



Java Executor Package

Readme

Version 1.0.0

11/09/2021

Table of Contents

1. Introduction	3
1.1 Overview	3
1.2 Use cases	3
2. Requirements & Prerequisites	4
2.1 System Requirements	4
2.2 Prerequisites	4
3. Getting Started	5
3.1 Quick Start	5
3.1.1 Setup	5
3.1.2 Configuration and Use	5
3.1.3 Data Type Mapping	9
4. Support & FAQs	11
4.1 Support	11
4.2 FAQs	11
Appendix A: Record of Changes	12
Appendix B: References	13

1. Introduction

This document contains all essential information for the user to make full use of this Automation 360 **Java Executor Package**. It includes a description of the functions and capabilities and step-by-step procedures for setup & configuration of the **Java Executor Package**.

1.1 Overview

The package provides actions to load Java JAR files which contain executable Java classes. You can execute Java methods in a sequence or in parallel by using Java Threads.

It uses *java.lang.reflect* which provides classes and interfaces for obtaining reflective information about classes and objects.

1.2 Use cases

The key use cases include:

- Execute complex Java logic without creating an Automation 360 package
- Execute logic in parallel using Java Threads

2. Requirements & Prerequisites

2.1 System Requirements

- Enterprise Automation 360.21+

2.2 Prerequisites

No prerequisites

3. Getting Started

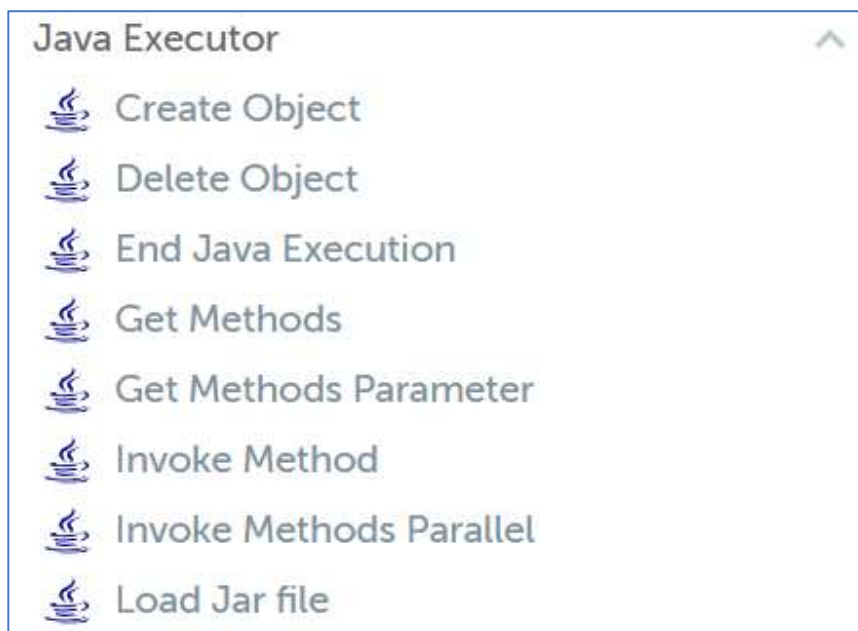
3.1 Quick Start

3.1.1 Setup

Import the Automation 360 Java Executor Package in the Control Room under Bots->Packages.

3.1.2 Configuration and Use

The package contains the following command actions



A sample Jar file 'ExecutorTestClasses.jar' and its source code as a Gradle project 'JavaExecutorTest_GradleProject.zip' is provided. If you use any additional classes please make sure that you build a Jar file that contains all 3rd party Java classes that are needed to execute the Java methods E.g. some of the sample methods depend on 'com.squareup.okhttp3'. So these Java classes are included in the sample Jar file.

Name	Description	Parameter	
Load Jar File	Load a Java Archive File	Session	Session name
		Jar File	File to be loaded
		Class Path	Full class path of the Java Class
		Constructor Parameter Types	List of Data types if a object constructor exists that has a list of arguments. Full type name, e.g. 'java.lang.String'
Get Methods	Get Methods of the Java class	Session	Session name
		Methods	List of Java methods
Get Methods Parameter	Get the list of Methods Parameter	Session	Session name
		Method Name	Name of the Java method
		Parameters	Dictionary of method parameters, key = parameter name, value = parameter data type
Create Object	Creates a Java object	Session	Session name
		Object Name	Identifier Name of the Java Object for 'Invoke' actions
		Object Parameter Values	List of parameter values, sequence and data types must match the data types list provided in 'Load Jar File'
		Status (optional)	Status of object creation
Delete Object	Removes Java object from current session	Session	Session name
		Object Name	Identifier Name of the Java Object
		Status (optional)	Status of object removal

Invoke Method	Invokes a Java method of a Java object	Session	Session name
		Object Name	Identifier Name of the Java Object for execute actions
		Method Name	Name of the Java method
		Method Parameter Values (optional)	List of parameter values, must match the parameter data types. See section 3.1.3 how data types are mapped
		Is Static	If class methods are static. In this case no Java object will be passed to the method call
		Map Types	If Bot data types should be mapped to Java Data Types or not. See section 3.1.3 how data types are mapped
		Result	Result of the method execution See section 3.1.3 how data types are mapped
Invoke Methods Parallel	Invokes a list of Java methods of a Java object in parallel, each in its own Java Thread	Session	Session name
		Object Name	Identifier Name of the Java Object for execute actions
		Method Names	List of Java method names
		Method Parameter Values (optional)	List of parameter value lists. Must match the parameter data types. Sequence of list must match the sequence of list 'Methods Names'. See section 3.1.3 how data types are mapped
		Is Static	If class methods are static. In this case no Java object will be passed to the method call

		Map Types	If Bot data types should be mapped to Java Data Types or not. See section 3.1.3 how data types are mapped
		Results	Result of each method execution. See section 3.1.3 how data types are mapped
End Java Execution	Ends Java Execution	Session	Session name

3.1.3 Data Type Mapping

The actions 'Invoke Method' and 'Invoke Methods Parallel' have the flag 'Map Types' which determines if parameter values and methods results should be mapped to native Java data types or not.

'Map Types' = FALSE

The Java class methods should use the Bot API Java data types for parameter values and return values. The package has been tested with these Bot API data types, others might work as well

Bot Variable Data Type	Java Data Type
String	com.automationanywhere.botcommand.data.impl.StringValue
Number	com.automationanywhere.botcommand.data.impl.NumberValue
Boolean	com.automationanywhere.botcommand.data.impl.BooleanValue
Datetime	com.automationanywhere.botcommand.data.impl.DateTimeValue
List	com.automationanywhere.botcommand.data.impl.ListValue
Dictionary	com.automationanywhere.botcommand.data.impl.DictionaryValue
Table	com.automationanywhere.botcommand.data.impl.TableValue

'Map Types' = TRUE

Data types of Java parameter values and return values will be mapped from Bot API Java data types to native Java data types

Bot Variable Data Type	Java Data Type
String	Java.lang.String . If the return value is a complex object, the actions try to convert it to a JSON object using the class ' com.fasterxml.jackson.databind.ObjectMapper '
Number	Based on the value format converted to java.lang.Integer or java.lang.Double. See algorithm below
Boolean	Java.lang.Boolean
List	Java.lang.Object[], each list entry will be converted based on the above mapping rules

Java code to detect an Integer or Double Java object

```
public boolean isDouble(NumberValue value) {  
  
    Double dvalue = value.get();  
    Integer ivalue = dvalue.intValue();  
    Double diff = dvalue - Double.valueOf(ivalue);  
    return (diff != 0.0000000000);  
  
}
```

4. Support & FAQs

4.1 Support

Free bots are not officially supported. You can get access to Community Support through the following channels:

- You can get access to Community Support, connecting with other Automation Anywhere customers and developers on [APeople](#) – the [Bot Building Forum](#), the [Bot Store Support Forum](#), or the [Developers Everywhere Group](#).
- Automation Anywhere also provides a [Product Documentation portal](#) which can be accessed for more information about our products and guidance on [Enterprise Automation 360](#).

4.2 FAQs

For questions relating to Enterprise Automation 360: See the [Enterprise Automation 360 FAQs](#).

Appendix A: Record of Changes

Instructions: Provide information on the version number, the date of the version, the author/owner of the version, and a brief description of the reason for creating the revised version.

No.	Version Number	Date of Change	Author	Notes
1	1.0.0	11/09/2021	Stefan Karsten	Initial version

Appendix B: References

No.	Topic	Reference Link
1	Overview of Enterprise Automation 360	Click here
2	Guidance: Building basic Automation 360 bots	Click here
3	Guidance: Building Automation 360 action packages	Click here
4	APeople Community Forum	Click here
5	Automation Anywhere University	Click here