PEtALS-SE-Bonita

This document explain how to use the petals-se-bonita JBI component.

Open Wide

Guillaume Decarnin <guillaume.decarnin@openwide.fr>

- October 2007 -

Table of Contents

PEtALS-SE-Bonita

This service engine is used to interact with Bonita workflows: instanciate projects, start and stop activities, set workflow properties.

See http://wiki.bonita.objectweb.org.

Chapter 1. Operations

1.1. listProjects

Purpose
Message format
Engine return

Return a list of projects (Bonita process models). Optional: include the initial properties (`returnProperties = true`) and list only the models startable by the user (`startableOnly = true` for using the initiator mappers).

```
<content>
    <login>bsoa</login>
    <returnProperties>true</returnProperties>
    <startableOnly>true</startableOnly>
</content>
```

```
<content>
    <process projectName="...">
        <key1>value</key1>
        <key2>value</key2>
    </process>
    <process projectName="...">
        <key1>value</key1>
        <key2>value</key2>
    </process>
    <process projectName="...">
        <key1>value</key1>
        <key2>value</key2>
    </process>
</content>
```

1.2. getProjectProperties

Purpose
Message format
Engine return

Return the initial project properties.

```
<content>
    <process projectName="MyBonitaWorkflow"/>
</content>
```

```
<content>
    <process projectName="MyBonitaWorkflow">
        <key1>value</key1>
        <key2>value</key2>
    </process>
</content>
```

1.3. instanciateProject

Purpose
Message format - sample 1
Message format - sample 2
Engine return

This operation creates a new instance of the specified project. The instance properties and the activity properties (optional) are set with transmitted values.

```
<content>
    <process projectName="MyBonitaWorkflow">
```

```
        <key1>value</key1>
    </process>
</content>
```

```
<content>
    <process projectName="MyBonitaWorkflow" projectVersion="1.1">
        <key1>value</key1>
    </process>
    <activity activityName="FirstActivity">
        <key3>value</key3>
    </activity>
</content>
```

The engine returns the instance name and all properties:

```
<content>
    <process projectName="MyBonitaWorkflow" instanceName="MyBonitaWorkflow_instance36">
        <key1>value</key1>
        <key2>defaultValue</key2>
    </process>
    <activity activityName="FirstActivity" projectName="MyBonitaWorkflow"
 instanceName="MyBonitaWorkflow_instance36">
        <key3>value</key3>
    </activity>
</content>
```

1.4. listUserActivities

Purpose
Message format
Engine return

Return a list of activities of a user (to do list). Optional: include the activity properties (`returnProperties = true`) and include terminated activities (`returnTerminatedActivities = true`).

```
<content>
    <login>bsoa</login>
    <returnProperties>true</returnProperties>
    <returnTerminatedActivities>true</returnTerminatedActivities>
</content>
```

```
<content>
    <activity activityName="..." projectName="..." instanceName="..." state="EXECUTING">
        <key1>value</key1>
        <key2>value</key2>
    </activity>

    <activity activityName="..." projectName="..." instanceName="..." state="TERMINATED">
        <key1>value</key1>
        <key2>value</key2>
    </activity>
</content>
```

1.5. startActivity

Purpose
Message format
Engine return

Bonita Engine can start the activities who have the state "Ready". The instance properties and the activity properties are updated with transmitted values.

```
<content>
    <process instanceName="MyBonitaWorkflow_instance36"/>
    <activity activityName="MyActivity">
        <key2>value</key2>
    </activity>
```

```
</content>
```

```
<content>
    <process projectName="MyBonitaWorkflow" instanceName="MyBonitaWorkflow_instance36">
        <key1>defaultValue</key1>
    </process>
    <activity activityName="MyActivity" projectName="MyBonitaWorkflow"
 instanceName="MyBonitaWorkflow_instance36">
        <key2>value</key2>
        <key3>defaultValue</key3>
    </activity>
</content>
```

1.6. getActivityProperties

Purpose
Message format
Engine return

Return the instance properties and the activity properties.

```
<content>
    <process instanceName="MyBonitaWorkflow_instance36"/>
    <activity activityName="MyActivity"/>
</content>
```

```
<content>
    <process projectName="MyBonitaWorkflow" instanceName="MyBonitaWorkflow_instance36">
        <key1>value</key1>
        <key2>value</key2>
    </process>
    <activity activityName="MyActivity" projectName="MyBonitaWorkflow"
 instanceName="MyBonitaWorkflow_instance36">
        <key3>value</key3>
        <key4>value</key4>
    </activity>
</content>
```

1.7. setActivityProperties

Purpose
Message format
Engine return

Set the activity properties and the instance properties with the specified values.

```
<content>
    <process instanceName="MyBonitaWorkflow_instance36">
        <key1>value1</key1>
        <key2>value2</key2>
    </process>
    <activity activityName="MyActivity">
        <key1>11</key1>
        <key2>22</key2>
    </activity>
</content>
```

If there is no exception, the returned message is:

```
<success/>
```

1.8. terminateActivity

Purpose
Message format
Engine return

Bonita Engine can terminate the activities who have the state "Executing". The instance properties and the activity properties are updated with transmitted values.

```
<content>
    <process instanceName="MyBonitaWorkflow_instance36"/>
    <activity activityName="MyActivity">
        <key1>value</key1>
    </activity>
</content>
```

If there is no exception, the returned message is:

```
<success/>
```

Chapter 2. Service Unit configuration

The key-value extension is used to configure the Service Unit:

- **rmi-url**: URL of the JOnAS RMI registry used to access EJB.
- **loginContextName**: the login contexts are in the file `jaas.config`.
- **login**: Bonita login of the default user.
- **password**: Bonita password of the default user.
- **role**: Bonita role of the default user (ex.: BONITAUSER).
- **loginX**: Bonita login of the user X (replace X by a string id).
- **passwordX**: Bonita password of the user X (replace X by a string id).
- **roleX**: Bonita role of the user X (replace X by a string id).

Several users can be specified using different logins and roles or passwords (see samples below). There are two authentication methods: direct authentication (with password) or using Jonas propagation context (with role).

Sample of `jbi.xml` file with login/password authentication:

```
<?xml version="1.0" encoding="UTF-8"?>
<jbi:jbi version="1.0" xmlns:jbi="http://java.sun.com/xml/ns/jbi"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:petals="http://petals.ow2.org/extensions"
  xmlns:keyvalue="http://petals.ow2.org/extensions/key-value/">
  <jbi:services binding-component="false">
    <jbi:provides interface-name="petals:SampleBonita"
                  service-name="petals:SampleBonitaService"
                  endpoint-name="SampleBonitaEndpoint">
      <petals:wsdl>samplebonita.wsdl</petals:wsdl>
      <petals:params>
        <keyvalue:param>
          <petals:param name="rmi-url">rmi://127.0.0.1:1099</petals:param>
          <petals:param name="loginContextName">bonita</petals:param>

          <petals:param name="login">bsoa</petals:param>
          <petals:param name="password">bsoa</petals:param>

          <petals:param name="login2">admin</petals:param>
          <petals:param name="password2">admin</petals:param>
        </keyvalue:param>
      </petals:params>
    </jbi:provides>
  </jbi:services>
</jbi:jbi>
```

Sample of `jbi.xml` file with login/role authentication:

```
<?xml version="1.0" encoding="UTF-8"?>
<jbi:jbi version="1.0" xmlns:jbi="http://java.sun.com/xml/ns/jbi"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:petals="http://petals.ow2.org/extensions"
  xmlns:keyvalue="http://petals.ow2.org/extensions/key-value/">
  <jbi:services binding-component="false">
    <jbi:provides interface-name="petals:SampleBonita"
                  service-name="petals:SampleBonitaService"
                  endpoint-name="SampleBonitaEndpoint">
      <petals:wsdl>samplebonita.wsdl</petals:wsdl>
      <petals:params>
        <keyvalue:param>
          <petals:param name="rmi-url">rmi://127.0.0.1:1099</petals:param>
          <petals:param name="loginContextName">bonita-petals</petals:param>
          <petals:param name="login">bsoa</petals:param>
          <petals:param name="role">BONITAUSER</petals:param>

          <petals:param name="login2">admin</petals:param>
          <petals:param name="role2">BONITAUSER</petals:param>
        </keyvalue:param>
      </petals:params>
    </jbi:provides>
  </jbi:services>
</jbi:jbi>
```

Chapter 3. Security

In order to access Bonita, the engine needs two security files:

- `{user.home}/petals-bonita-engine_default_java.login.config`
- `{user.home}/petals-bonita-engine_default_java.policy`

If these files do not exist, the engine copies in the user home the default files from the engine jar. If you want to customize these files, put your customized files in the user home, with the same filenames as above.

For the propagation context (login/role authentication), you have to add a login context in the file `$JONAS_BASE/conf/jaas.config`:

```
bonita-petals {
    org.ow2.petals.se.bonita.security.PetalsSeBonitaDefaultLoginModule required
    ;
    // Use the login module to propagate security to the JOnAS server
    org.objectweb.jonas.security.auth.spi.ClientLoginModule  required
    ;
};
```

The propagation must be activated in the file `$JONAS_BASE/conf/jonas.properties`:

```
jonas.security.propagation              true
jonas.transaction.propagation           true
```

No security manager is set in the Java code (there is no `System.setSecurityManager(new RMISecurityManager());`). If you have errors because Petals try to download code from remote location, you must add a security manager in the Petals startup command line:

```
java -Djava.security.manager=java.rmi.RMISecurityManager -jar server.jar
```

See the file `%PETALS_HOME%/bin/startup.bat` or `$PETALS_HOME/bin/startup.sh`.