*-PaaS API specification

Applications, Environments, monitoring & logging Management

Sami Yangui, Mohamed Sellami, Mohamed Mohamed, Samir Tata

09/11/2012
Version 1.4.2

Table of contents

Part 1: Introduction

This document provides a description of the generic PaaS API REST/JSON based.
It describes the allowed actions for both User and Admin actions.

The following actions of the REST protocol are used:

REST	**CRUD**
POST	Create
GET	Read
PUT	Create or Update
DELETE	Delete

Part 2: User API

1. API authentication

The API authentication system is "Basic". It is essentially based on adding a field "Authorization" in the header of requests containing the encoded string *user: password*. If the request does not contain this entry, an error "*401: Unauthorized*" is returned.

2. Common parameters

This section describes all the request parameters that all actions use.

Parameter	Description	Required
Action	The action to perform	Yes
Format	The format of the result. Currently, only JSON is supported. Default: JSON	No
AccessKeyId	*The key id used to sign the request (user name)*	
Signature	*The digital signature you created for the request.*	
SignatureMethod	*The algorithm used to sign the request.*	
Timestamp	The time at which the query was generated, formatted as the number of seconds since Jan 1, 1970 UTC.	No
Version	The API version to use. Expected value: 0.1.	Yes

3. Common errors

This section lists common errors that all action returns.

Error	Description	HTTP Status Code
Invalid signature	The request signature is not valid	400
Internal Failure	The internal processing has failed due to some unexpected errors.	500
Invalid Action	The action is invalid	400
InvalidKeyId	The key id is invalid	403
InvalidParameter	The parameter is not expected	400
InvalidValue	The value is not expected	400
MissingAuthentication	The key id is missing	403
MissingParameter	The parameter is missing	400
RequestExpired	The request expires	400
ServiceUnavailable	The request has failed due to a temporary failure on the server	503

4. HEAD command

The HEAD command provides a kind of contextual help and returns in the header a list of accessible uri.

5. API objects description

All API objects are described using WADL specification (See http://www.w3.org/Submission/wadl/).

WADL descriptions are automatically generated with Jersey using JAX-RS annotations. The WADL is recoverable by a GET request on the url *http:// <host>: <port> / paas-api/paas-api.wadl*.

An example is provided in Annex 1.

6. API objects references

We use Links and Link Relations concepts in responses to explore API objects and to navigate between them. These concepts are inspired from the vCloud project (See http://www.vmware.com/pdf/vcd_15_admin_guide.pdf).

Link elements provide references to objects and the actions that they support. These elements are the primary mechanisms by which a server tells a client how to access and operate on an object. The server creates Link elements in a response body. They are read-only for the client.

Each Link element has the following form:

```
<Link rel="relationship"
type="application/jpaas.object_type+xml"
href="URL"
name="string"/>
```

Attribute	detail
rel	Defines the relationship of the link to the object that contains it. A relationship can be the name of an operation on the object, a reference to a contained or containing object, or a reference to an alternate representation of the object. The relationship value implies the HTTP verb to use when you use the link's href value as a request URL.
type	The object type, specified as a MIME content type, of the object that the link references. This attribute is present only for links to objects. It is not present for links to actions.
href	An object reference, expressed in URL format. Because this URL includes the object identifier portion of the id attribute value, it uniquely identifies the object, persists for the life of the object, and is never reused. The value of the href attribute is a reference to a view of the object, and can be used to access a representation of the object that is valid in a particular context. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat each href as an opaque string. The rules that govern how the server constructs href strings might change in future releases.
name	The name of the referenced object, taken from the value of that object's name attribute. Action links do not include a name attribute.

For more details, the list of possible values of *rel* attribute is detailed in vCloud admin guide (See http://www.vmware.com/pdf/vcd_15_admin_guide.pdf).

7. Application Management Operations

Description	Command	Specific parameters	Response	Return value
Start an application version instance	POST /app/{appId}/version/{versionId}/instance/{instanceId}/action/start			200 OK
Stop an application version instance	POST /app/{appId}/version/{versionId}/instance/{instanceId}/action/stop			200 OK
Creates a new application. If the application is multitenant, it will be accessible for all tenant An Application Descriptor must be provided.	POST /app	-Application name - Description -Multi-tenant (yes/no).	ApplicationDescription - with appId -with Link element	
Creates a new version with either a file in attachment (Content-Type: multipart/form-data). or a url. The supported artefacts are ear, bundle, war, ejbjar or a zip (war dir). An Application Version Descriptor must be provided.	POST /app/{appId}/version	- Artefact - Description - VersionLabel	ApplicationVersionDescription - with versioned -with Link element	
Creates a new application version instance. An Application Version Instance Descriptor must be provided.	POST /app/{appId}/version/{versionId}/instance	- Deployable topology mapping - Environment uri	ApplicationVersionInstanceDescription - with instanceid -with Link element	
Start an application version instance	POST /app/{appId}/version/{versionId}/instance/{instanceId}/action/start		ApplicationVersionInstanceDescription	
Stop an application version instance	POST /app/{appId}/version/{versionId}/instance/{instanceId}/action/stop		ApplicationVersionInstanceDescription	
List applications	GET /app/			Application Description List
List application versions	GET /app/{appId}/version			Application VersionDescriptionList
List application version instances	GET /app/{appId}/version/{versionId}			Application VersionInstanceDescriptionList
List application version artefacts	GET /artefact/{appId}/{versionId}			Application VersionArtefactsList
Describe application.	GET /app/{appId}			Application Description
Describe application version	GET /app/{appId}/version/{versionId}			Application VersionDescription

Describe application version instance	GET /app/{appId}/version/{versionId}/instance/{instanceId}			ApplicationVersionInstanceDescription
Delete application. Removes all existing versions	DELETE /app/{appId}			
Delete application version	DELETE /app/{appId}/version/{versionId}			
Delete application version instance	DELETE /app/{appId}/version/{versionId}/instance/{instanceId}			

8. Environment Management Operations

Description	**Command**	**Specific parameters**	**Response**	**Return value**
Starts an environment	POST /environment/{envId}/action/start			200 OK
Stops an environment	POST /environment/{envId}/action/stop			200 OK
Deploy an application instance on an available environment	POST /environment/{envId}/action/deploy/app/{appId}/version/{versionId}/instance/{instanceId}			200 OK
Undeploy an application instance on an available environment	DELETE /environment/{envId}/action/undeploy/app/{appId}/version/{versionId}/instance/{instanceId}			200 OK
Creates a new environment. An environment template descriptor must be provided.	POST /environment/		EnvironmentDescription. -with envId -with Link element	
Deletes an environment	DELETE /environment/{envId}			
List the available environments	GET /environment	Optional : name		EnvironmentDescriptionList.
Starts an environment	POST /environment/{envId}/action/start			
Stops an environment	POST /environment/{envId}/action/stop			
Deploy an application instance on an available environment	POST /environment/{envId}/action/deploy/app/{appId}/version/{versionId}/instance/{instanceId}			
Undeploy an application instance on an available environment	DELETE /environment/{envId}/action/undeploy/app/{appId}/version/{versionId}/instance/{instanceId}			
Get the description of an environment	GET /environment/{envid}			EnvironmentDescription.
List the deployed application instances in an environment	GET /environment/{envId}/app/			ApplicationVersionInstanceDescriptionList.

9. Monitoring & Logging Management Operations

Description	Command	Specific parameters	Response	Return value
Describes Events Platform related events.	GET /event GET /app/{appId}/event GET /app/{appId}/version/{versionId}/event GET /environment/{env}/event	Optional parameters : - start time - next token (pagination)	Events	
Download events file The first operation gets the files and the second one downloads the file.	GET /event/file/ GET /event/file/{fileId}			
Describes logs Application related logs	GET /log GET /app/{appId}/log GET /app/{appId}/version/{versionId}/log GET /environment/{envId}/log	Optional parameters : - start time - next token (pagination)	Logs	
Download logs file The first operation gets the files and the second one downloads the file.	GET /log/file/ GET /log/file/{fileId/}			
Describes stats	GET / stat GET /app/{appId}/ stat GET /app/{appId}/version/{versionId}/stat GET /environment/{envId}/ stat	Optional parameters : - start time - next token (pagination)	Stats	

Part3: Admin API

The admin API is accessible through JMX and requires admin rights.

1. User Management API

Description	Operation	Specific parameters	Response	Return value
User creation.	createUser	Quota (max number of apps)	User (with a secret)	User
User deletion.	deleteUser	UserId		
Get users list.	listUsers		Array of user with quota information	
Get user	getUser	UserId		User

2. Platform Management API (In progress)

Description	Operation	Specific parameters	Response	Return value
List applications	listApplications	Optional : - user - appName - version - multitenantOnly		
List environments	listEnvironments	Optional : - user - name		
Get events	getEvents	Optional : - user - app - env		
Get logs	getLogs	Optional : - user - app - env		
Start PaaS				
Stop PaaS				
Deploy PaaS				
Provision PaaS				

Annex 1: WADL description

```xml
<application xmlns="http://research.sun.com/wadl/2006/10">
  <doc xmlns:jersey="http://jersey.java.net/" jersey:generatedBy="Jersey: 1.8 06/24/20
  12:17 PM"/>
  <resources base="http://127.0.0.1:8080/CF-api/rest/">
    <resource path="app">
      <method id="createApplication" name="POST">
        <request>
          <representation mediaType="application/xml"/>
        </request>
        <response>
          <representation mediaType="application/xml"/>
        </response>
      </method>
      <method id="findApplications" name="GET">
        <response>
          <representation mediaType="*/*"/>
        </response>
      </method>
      <resource path="{appId}/version/{versionId}">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="appId" style="template
        type="xs:string"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="versionId"
        style="template" type="xs:string"/>
        <method id="findApplicationVersionInstances" name="GET">
          <response>
            <representation mediaType="*/*"/>
          </response>
        </method>
      </resource>
      <resource path="{appId}/version/{versionId}/instance">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="appId" style="template
        type="xs:string"/>
        <method id="createApplicationVersionInstance" name="POST">
          <request>
            <representation mediaType="application/xml"/>
          </request>
          <response>
            <representation mediaType="application/xml"/>
          </response>
        </method>
      </resource>
      <resource path="{appId}/version/{versionId}/instance/{instanceId}/action/start">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="appId" style="template
        type="xs:string"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="instanceId"
        style="template" type="xs:string"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="versionId"
        style="template" type="xs:string"/>
        <method id="startApplicationVersionInstance" name="POST">
          <response>
            <representation mediaType="*/*"/>
          </response>
        </method>
      </resource>
      <resource path="{appId}/version/{versionId}/instance/{instanceId}/action/stop">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="appId" style="template
        type="xs:string"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="instanceId"
        style="template" type="xs:string"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="versionId"
        style="template" type="xs:string"/>
        <method id="stopApplicationVersionInstance" name="POST">
          <response>
            <representation mediaType="*/*"/>
          </response>
        </method>
      </resource>
      <resource path="{appId}">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="appId" style="template"
        type="xs:string"/>
        <method id="describeApplication" name="GET">
          <response>
            <representation mediaType="application/xml"/>
          </response>
        </method>
      </resource>
      <resource path="{appId}/delete">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="appId" style="template"
        type="xs:string"/>
        <method id="deleteApplication" name="DELETE">
          <response>
            <representation mediaType="application/xml"/>
          </response>
        </method>
      </resource>
      <resource path="delete">
        <method id="deleteApplications" name="DELETE">
          <response>
            <representation mediaType="application/xml"/>
          </response>
        </method>
      </resource>
      <resource path="{appId}/version/create">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="appId" style="template"
        type="xs:string"/>
        <method id="createApplicationVersion" name="POST">
          <request>
            <representation mediaType="*/*"/>
          </request>
          <response>
            <representation mediaType="application/xml"/>
          </response>
        </method>
      </resource>
      <resource path="{appId}/version">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="appId" style="template"
        type="xs:string"/>
        <method id="findApplicationVersions" name="GET">
          <response>
            <representation mediaType="*/*"/>
          </response>
        </method>
      </resource>
    </resource>
    <resource path="environment">
      <method id="findEnvironments" name="GET">
        <response>
          <representation mediaType="application/xml"/>
        </response>
      </method>
      <method id="createEnvironment" name="POST">
        <request>
          <representation mediaType="application/xml"/>
        </request>
        <response>
          <representation mediaType="application/xml"/>
        </response>
      </method>
      <resource path="{envId}">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="envId" style="template"
        type="xs:string"/>
    <method id="getEnvironment" name="GET">
      <response>
        <representation mediaType="*/*"/>
      </response>
    </method>
    <method id="deleteEnvironment" name="DELETE">
      <response>
        <representation mediaType="*/*"/>
      </response>
    </method>
  </resource>
  <resource path="{envId}/app/">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="envId" style="template"
    type="xs:string"/>
    <method id="getDeployedApplicationVersionInstance" name="GET">
      <response>
        <representation mediaType="*/*"/>
      </response>
    </method>
  </resource>
  <resource path="{envId}/action/start">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="envId" style="template"
    type="xs:string"/>
    <method id="startEnvironment" name="POST">
      <response>
        <representation mediaType="*/*"/>
      </response>
    </method>
  </resource>
  <resource path="{envId}/action/stop">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="envId" style="template"
    type="xs:string"/>
    <method id="stopEnvironment" name="POST">
      <response>
        <representation mediaType="*/*"/>
      </response>
    </method>
  </resource>
  <resource
  path="{envId}/action/deploy/app/{appId}/version/{versionId}/instance/{instanceId}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="envId" style="template"
    type="xs:string"/>
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="appId" style="template"
    type="xs:string"/>
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="instanceId"
    style="template" type="xs:string"/>
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="versionId"
    style="template" type="xs:string"/>
    <method id="deployApplication" name="POST">
      <response>
        <representation mediaType="*/*"/>
      </response>
    </method>
  </resource>
  <resource
  path="{envId}/action/undeploy/app/{appId}/version/{versionId}/instance/{instanceId}"
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="envId" style="template"
    type="xs:string"/>
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="appId" style="template"
    type="xs:string"/>
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="instanceId"
    style="template" type="xs:string"/>
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="versionId"
    style="template" type="xs:string"/>
    <method id="undeployApplication" name="POST">
      <response>
        <representation mediaType="*/*"/>
```

```
          </response>
        </method>
      </resource>
    </resource>
  </resources>
</application>
```