Automic
Let's Automate Business.

Automic JAVA API

A Simple Getting Started

Brendan Sapience – bsp@automic.com

Our Goal with this getting started

Automic™
Let's Automate Business.

- Showing what resources are needed to use the API
- Writing a short Java program in order to:
  - Connect to an Automic (AE) instance
  - List all existing Folders and existing Objects for your Client

Why are we doing it?

Automic™
Let's Automate Business.

Because anything can be automated with the Java API in Automic

- It is **fully** supported
- It is **fully** documented

What do we need?

- □ Access to an Automic Environment
- □ Access to the Automic rich GUI binaries
- □ Java JRE or JDK 1.7+ installed. Find it here
- □ Any JAVA IDE (Integrated Development Environment)
  - This getting started uses Eclipse. Find it here

Our Process through this getting started

Automic™

Let's Automate Business.

Step 1
Retrieve the Sample Code

Step 2
Import the Code

Step 3
Run the Code

Step 1
Retrieve the Sample Code

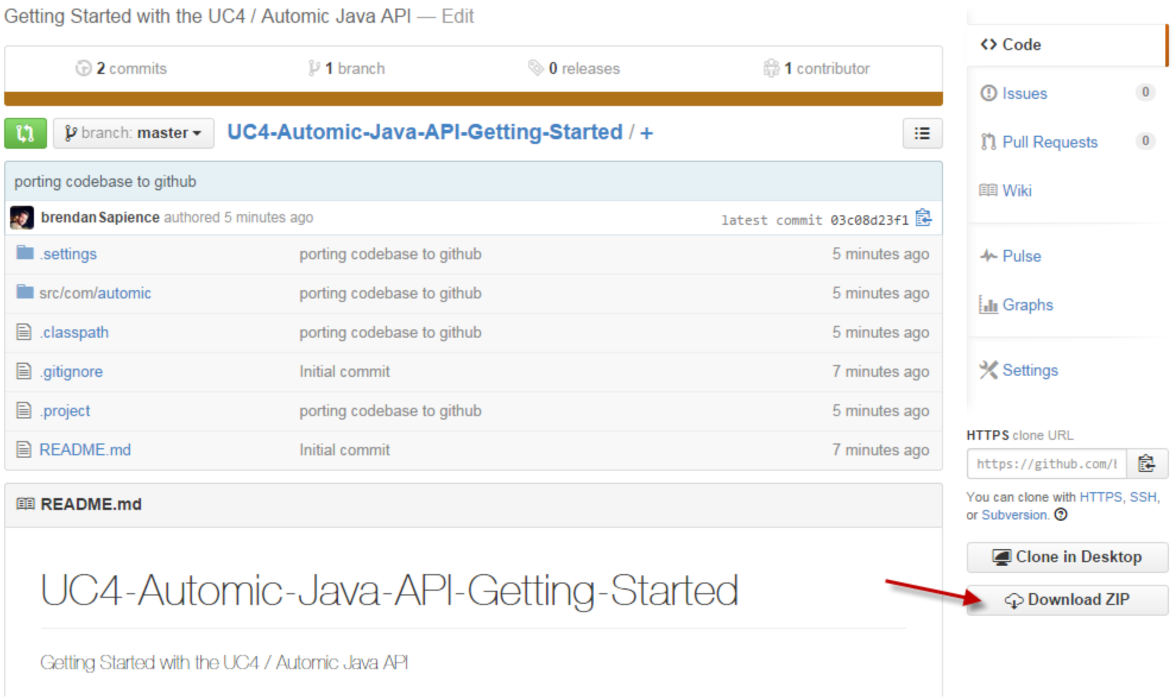
Go to public repository: https://github.com/brendanSapience/UC4-Automic-Java-API-Getting-Started

Alternatively: You can clone the git repository locally by using the "clone" command in Git.

Step 1
Retrieve the Sample Code

Automic™
Let's Automate Business.

Click on the Download section:



Getting Started with the UC4 / Automic Java API — Edit

2 commits | 1 branch | 0 releases | 1 contributor

branch: master ▾ UC4-Automic-Java-API-Getting-Started / +

porting codebase to github

brendan Sapience authored 5 minutes ago latest commit 03c08d23f1

.settings	porting codebase to github	5 minutes ago
src/com/automic	porting codebase to github	5 minutes ago
.classpath	porting codebase to github	5 minutes ago
.gitignore	Initial commit	7 minutes ago
.project	porting codebase to github	5 minutes ago
README.md	Initial commit	7 minutes ago

README.md

UC4-Automic-Java-API-Getting-Started

Getting Started with the UC4 / Automic Java API

HTTPS clone URL

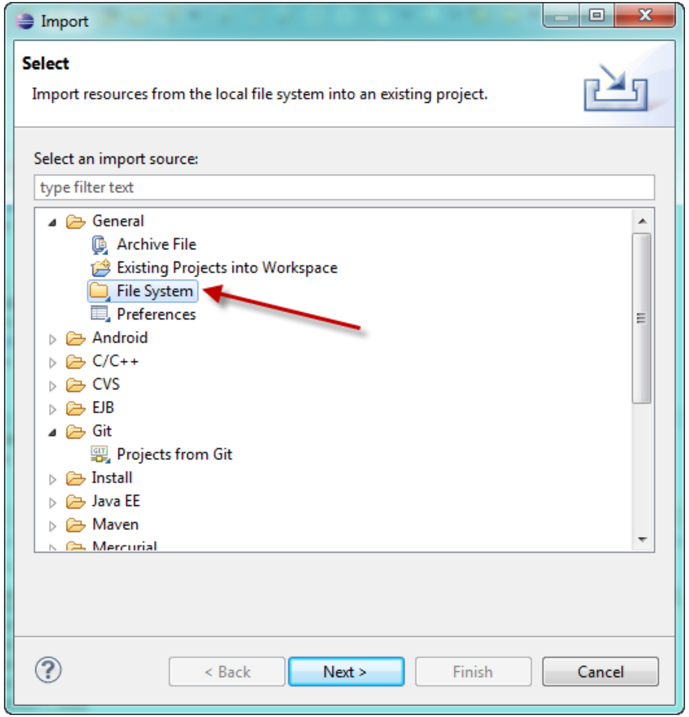
https://github.com/l

You can clone with HTTPS, SSH, or Subversion.

Step 2
Import the Code

In Eclipse: File -> Import.

Under General, pick "File System"



Import

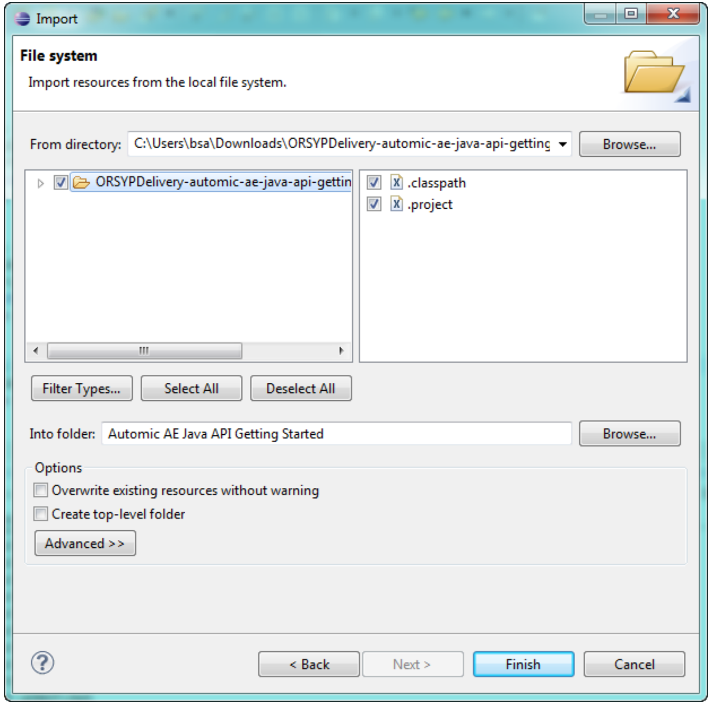
Select

Import resources from the local file system into an existing project.

Select an import source:

Step 2
Import the Code

Import the folder downloaded in Step 1



Import

File system

Import resources from the local file system.

From directory: C:\Users\bsa\Downloads\ORSYPDelivery-automic-ae-java-api-getting

☑ ORSYPDelivery-automic-ae-java-api-gettin

☑ .classpath

☑ .project

Into folder: Automic AE Java API Getting Started

Options

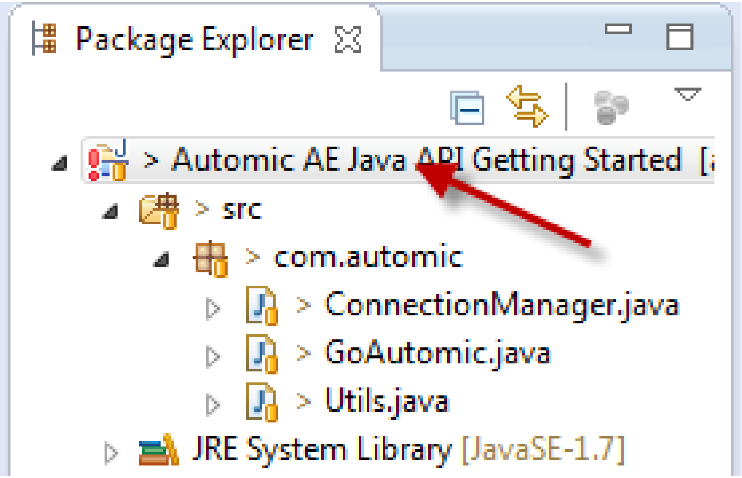
□ Overwrite existing resources without warning

□ Create top-level folder

Step 2
Import the Code

Automic™
Let's Automate Business.

Right click on the project name,
Then click "Properties"



Package Explorer

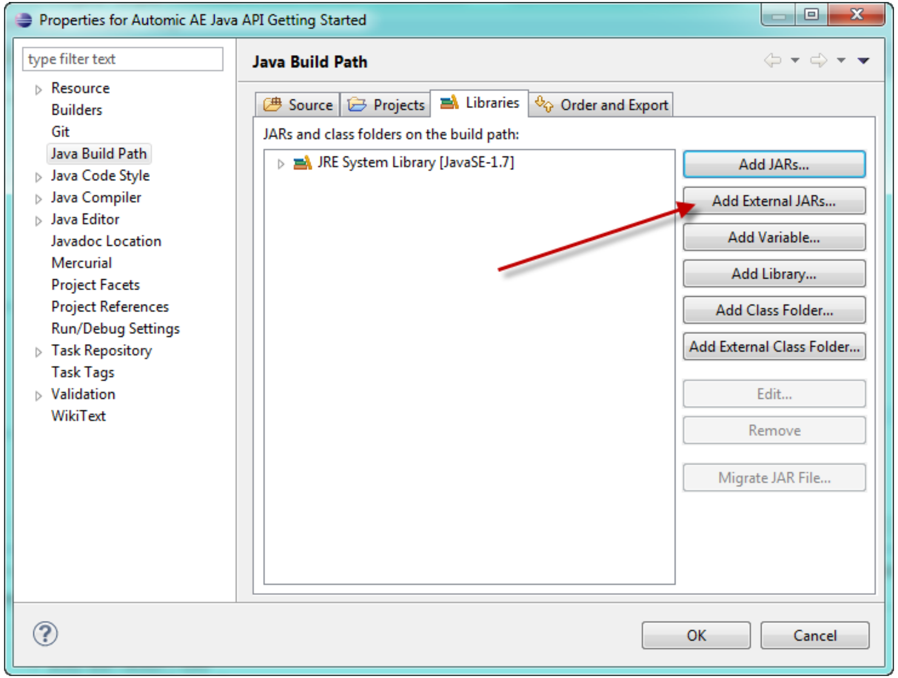
- > Automic AE Java API Getting Started [a
  - > src
    - > com.automic
      - > ConnectionManager.java
      - > GoAutomic.java
      - > Utils.java
  - JRE System Library [JavaSE-1.7]

Step 2
Import the Code

Automic™
Let's Automate Business.

Under Java Build Path -> Libraries,

Click "Add External JARs"



Properties for Automic AE Java API Getting Started

Java Build Path

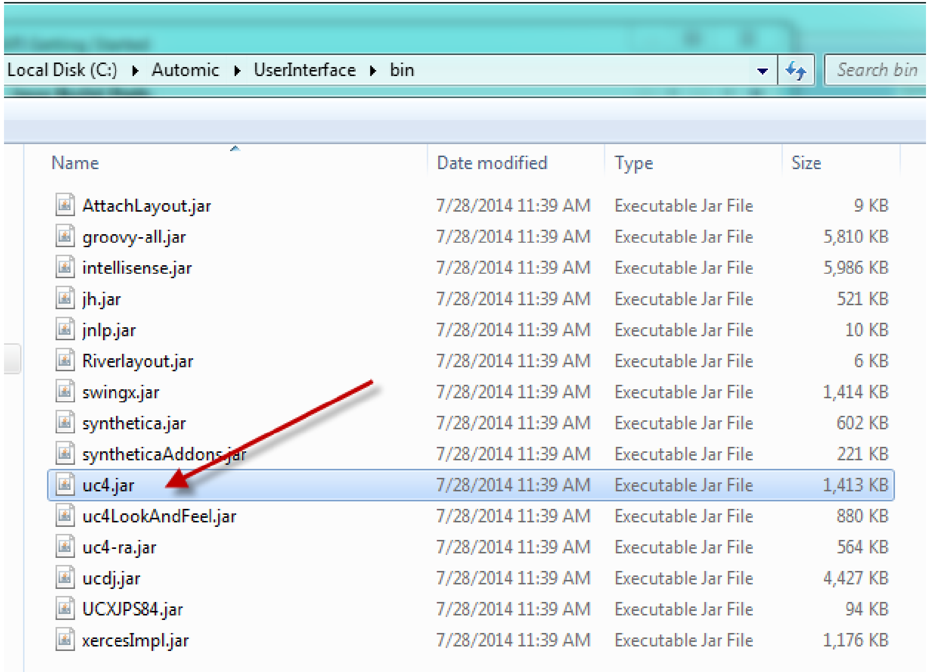
JARs and class folders on the build path:

JRE System Library [JavaSE-1.7]

Step 2
Import the Code

Automic™
Let's Automate Business.

Go to the User Interface "bin" folder, Select "uc4.jar"



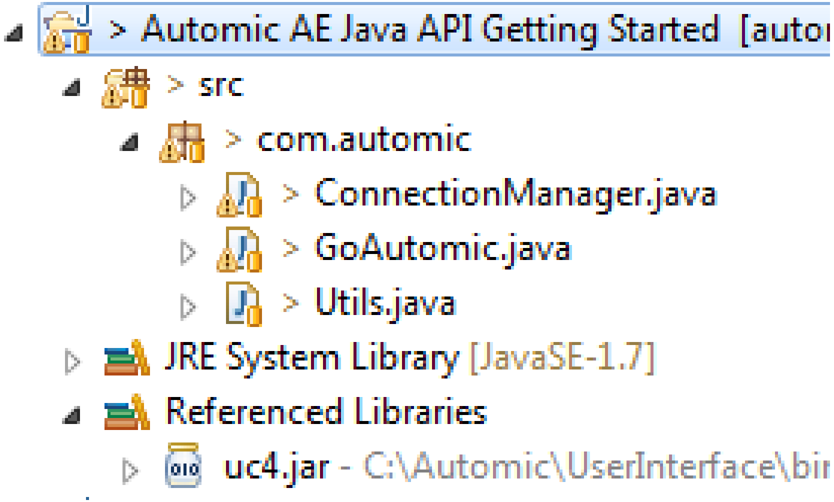
Local Disk (C:) ▸ Automic ▸ UserInterface ▸ bin

Search bin

Name	Date modified	Type	Size
AttachLayout.jar	7/28/2014 11:39 AM	Executable Jar File	9 KB
groovy-all.jar	7/28/2014 11:39 AM	Executable Jar File	5,810 KB
intellisense.jar	7/28/2014 11:39 AM	Executable Jar File	5,986 KB
jh.jar	7/28/2014 11:39 AM	Executable Jar File	521 KB
jnlp.jar	7/28/2014 11:39 AM	Executable Jar File	10 KB
Riverlayout.jar	7/28/2014 11:39 AM	Executable Jar File	6 KB
swingx.jar	7/28/2014 11:39 AM	Executable Jar File	1,414 KB
synthetica.jar	7/28/2014 11:39 AM	Executable Jar File	602 KB
syntheticaAddons.jar	7/28/2014 11:39 AM	Executable Jar File	221 KB
uc4.jar	7/28/2014 11:39 AM	Executable Jar File	1,413 KB
uc4LookAndFeel.jar	7/28/2014 11:39 AM	Executable Jar File	880 KB
uc4-ra.jar	7/28/2014 11:39 AM	Executable Jar File	564 KB
ucdj.jar	7/28/2014 11:39 AM	Executable Jar File	4,427 KB
UCXJPS84.jar	7/28/2014 11:39 AM	Executable Jar File	94 KB
xercesImpl.jar	7/28/2014 11:39 AM	Executable Jar File	1,176 KB

Step 2
Import the Code

You should now see your project with a reference to uc4.jar

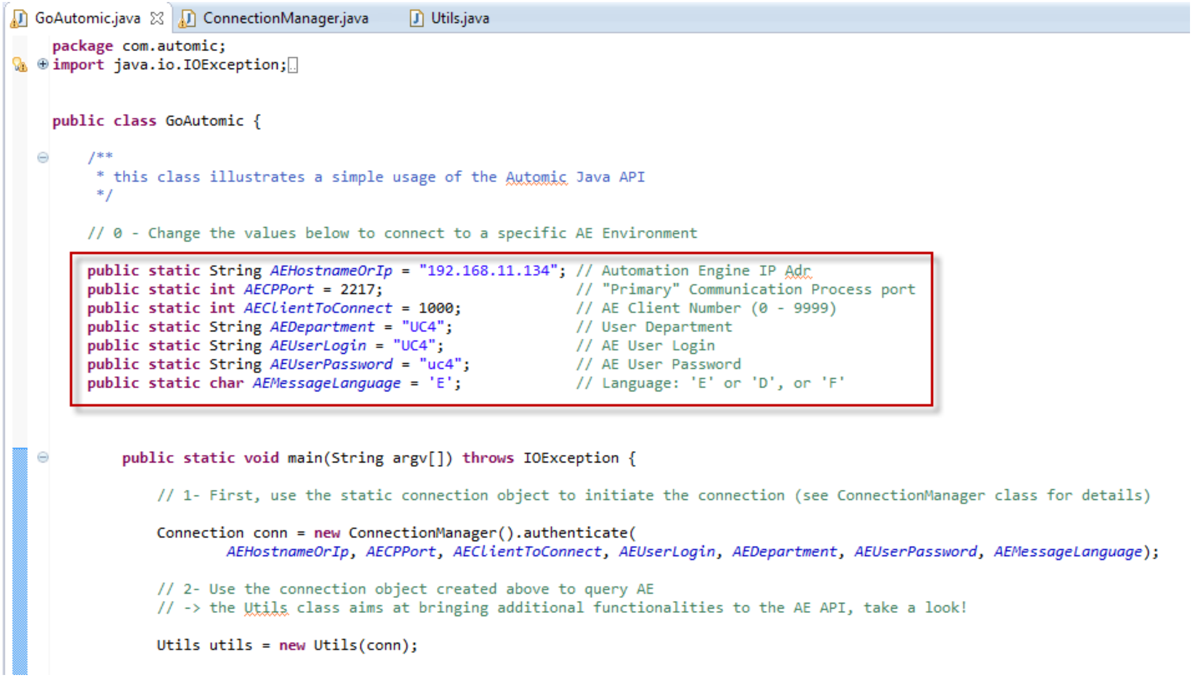


```
◢ > Automic AE Java API Getting Started  [autoı
  ◢ > src
    ◢ > com.automic
      ▷ > ConnectionManager.java
      ▷ > GoAutomic.java
      ▷ > Utils.java
  ▷ JRE System Library [JavaSE-1.7]
  ◢ Referenced Libraries
    ▷ uc4.jar - C:\Automic\UserInterface\biı
```

Step 3
Run the Code

Automic™
Let's Automate Business.

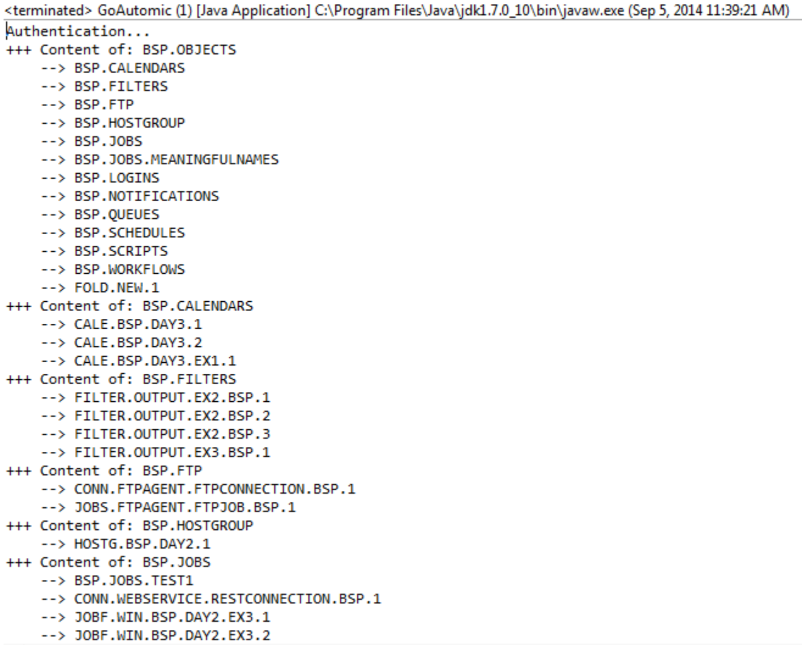
In your GoAutomic.java class, modify the connection parameters to your own:



```java
package com.automic;
import java.io.IOException;

public class GoAutomic {

    /**
     * this class illustrates a simple usage of the Automic Java API
     */

    // 0 - Change the values below to connect to a specific AE Environment

    public static String AEHostnameOrIp = "192.168.11.134"; // Automation Engine IP Adr
    public static int AECPPort = 2217;                      // "Primary" Communication Process port
    public static int AEClientToConnect = 1000;             // AE Client Number (0 - 9999)
    public static String AEDepartment = "UC4";              // User Department
    public static String AEUserLogin = "UC4";               // AE User Login
    public static String AEUserPassword = "uc4";            // AE User Password
    public static char AEMessageLanguage = 'E';             // Language: 'E' or 'D', or 'F'


        public static void main(String argv[]) throws IOException {

            // 1- First, use the static connection object to initiate the connection (see ConnectionManager class for details)

            Connection conn = new ConnectionManager().authenticate(
                    AEHostnameOrIp, AECPPort, AEClientToConnect, AEUserLogin, AEDepartment, AEUserPassword, AEMessageLanguage);

            // 2- Use the connection object created above to query AE
            // -> the Utils class aims at bringing additional functionalities to the AE API, take a look!

            Utils utils = new Utils(conn);
```

Step 3
Run the Code

Run the code and see the result (a list of all Folders and Objects within them):



```
<terminated> GoAutomic (1) [Java Application] C:\Program Files\Java\jdk1.7.0_10\bin\javaw.exe (Sep 5, 2014 11:39:21 AM)
Authentication...
+++ Content of: BSP.OBJECTS
    --> BSP.CALENDARS
    --> BSP.FILTERS
    --> BSP.FTP
    --> BSP.HOSTGROUP
    --> BSP.JOBS
    --> BSP.JOBS.MEANINGFULNAMES
    --> BSP.LOGINS
    --> BSP.NOTIFICATIONS
    --> BSP.QUEUES
    --> BSP.SCHEDULES
    --> BSP.SCRIPTS
    --> BSP.WORKFLOWS
    --> FOLD.NEW.1
+++ Content of: BSP.CALENDARS
    --> CALE.BSP.DAY3.1
    --> CALE.BSP.DAY3.2
    --> CALE.BSP.DAY3.EX1.1
+++ Content of: BSP.FILTERS
    --> FILTER.OUTPUT.EX2.BSP.1
    --> FILTER.OUTPUT.EX2.BSP.2
    --> FILTER.OUTPUT.EX2.BSP.3
    --> FILTER.OUTPUT.EX3.BSP.1
+++ Content of: BSP.FTP
    --> CONN.FTPAGENT.FTPCONNECTION.BSP.1
    --> JOBS.FTPAGENT.FTPJOB.BSP.1
+++ Content of: BSP.HOSTGROUP
    --> HOSTG.BSP.DAY2.1
+++ Content of: BSP.JOBS
    --> BSP.JOBS.TEST1
    --> CONN.WEBSERVICE.RESTCONNECTION.BSP.1
    --> JOBF.WIN.BSP.DAY2.EX3.1
    --> JOBF.WIN.BSP.DAY2.EX3.2
```

You are done! But what is next?

Modify the Code, Explore and Experiment using the following resources:

API Documentation:

http://docs.automic.com/documentation/AE/10.0.2/english/AE_API/pages/index.html

Extended version of the code used here (more methods, more complex):

https://github.com/brendanSapience/UC4-Automic---Java-API-Framework-Simplified