REST API Documentation

The REST API allows you to access AnthillPro's web services as resources, via their URLs, using standard HTTP protocol. The responses to most of the GET methods (except property values, which are simple text) are in XML, and may be consumed by any application, etc., that is able to process XML files. Currently, REST services are available for the following AnthillPro resources:

- **Agent Properties.** List the current properties of configured AnthillPro agents.

  [Click here for Agent Properties methods]

- **Build Life.** List the current artifacts, dependencies, issues, properties, source analytics, source changes, test coverage, or tests of a particular Build Life.

  [Click here for Build Life methods]

- **Folder Contents.** List the current contents of folders on the Administration page.

  [Click here for Folder Contents methods]

- **Jobs.** List the properties or a property value of a job, based on the `job_trace_id` number.

  [Click here for Jobs methods]

- **Project Originating Workflows.** Lists all the originating (build) workflow for a given project, including any branches, etc.

  [Click here for Project Originating Workflows methods]

- **Request Properties.** List all of the Workflow Request properties for a particular request.

  [Click here for Request Properties methods]

- **System Properties.** List the AnthillPro System Properties.

  [Click here for System Properties methods]

- **Workflow Build Lives.** List all the Build Lives associated with an originating workflow.

  [Click here for Workflow Build Lives methods]

Agent Properties

List the current properties of configured AnthillPro agents. When an agent connects to the system, it is assigned a number: e.g. "4". You can view the agent ID by either hovering over the link in the UI or in the agent's address bar (go to **System > Agents**). You can list the properties of ignored agents, as long as they are still configured. If an agent is renamed, the ID will remain the same.

List All Properties for an Agent

List all the properties for a given agent. This includes all the locked properties set by the system (Locked, Anthill, System, Environment), as well as properties you have set on the agent.

URL:

/rest/agent/{agentId}/properties

Argument:

`agentId` -- The ID of the agent you wish to list properties for.

Usage Example:

http://localhost:8080/rest/agent/13/properties

Example XML Response:

```
<properties>
  <property name="User_Defined_Property_01"
   secure="false">User_Defined_Property_01_Value</property>
  <property name="User_Defined_Property_02"
   secure="true">pbe{DmfOa/VLClhObUtcC1r9Z6Y4kygmU88A}</property>
  <property name="anthill3/java.home"
   secure="false">C:\Program Files (x86)\Java\jdk1.6.0_16</property>
  <property name="env/AGENT_HOME"
   secure="false">C:\anthill-agents\build</property>
  <property name="locked/agent.id"
   secure="false">cG14MnczMWxrbGpjejd4ZTcwMj</property>
  <property name="locked/agent.name" secure="false">build</property>
  <property name="sys/agent.log.to.console" secure="false">y</property>
</properties>
```

List Value of a Single Agent Property

List the value of any property for a given agent. This includes all the locked properties set by the system (Locked, Anthill, System, Environment), as well as properties you have set on the agent.

URL:

/rest/agent/{agentId}/properties/{propName}

Arguments:

`agentId` -- The ID of the agent you wish to list properties for.

`propName` -- The exact name of the property you want to get the value for.

Usage Example:

http://localhost:8080/rest/agent/13/properties/User_Defined_Property_01

Example Response:

```
User_Defined_Property_01_Value
```

Build Life

List the current artifacts, dependencies, issues, properties, source analytics, source changes, test coverage, or tests of a particular Build Life. When a Build Life is created, it is assigned a number: e.g. "4". The ID corresponds to the Build Life number listed on the Dashboard.

List Artifacts for a Build Life

List all the artifacts for a given Build Life. All artifact sets for the Build Life are listed, including the files within each artifact set.

URL:

/rest/buildlife/{buildLifeId}/artifacts

Argument:

`buildLifeId` -- The ID of the Build Life you wish to list artifacts for.

Usage Example:

http://localhost:8080/rest/buildlife/1001/artifacts

Example XML Response:

```
<artifactSets buildlifeId="1001">
  <artifactSet name="PDF" id="69">
    <artifact filename=".ahs.manifest" path=".ahs.manifest"
     length="1818" lastModifiedDate="1277744033000" hash=""/>
    <artifact filename=".ahs.dig" path=".ahs.dig" length="974"
     lastModifiedDate="1277744033000" hash=""/>
    <artifact filename="Document.pdf" path="Document.pdf"
     length="6268907" lastModifiedDate="1277744033000"
     hash="SHA-1{90613e681aff39864894500db721200fc979aeb8}"/>
  </artifactSet>

  <artifactSet name="Online Content" id="68">
    <artifact filename="content.zip" path="content.zip" length="5221996"
     lastModifiedDate="1277744034000"
     hash="SHA-1{8a6ba3907eae70c108dc2baf2c776cca8b263bfd}"/>
    <artifact filename=".ahs.manifest" path=".ahs.manifest"
     length="159" lastModifiedDate="1277744034000" hash=""/>
    <artifact filename=".ahs.dig" path=".ahs.dig" length="136"
     lastModifiedDate="1277744034000" hash=""/>
  </artifactSet>
</artifactSets>
```

List Dependencies for a Build Life

List all the dependencies for a given Build Life. Each dependency listed includes the project, workflow, Build Life, and artifact set used in the dependency relationship.

URL:

/rest/buildlife/{buildLifeId}/dependencyplan

Argument:

`buildLifeId` -- The ID of the Build Life you wish to list dependencies for.

Usage Example:

http://localhost:8080/rest/buildlife/1001/dependencyplan

Example XML Response:

```
<project profileId="131" workflowId="191" workflow="build" projectId="151"
 name="Project Name">
  <dependencies>
    <dependency>
      <ah-project profileId="61" workflowId="111" workflow="build"
       projectId="71" name="Project 2"/>
      <criteria build-life-id="1002"/>
      <delivery transitive="false" artifact-set="Default">
        <directory>lib/build</directory>
      </delivery>
      <delivery transitive="false" artifact-set="lib">
        <directory>lib/</directory>
      </delivery>
    </dependency>
    <dependency>
      <cs-project projectId="137" name="Project 3"/>
      <criteria build-life-id="237"/>
      <delivery transitive="false" artifact-set="Default">
        <directory>download</directory>
      </delivery>
    </dependency>
    <dependency>
      <cs-project projectId="260" name="Project 4"/>
      <criteria build-life-id="377"/>
      <delivery transitive="false" artifact-set="Default">
        <directory>lib</directory>
      </delivery>
    </dependency>
  </dependencies>
</project>
```

List Issues for a Build Life

List all the issues associated for a given Build Life. If you are using one of the issue-tracking integrations, any issues (which appear on the Build Life Issues tab) associated with the Build Life will be listed.

URL:

/rest/buildlife/{buildLifeId}/issues

Argument:

`buildLifeId` -- The ID of the Build Life you wish to list issues for.

Usage Example:

http://localhost:8080/rest/buildlife/1001/issues

Example XML Response:

```
<issues date="Mon Jun 28 16:45:03 EDT 2010">
  <issue id="Issue-101" issue-tracker="JIRA">
    <name>
      Issue Name
    </name>
  <type>Bug</type>
  <status>Open</status>
    <description>
      Detailed issue description.
    </description>
  </issue>
```

```
</issues>
```

List Properties for a Build Life

List all the properties associated for a given Build Life. These properties are set during run-time, and are included in your build job.

URL:

/rest/buildlife/{buildLifeId}/properties

Argument:

`buildLifeId` -- The ID of the Build Life you wish to list properties for.

Usage Example:

http://localhost:8080/rest/buildlife/1001/properties

Example XML Response:

```
<properties>
  <property name="Build_Life_Property_1"
   secure="false">Build_Life_Property_1_Value</property>
  <property name="Build_Life_Property_2"
   secure="false">Build_Life_Property_2_Value</property>
</properties>
```

List Property Value for a Build Life

List the value of a Build Life property. The property is set during run-time, and is included in your build job.

URL:

/rest/buildlife/{buildLifeId}/properties/{propName}

Arguments:

`buildLifeId` -- The ID of the Build Life you wish to list dependencies for.

`propName` -- The name of the Build Life property you wish to list the value of.

Usage Example:

http://localhost:8080/rest/buildlife/1001/properties/Build_Life_Property_1

Example Response:

```
Build_Life_Property_1_Value
```

List Source-analytics Test Report for a Build Life

List the source-analytics test report, including the tests run, for a particular Build Life. If you are using one of the source-analytics testing integrations, the test report (which appear on the Build Life Analytics tab) associated with the Build Life will be listed.

URL:

/rest/buildlife/{buildLifeId}/sourceanalytics/{reportName}

or

rest/buildlife/{buildLifeId}/sourceanalytics?reportName={reportName}

Arguments:

`buildLifeId` -- The ID of the Build Life you wish to get the test report for.

`reportName` -- The name of the test-analytics Test Report associated with the Build Life.

Usage Example:

http://localhost:8080/rest/buildlife/1001/sourceanalytics/PMD

or

http://localhost:8080/rest/buildlife/1001/sourceanalytics?reportName=PMD

Example XML Response:

```
<analytics type="PMD" name="PMD" buildLifeId="1001">
  <finding>
    <id/>
    <file>
    project-servlet\java\project\domain\File.java
    </file>
    <line>21</line>
    <name>VariableNamingConventions</name>
    <severity>Naming Rules</severity>
  </finding>
</analytics>
```

List Source Changes for a Build Life

List the source changes that went into a particular Build Life. If multiple changes were combined in the build (for example, two developers committed changes to the Build Life), each one will be listed as a separate <change-set>.

URL:

/rest/buildlife/{buildLifeId}/sourcechanges

Argument:

`buildLifeId` -- The ID of the Build Life you wish to list source changes for.

Usage Example:

http://localhost:8080/rest/buildlife/1001/sourcechanges

Example XML Response:

```
<change-log>
  <log-info>
    <build-life>1001</build-life>
  </log-info>
  <change-set>
    <repository-type>Subversion</repository-type>
    <repository-id>101</repository-id>
    <anthill-id>20002</anthill-id>
    <id>30003</id>
    <user>User Name</user>
    <date>2010-06-28 12:19:14.0 -0400</date>
      <file-set>
         path/to/the/file/set/source/file.java
        </file>
      </file-set>
      <comment>
      Commit comment made by developer who made change.
      </comment>
  </change-set>
</change-log>
```

List Coverage Test Report for a Build Life

List the coverage test report, including the coverage percentage and complexity for each group, for a particular Build Life. If you are using one of the coverage testing integrations, the test report (which appear on the Build Life Coverage tab) associated with the Build Life will be listed.

URL:

/rest/buildlife/{buildLifeId}/testcoverage/{reportName}

or

rest/buildlife/{buildLifeId}/testcoverage?reportName={reportName}

Arguments:

`buildLifeId` -- The ID of the Build Life you wish to get the test report for.

`reportName` -- The name of the coverage Test Report associated with the Build Life.

Usage Example:

http://localhost:8080/rest/buildlife/1001/testcoverage/Coverage

or

http://localhost:8080/rest/buildlife/1001/testcoverage?reportName=Coverage

Example XML Response:

```
<coverage-report name="Coverage" type="cobertura" job-id="417710"
 line-percentage="0.0641" branch-percentage="0.0561">
  <coverage-group name="com.yourcompany.product" line-percentage="0.25"
   branch-percentage="1.0" complexity="1.0"/>
  <coverage-group name="com.yourcompany.product.command" line-percentage="0.6047"
   branch-percentage="0.8889" complexity="1.7333"/>
  <coverage-group name="com.yourcompany.product.command.component"
   line-percentage="0.8354" branch-percentage="0.8" complexity="2.0"/>
</coverage-report>
```

List Test Report for a Build Life

List the test report, including the tests run, for a particular Build Life. If you are using one of the testing integrations, the test report (which appear on the Build Life Tests tab) associated with the Build Life will be listed.

URL:

/rest/buildlife/{buildLifeId}/tests/{reportName}

or

rest/buildlife/{buildLifeId}/tests?reportName={reportName}

Arguments:

`buildLifeId` -- The ID of the Build Life you wish to get the test report for.

`reportName` -- The name of the Test Report associated with the Build Life.

Usage Example:

http://localhost:8080/rest/buildlife/1001/tests/Unit Tests

or

http://localhost:8080/rest/buildlife/1001/tests?reportName=Unit Tests

Example XML Response:

```
<test-report name="Unit Tests" type="JUnit" job-id="417432" successes="343"
 failures="0">
  <test-suite name="XMLMarshaller" successes="1" failures="0">
    <test name="testMarshal" class-name="class.name.being.tested.XMLMarshaller"
     result="success" time="26">
    </test>
  </test-suite>
</test-report>
```

Folder Contents

List the current contents of folders on the Administration page. When a folder is created, it is assigned a number: e.g. "4". You can view the folder ID by either hovering over the link in the UI or in the folder's address bar. You can list the contents of both active and inactive folders. If a folder is moved and/or renamed, the ID will remain the same. Note that any folders pre-installed in AnthillPro have an ID that begins with the negative sign (-): e.g., the root Administration folder (which can't be deleted) will always have an ID of "-1". By looking up the root folder, you can quickly view the ID of folders or projects.

List Folder Contents

List all the subfolders and projects within the specified folder.

URL:

/rest/folder/{folderId}/contents

Argument:

`folderId` -- The ID of the folder you wish to list.

Usage Example:

http://localhost:8080/rest/folder/-1/contents

Example XML Response:

```
<folder name="/" id="-1" description="Root Folder of Project Tree">
  <subfolders>
    <folder name="Codestation Projects" id="-4"
     description="Default folder for Codestation Projects"/>
    <folder name="Example Projects" id="-5"
     description="Pre-installed example projects"/>
    <folder name="Job Library" id="-3"
     description="Default folder for reusable jobs"/>
    <folder name="Workflow Library" id="-2"
     description="Default folder for reusable workflows"/>
  </subfolders>
  <projects>
    <project name="F5 Test" id="51"
     description="Used for testing the integration."/>
    <project name="maven" id="22" description=""/>
    <project name="TeamForge Test" id="41"
     description="Test development of the CollabNet TeamForge plugin."/>
  </projects>
</folder>
```

Jobs

List the properties or a property value of a job, based on the `job_trace_id` number. To find the `job_trace_id` number, you can hover over the View Job icon or go to the job's Dashboard page URL. Job-step properties are only available for steps that use one of the AnthillPro Plugins -- no other job step in AnthillPro has properties associated with it.

List Job Properties

List all the runtime properties of a job. The list includes the working directory path and workspace date and version, etc.

URL:

/rest/job/{jobId}/properties

Argument:

`jobId` -- The ID of the job at runtime (the `job_trace_id` number).

Usage Example:

http://localhost:8080/rest/job/26/properties

Example XML Response:

```
<properties>
  <property name="work.dir.path" secure="false">
  C:\anthill-agents\build\var\jobs/projects/Cvs_Anthill_Example/Trunk_Build_Workflow
  </property>
  <property name="workspace.date"
   secure="false">2010-06-29 11:56:22.608 AM EDT</property>
  <property name="workspace.version" secure="false">1.0.4</property>
```

```
</properties>
```

List Job Property Value

List the value of a single runtime job property. The list includes the working directory path and workspace date and version, etc.

URL:

/rest/job/{jobId}/properties/{propName}

Arguments:

`jobId` -- The ID of the job at runtime (the `job_trace_id` number).

`propName` -- The ID of the job property for which the value is to be listed.

Usage Example:

http://localhost:8080/rest/job/26/properties/work.dir.path

Example Response:

```
C:\anthill-agents\build\var\jobs/projects/Cvs_Anthill_Example/Trunk_Build_Workflow
```

List Job-Step Properties

List all the runtime properties of a job step that uses an AnthillPro Plugin. Currently, only steps that are part of an AnthillPro Plugin will have properties, so only use this method if you want to get the properties for a job step that is included as part of an AnthillPro Plugin (go to **System** > **Plugins** to see which Plugins you have installed).

URL:

/rest/job/{jobId}/step/{stepIndex}/properties

Argument:

`jobId` -- The ID of the job at runtime (the `job_trace_id` number).

`stepIndex` -- The ID of the job step at runtime. The job-steps run in order, starting with 0 (zero). So, when looking at the job on the Dashboard, job-step 1 would have the ID of 0; job-step 2 would have the ID of 1; job-step 3 would have the ID of 1; etc.

Usage Example:

http://localhost:8080/rest/job/1001/step/3/properties (The {stepIndex} of 3 is the ID of job-step 4 in the UI.)

Example XML Response:

```
<properties>
  <property name="tomcatContext" secure="false">/</property>
  <property name="tomcatManagerUrl"
   secure="false">http://tomcat.manager.url.com:9080/manager</property>
  <property name="tomcatPassword"
   secure="true">pbe{1Gv1SZNOZ2U2YbF4WGmMYgFBTXWPSVOXl5BX0eFvs50=}</property>
  <property name="tomcatUsername" secure="false">tomcatusername</property>
  <property name="warFile" secure="false">web-app.war</property>
```

```
</properties>
```

List Job-Step Property Value

List the runtime property value of a job step that uses an AnthillPro Plugin. Currently, only steps that are part of an AnthillPro Plugin will have properties (and values), so only use this method if you want to get the properties for a job step that uses an AnthillPro Plugin (go to **System > Plugins** to see which Plugins you have installed).

URL:

/rest/job/{jobId}/step/{stepIndex}/properties/{propName}

Arguments:

`jobId` -- The ID of the job at runtime (the `job_trace_id` number).

`stepIndex` -- The ID of the job step at runtime. The job-steps run in order, starting with 0 (zero). So, when looking at the job on the Dashboard, job-step 1 would have the ID of 0; job-step 2 would have the ID of 1; job-step 3 would have the ID of 1; etc.

`propName` -- The name of the step property for which you want to return a value.

Usage Example (The {stepIndex} of 3 is the ID of job-step 4 in the UI.):

http://localhost:8080/rest/job/1001/step/3/properties/tomcatPassword

Example Response:

```
pbe{jEpVWNJvmW9cbtlVRVnBTVn//3Q8XKoTmxTnDqBeaMI=}
```

List Job Working Directory

List the working directory of a job. The list includes the working directory path and workspace date and version, etc.

URL:

/rest/job/{jobId}/workdir

Argument:

`jobId` -- The ID of the job at runtime (the `job_trace_id` number).

Usage Example:

http://localhost:8080/rest/job/26/workdir

Example XML Response:

```
<work-dir scope="job">
C:\anthill-agents\build\var\jobs/projects/Cvs_Anthill_Example/Trunk_Build_Workflow
</work-dir>
```

Project Originating Workflows

List all the originating (build) workflow for a given project, including any branches, etc. When a workflow is created, it is assigned a number: e.g. "4". You can view the folder ID by either hovering over the link in the UI (on the Administration page) or in the project's address bar. If a project is moved and/or renamed, or if one of the workflows is renamed, the IDs will all remain the same.

List Project Originating Workflows

List all the configuring originating workflows for a project. This includes any branches, etc. However, any secondary workflow (e.g., a deploy workflow) will not be listed.

URL:

/rest/project/{projectId}/workflows

Argument:

`projectId` -- The ID of the project you wish to list originating workflows for.

Usage Example:

http://localhost:8080/rest/project/11/workflows

Example XML Response:

```
<project name="Cvs Anthill-Example" id="11"
 description="An example project with a basic configuration.">
  <originatingWorkflow name="Trunk Build Workflow" id="41"
   description="A sample workflow"/>
  <originatingWorkflow name="Trunk Build Workflow [Branch]" id="83"
   description="A sample workflow that builds a branch."/>
</project>
```

Request Properties

List all of the Workflow Request properties for a particular request. When a request is created, it is assigned a number: e.g. "4". You can view the request ID in the UI (go to a Build Life and hover over the View Request link) or go to the Build Request page. Request properties are configured on the workflow (go to **Administration** > **Workflow** > **Properties** tab for an example).

List Request Properties

List all of the workflow request properties associated with a Build Life request.

URL:

/rest/request/{requestId}/properties

Argument:

`requestId` -- The ID of the request you wish to list the properties of.

Usage Example:

http://localhost:8080/rest/request/15/properties

Example XML Response:

```
<properties>
```

```
  <property name="Workflow_Request_Property_1"
   secure="false">Workflow_Request_Property_1_Value</property>
  <property name="Workflow_Request_Property_2" secure="false">true</property>
</properties>
```

List Request Property Value

List the value of a single workflow request property associated with a Build Life request.

URL:

/rest/request/{requestId}/properties/{propName}

Arguments:

`requestId` -- The ID of the request you wish to list the property value of.

`propName` -- The ID of the request property you wish to list the value of.

Usage Example:

http://localhost:8080/rest/request/15/properties/Workflow_Request_Property_1

Example Response:

```
Workflow_Request_Property_1_Value
```

System Properties

List the System Properties configured on the **System** > **Server** > **Properties** page. For secure properties, the value is obfuscated.

List System Properties

List all the system properties. The lists includes all property names and values. For secured properties, the value is obfuscated.

URL:

/rest/system/properties

Usage Example:

http://localhost:8080/rest/system/properties

Example XML Response:

```
<properties>
  <property name="system_property" secure="false">system_property_value</property>
  <property name="system_property_secure"
   secure="true">pbe{gGUYb2Fg3XRfdJlr42ufX63pr++9CRGc}</property>
</properties>
```

List System Property Value

List the value of a single System Properties. The list returns the value of the specified property.

URL:

/rest/system/properties/{propName}

Argument:

`propName` -- The name of the system property for which you want to list the value of.

Usage Example:

http://localhost:8080/rest/system/properties/system_property

Example Response:

```
system_property_value
```

Workflow Build Lives

List all of the Build Lives created for an originating workflow. When a workflow is created, it is assigned a number: e.g. "4". You can view the workflow ID in the UI (hover over the workflow name) or go to the workflow Administration page navigation bar.

List Workflow Build Lives

List all the Build Lives associated with an originating workflow. The lists includes the ID, stamps and statuses of the Build Lives. The Build Lives are ordered from most recent to oldest. Deactivated Build Lives will be listed along with active Build Lives.

URL:

/rest/workflow/{workflowId}/buildlives

Argument:

`workflowId` -- The ID of the originating workflow you wish to list the Build Lives of.

Usage Example:

http://localhost:8080/rest/workflow/41/buildlives

Example XML Response:

```
<buildlives workflowName="Trunk Build Workflow" workflowId="41"
 description="A sample workflow">
  <buildlife id="7">
    <statuses>
      <status status="success" date="1277827003140"/>
    </statuses>
    <stamps>
      <status stamp="1.0.4" date="1277826994848"/>
    </stamps>
  </buildlife>
  <buildlife id="6">
    <statuses>
      <status status="success" date="1277826327429"/>
    </statuses>
    <stamps>
      <status stamp="1.0.3" date="1277826320785"/>
    </stamps>
  </buildlife>
</buildlives>
```