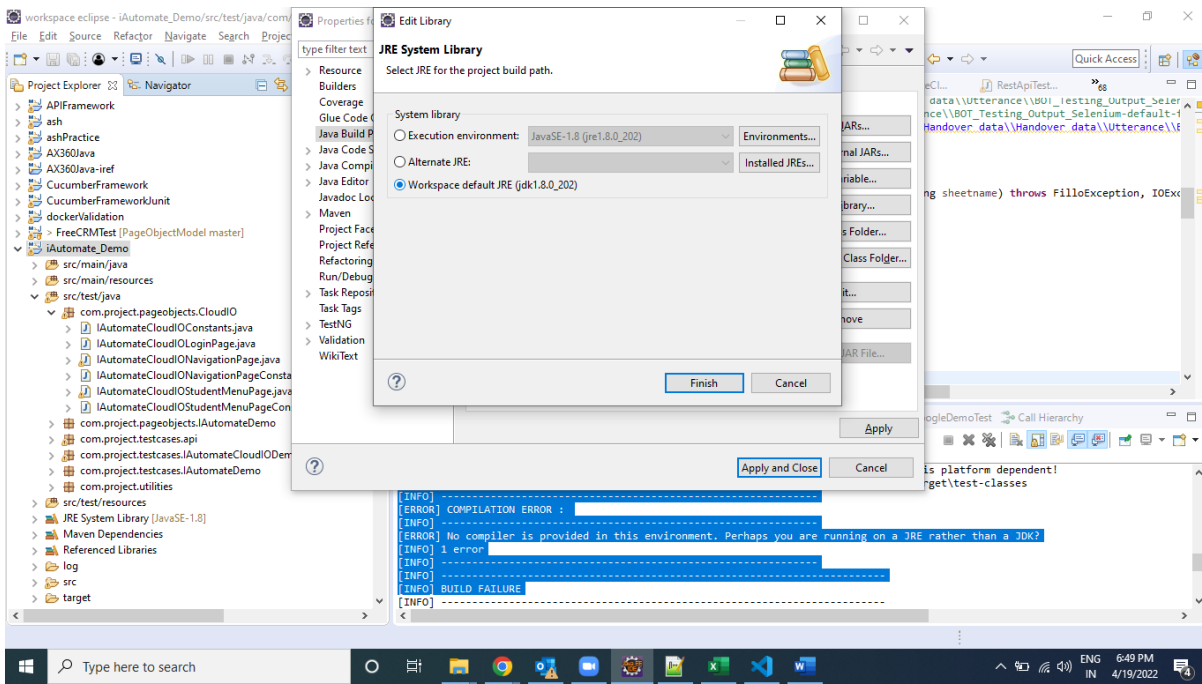
[INFO] -----

[INFO] BUILD FAILURE

Then click on properties → Java build path



Click on runtime environment as jdk 1.8 instead of jre 1.8 and click on Finish.



2.constants.properties missing

If constants.properties is missing. As this file has been added in newest version [TestNGContentHandler] [WARN] It is strongly recommended to add "<!DOCTYPE suite SYSTEM \"https://testng.org/testng-1.0.dtd\" >" at the top of the suite file [D:\Handover data\iAutomate-Jars\iAutomate_java\testng.xml] otherwise TestNG may fail or not work as expected.

[java.io.FileNotFoundException: D:\Handover data\iAutomate-Jars\iAutomate_java\src\test\resources\constants.properties](#)

If replacing existing iAutomate jar then copy constants.properties file into src/test/resources



constants.properties

3. webdrivermanager missing in pom.xml

launchBrowser(com.project.testcases.IAutomateDemo.IAutomateGoogleDemoTest) Time elapsed: 1.177 sec <<< FAILURE!

java.lang.NoClassDefFoundError: io/github/bonigarcia/wdm/WebDriverManager

add below dependency into pom.xml

```
<dependency>
  <groupId>io.github.bonigarcia</groupId>
  <artifactId>webdrivermanager</artifactId>
  <version>5.0.3</version>
</dependency>
```

4. Merged ExcelFileReader with ExcelReader

If using **import** com.project.utilities.ExcelFileReader; in any class please remove that entry and replace it with **import** com.project.utilities.ExcelReader;

Methods in ExcelFileReader has been merged with ExcelReader

5. If using only iAutomate.jar in existing project then do following steps

- add below entries in pom.xml

```
<dependency>
    <groupId>com.microsoft.azure</groupId>
    <artifactId>azure-storage</artifactId>
    <version>8.6.6</version>
</dependency>

driver -->
<!-- https://mvnrepository.com/artifact/org.mongodb/mongo-java-
<dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongo-java-driver</artifactId>
    <version>3.12.11</version>
</dependency>

<dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>7.4.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.codehaus.groovy</groupId>
    <artifactId>groovy-all</artifactId>
    <version>3.0.8</version>
    <type>pom</type>
    <exclusions>
        <exclusion>
            <artifactId>tools</artifactId>
            <groupId>com.sun</groupId>
        </exclusion>
    </exclusions>
</dependency>

<dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>2.5</version>
</dependency>
<dependency>
    <groupId>io.github.bonigarcia</groupId>
    <artifactId>webdrivermanager</artifactId>
    <version>5.0.3</version>
</dependency>
<dependency>
```

```

        <groupId>io.appium</groupId>
        <artifactId>java-client</artifactId>
        <version>7.3.0</version>
    </dependency>

```

- Add new constants.properties in src/test/resources mentioned in Step 2. Basically all fields previously mentioned in Constants.java is moved to constants.properties
- For data driven testing of UI testcases file is maintained in location
 \iAutomate-Codebase-BaseVersion\test-data-files\UI-TestData\GoogleTests.xlsx. Create this folder structure if not present. This is already present in iAutomateRefImpl project
- ExcelFileReader class has been merged to ExcelReader class. If utilizing this class refer it to com.project.utilities.ExcelReader in imports.

8. MongoDB CRUD testcases handling and dynamic web table handling

Following generic functions for Mongo DB has been added in MongoDBGenericUtils.java

- InsertOne
- InsertMany
- DeleteOne
- DeleteMany
- UpdateOne
- UpdateMany
- FindOne
- FindMany
- Replace
- BulkWrite

MongoDBTestsCRUD.java has all tests for the above functions and its usage.

Mongo Db connection parameters are defined in constants.properties. You can edit the connection string and username and password as per your requirement.

```

44 user = Username
45 pwd = Password
46 mongoduri = mongodb://Username:Password@localhost:27017/?maxPoolSize=20&w=majority
47 mongoatlas= mongodb://Username:Password@localhost:27017/?maxPoolSize=20&w=majority
48 mongodatabase= sample_mflix
49 mongocollection= movies

```

Dynamic WebTable Handling

Two new generic methods have been added to get table data either by column or by row. Both have common properties to be provided that **rowxpath** which contains xpath of all rows in the table, **columnxpath**-suppose if there are multiple columns in a table and we need only a particular column data for the testcase we need to provide the column xpath. **Action** -the action to be taken on the particular cell whether data is to be read, clicked, text to be sent .

A POJO class named WebTable has been created if there are multi column action can be performed. Both these implementations of dynamic web table is provided in IAutomateWebtable.java and IAutomateWebtable2.java