CA Release Automation

RestFul API

Best Practices Guide

Author : Walter Guerrero

Version: 1.0

Filename: CA-Release-Automation-RestFul-API-Best-Practices-GuideV1.0.docx

Date: 12/6/2015

Table of Contents

Overview

As DevOps continues to grow and complexity of the systems being integrated from development into operations, it becomes necessary for applications to be accessed via REST APIs as desired by the customer base.

The REST capabilities offered by CA Release Automation revolve around configuration, execution, and monitoring of the ROC (Release Operation Center) activities, as well as the accompanying artifacts; without having to access the CA Release Automation ROC user interface.

Restful API Definition

The REST API for CA Release Automation is composed of GET/POST facilities. Up to the latest available release (5.5.2), there are four versions of the REST API; later on in this document we are going to provide a listing of the different REST API calls based off versions.

Accessibility of the REST API

The REST API can be easily accessed via the following links.

- Base URL
  - http://<localhost>:<port>/datamanagement/a/api/<version>
- Available documentation
  - https://wiki.ca.com/ca-release-automation/5-5-2/en/reference/rest-api-reference
  - http://<localhost>:<port>/datamanagement/restApi.jsp
- Where <localhost> → The hostname where the RA server is running
- Where <port> → The port number where the RA server is listening on
- Where <version> → The valid REST API version number: v1, v2, v3, v4
- If you don't provide a version number, the version number will default to v1

REST API Versions

The following table presents that available RESTful API version, up to release 5.5.1 of Release Automation.

Version	Description
V1	Created prior to release 4.7.0
V2	Created for releases 4.7.x+, this will present the commands containing /release*, and you will need to add /v2 after the base URL. The values available with this version are specific to status and stage

V3	Created for releases 5.0+ includes new operations
V4	Created for versions 5.5.2 and later, this version includes new operations on templates, template categories, and manual update approval
None	If you don't supply the API version, version V1 will be used by default

It is important that the correct versions of the RESTful API calls be used to accomplish the desired tasks at hand.

Calling the RESTful API

To successfully use the Release Automation RESTful API, the following pre-requisites are needed:

- HTTP basic authentication.
- Release Automation server running in HTTPS (this is the recommended mode, when using basic authentication) mode.
- Generate a Base64 string for the username/password combination. This can be generated by the postman application or you can use an online encoder/decoder to obtain this value.
- You can also use the Base64 API directly in your application.

Recommended Tools

To fully test the different RESTful API calls, the following tools can be used prior to the creation of the different packages that will be calling these RESTful API calls.

- Postman by postman running as a standalone Google Chrome application or plug-in.
- REST Easy extension in FireFox.
- HttpRequester extension in FireFox.
- Curl, which is a command-line utility used extensively to test the different HTTP get/post calls.
- SOAPIO 5.2.x or greater.

The recommendation is to use the Postman utility to conduct most of your testing of the Release Automation RESTful API that you would need to use. Not only will postman help you build the correct RESTful API call, but it will also build the "curl" options as well.

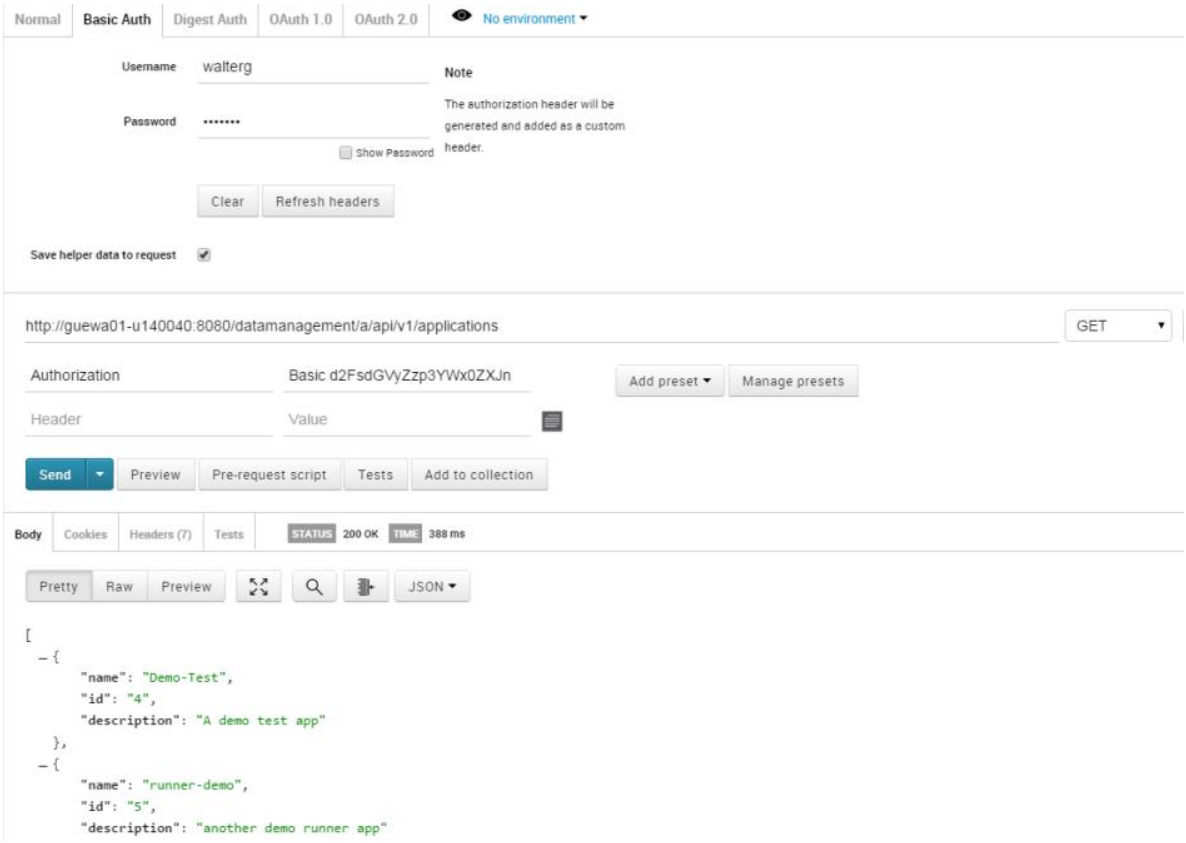


Figure 1: Typical Postman session

In the above figure, you see how the "/applications" call is built and the corresponding results in the postman session.

You will also need to take into account how the how the different parameters being used by the different GET/POST calls, these parameters can either be inline URL call, for example: /application?username=tester. The parameters can also be setup in a JSON body payload, for example {"username":"tester"}.

Here is another example where you can see how SOAPUI will need to be setup to test this Restful API call.

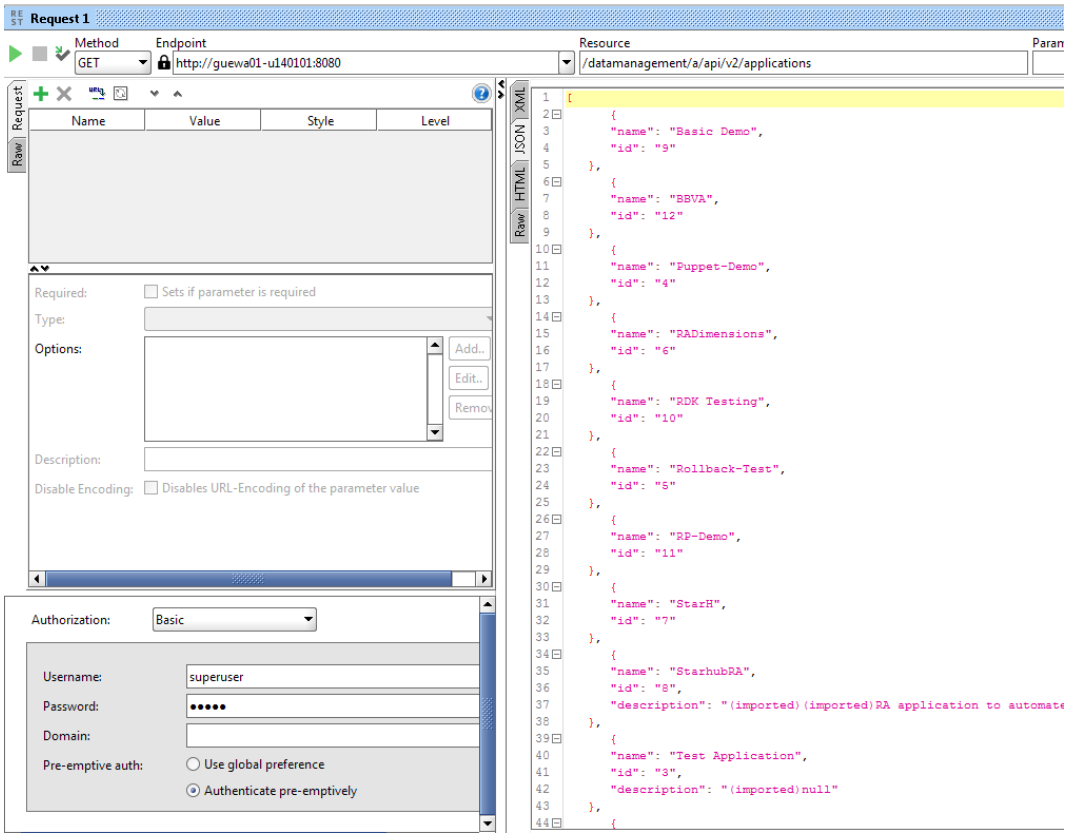


Figure 2: Typical SOAPUI Restful call

Best Practices

The following best practices will help you in improving your usage of the RESTful API calls by providing you with the most efficient methods to use.

Planning

As in any development project, it is important that you plan how the RESTful API calls are going to be used, and if you are planning in providing information from prior calls to other calls at runtime; therefore it becomes very important that you plan out the calls and the order to be followed and if the following calls use URL or JSON body parameters.

Tools

Prior to starting your development efforts to use the Release Automation RESTful API, it is recommended that you pick the right tool for the job to conduct your testing to see how the flow will work correctly. Our recommendation is that you use the “Postman” tool, since it provides you with a lot of flexibility and “curl” integration.

This also dictates if you need to use a BASE64 encoder externally or use the facilities provided by Postman.

Using the proper version of the API calls

Some of the API calls can be used as version 1 (v1) or version 2 (v2), and there are other calls that will provided you with different results based on the version being utilized (see the sections below for the correct description).

At the same time, you need to be aware of any API calls that have been deprecated, so you will have to use a new API call that would have been created for this scenario.

Using the correct parameter method

Given that some API calls use URL parameters and other calls are using JSON body parameters, it becomes very crucial that you become aware of the requirements for the API calls that you are going to be using. This is where the tools listed in this document can be of great help to you in determining how build the parameters require for a given API call.

Simple Use case

To help you in the understanding of how to use the Release Automation restful API functionality, we are going to create a very simple application that performs a series of delay actions lasting about 30 seconds each and adding a user input action to the flow and processes.

- Create a Release Automation application
- Setup the corresponding server types, architectures, and environments
- Now, we need to switch to the "Designer→Process Design" and add the following actions: delay, create empty file, add text to a file, and User Input – Stop for Manual Operation
- Create a flow and call it "Restful-Demo"

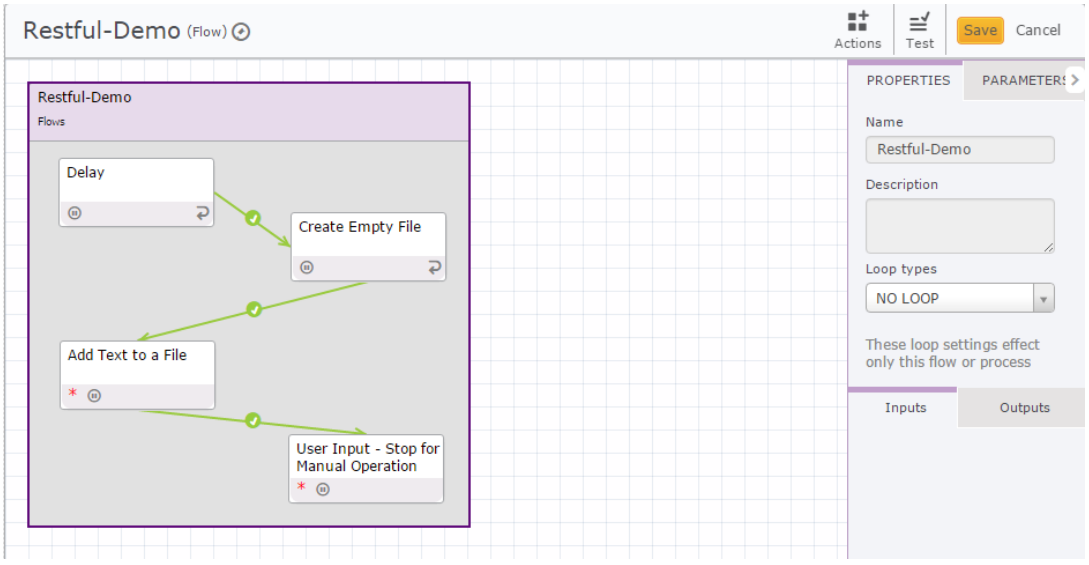


Figure 3: Restful-Demo flow

- The full path field in the “Create Empty File” action that is part of the flow needs to have the following entry “c:\temp\restful-demo.txt” and the “Fail If Exist” needs to be set to ‘false’.
- For the “Add Text to a File” action, please add “c:\temp\restful-demo.txt” in the File Path field and add “This a test” in the ‘Text To Add’ field.
- For the “User Input – Stop for Manual Operation” action
    - Operation Description = Restful Demo
    - successLabel = Success
    - successMessage = Successful run
    - failureLabel = Failure
    - failureMessage = This failed
- Create a process “Restful-Demo” and publish it
- Now we need to switch to “Releases→Template Categories” and create the following
    - Template category
    - Deployment template
    - Add the process that you have already published, and add it as deployment step
    - Deployment plan
    - Create a project
    - Create a new deployment off the deployment template
    - Create a deployment
- Once the deployment is running and you get to the user input action

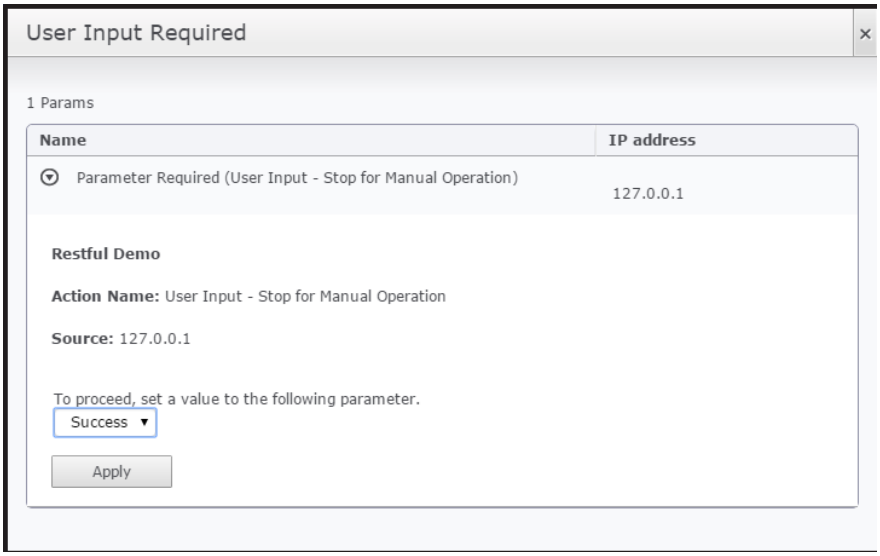


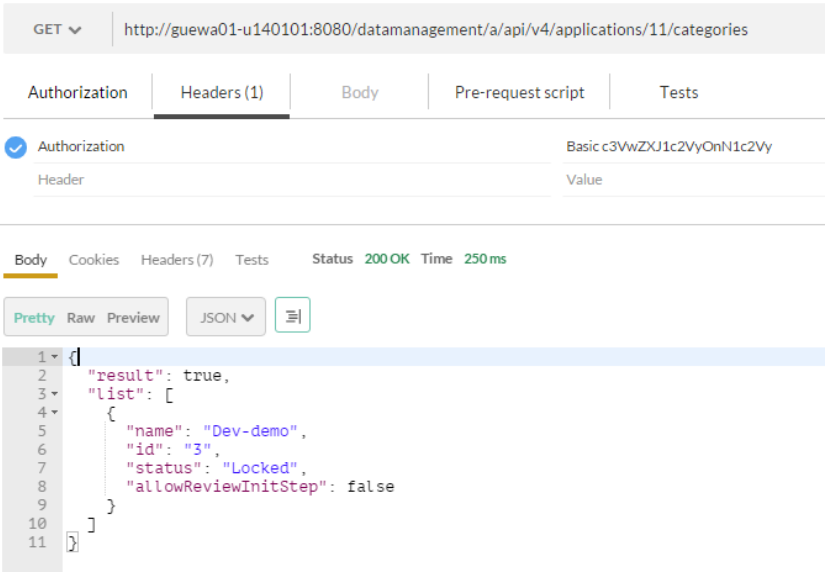
Figure 4: Typical SOAPUI Restful call

Running the Restful Calls

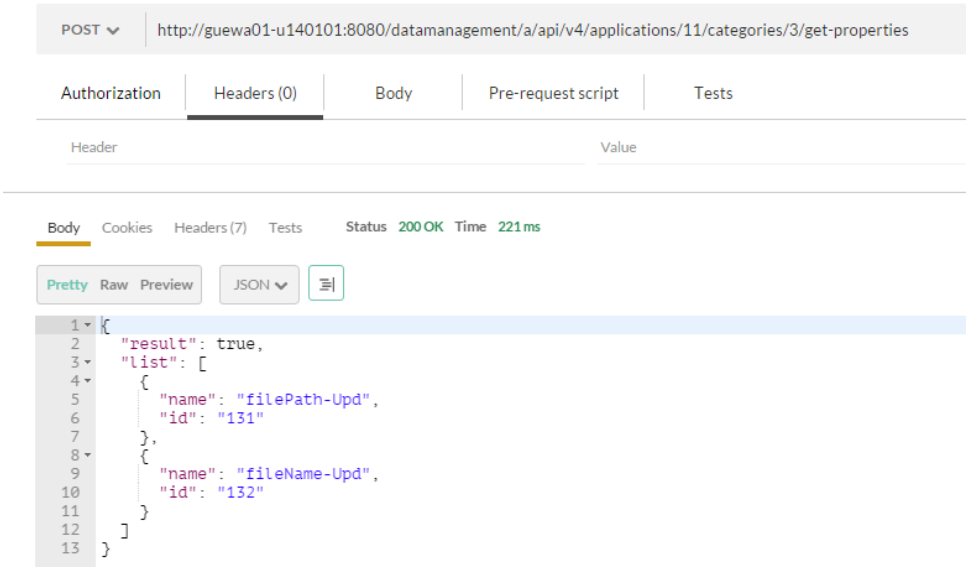
Now we are going to be running the following restful calls, you can use your restful processor:

- Applications – getting a list of the applications.
    - This is a get -- http://<RA-server>:8080/datamanagement/a/api/v2/applications
- Note, the application ID that you would like to use.

- Now, as it appears below, you need to run the following URI, in this case this is for a version 4 call.
    - http://<RA-Server>:8080/datamanagement/a/api/v4/applications/<appID>/categories
-



- Notice the results as JSON-based, and you will get the names of the available templates categories under the Releases→Templates Categories in the ROC.
- Now we are going to get the properties of one of the template categories that we just obtained.
    - http://<RA-server>:8080/datamanagement/a/api/v4/applications/<appID>/categories/<catId>/get-properties



- As you can see values shown in the JSON envelope represent the deployment template properties that have been setup for this given deployment.

The Release Automation restful API calls that we have executed as part of this small use-case show the power of the available API calls, and how you will go about setting up the execution of these Restful API call.

REST API Calls

The following section lists the GET/POST calls by version to make it easier to find the desired REST calls based on your needs.

To use these Restful methods successfully, you will need to understand how to use the DTO (Data Transfer Object), which permits the aggregation of the data values needed to complete the Restful API method call. Appendix B contains a definition of the different DTO definition and the type of parameters required for each of these DTO objects.

For some of these calls, the HTTP request will have to be built based off the DTO object as a JSON request. As shown in the examples below.

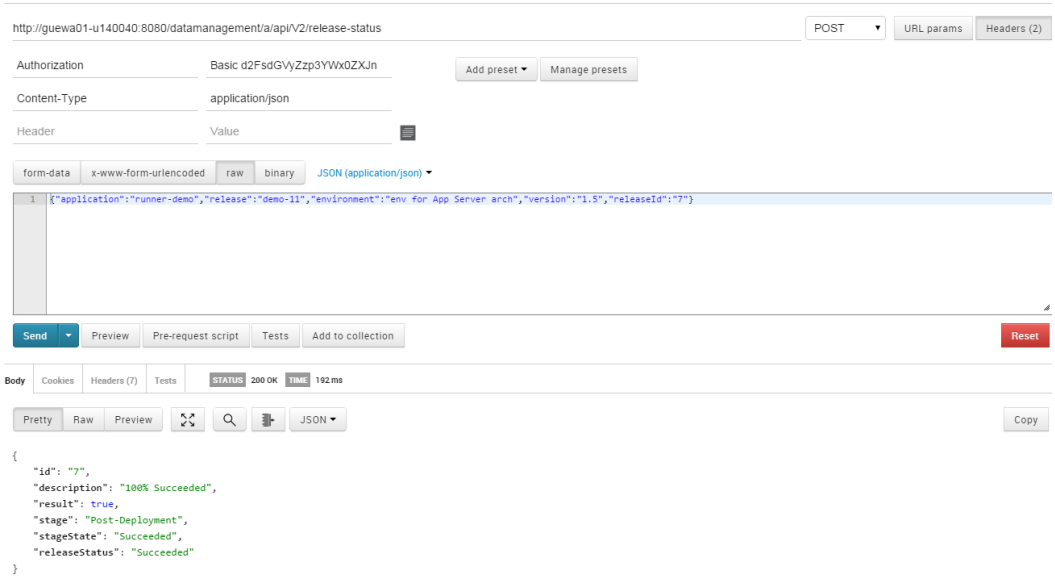


Figure 5 Example of API call with JSON body request

As you can see the above call for “release-status” uses a JSON body request, and you will also see the result set.

```
curl -X POST -H "Authorization: Basic d2FsdGVyZzp3YWx0ZXJn" -H "Content-Type: applica
tion/json" -H "Cache-Control: no-cache" -H "Postman-Token: 60b0e2bb-a8f3-48ca-b147-aa
88a73585b0" -d '{"application":"runner-demo","release":"demo-11","environment":"env f
or App Server arch","version":"1.5","releaseId":"7"}' http://guewa01-u140040:8080/dat
amanagement/a/api/V2/release-status
```

Figure 6: CURL Output

Here is the CURL line as created for the same API call, with all the correct JSON body parameters.

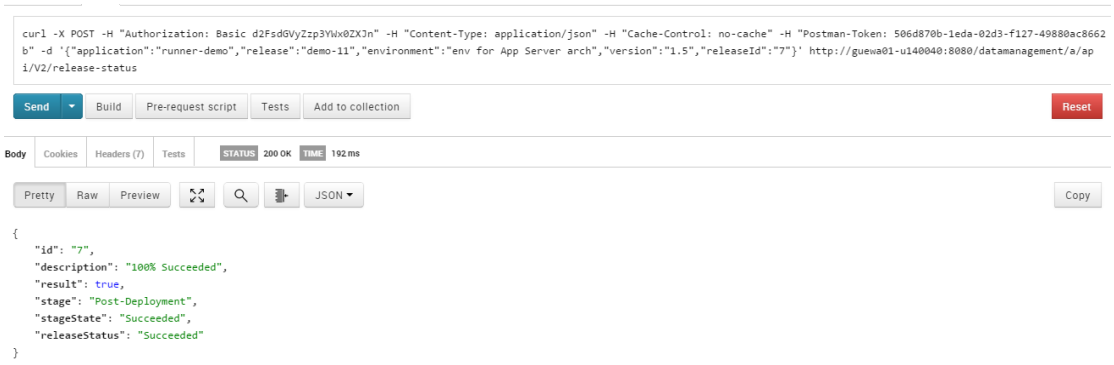


Figure 7: CURL line in Postman

In the figure above you can see how the Postman tool will make the call with the same Curl parameters and the response.

REST API V1 Calls

This section will list the available GET/POST for version 1 REST API calls, for which you don't have to include the "V1" value in the URL call.

GET API V1 Calls

These are the GET API calls available for "v1".

Method	URL	Description	DTO Used
GET	/applications	Retrieves all the applications that are part of the RA installation.	None

GET	/applications/{appId}	Retrieves a specific application by providing the application id in the path	None
GET	/applications/{appId}/environments	Retrieves all the environments associated with a specific application	None
GET	/applications/{appId}/environments/{envId}	Retrieves a specific environment linked with the provided application id	None
GET	/applications/{appId}/environments/{envId}/releases	Retrieves all the releases associated with the provided application and environment ids	None
GET	/applications/{appId}/environments/{envId}/releases/{releaseId}	Retrieves a specific release associated with the provided application and environment ids	None
GET	/applications/{appId}/templates	Retrieves all templates associated with a specific application id	None
GET	/applications/{appId}/templates/{templateId}	Retrieve a specific template associated with the provided application id	None
GET	/export/processes	Export a process details to a PDF document based off an application id **Query parameters:** reportName reportDescription **Filters** type period applications environments categories users statusType	Here is an example in how to build the URI: /export/processes?type=raw+data&reportName=example&reportDescription=example+of+export-processes&period=100 For additional information, please see Appendix A

		templates versions statuses	
GET	/export/releases	Export a release details to a PDF document based off an application id **Query parameters:** reportName reportDescription **Filters** type period applications environments categories users statusType templates versions statuses	Here is an example in how to build the URI: /export/releases?type=raw+data&reportName=example&reportDescription=example+of+export-releases&period=100 For additional information, please see Appendix A

POST API V1 Calls

These are the POST API calls available for "v1".

Method	URL	Description	DTO Used
POST	/release-status	Retrieves the status of a deployment based off the application, environment, release, and version names, as well as releaseId which are passed as parameters. The results are different for V1 and V2 support.	Request: ReleaseStatusApiDto

POST	/delete-release	Deletes a deployment based off the application, environment , and release names, as well as the version number. You could also pass just the releaseId in a JSON body.	Request: ReleaseBasicApiDto Response: ResponseData
POST	/export-release	Export a release based off the release id provided via a JSON body	Request: ReleaseBasicApiDto
POST	/export-template	Export a deployment template via XML format and the required parameters are: Application and template names, as well as the templateId in a JSON body	Request: TemplateBasicApiDto

REST API V2 Calls

This section will list the available GET/POST for version 2 (V2) REST API calls.

GET REST API V2 Calls

These are the GET API calls for V2.

Method	**URL**	**Description**	**DTO Used**
GET	/releases-reports	Retrieves all deployments by providing the following parameter values for template, application, and environment names or ids. You can also pass the period as number of days back for results (the default is 7) The following list values for: releaseStatus=Active, suspended, failed, 100% Suceeded Stage=deployment, post-deployment	Response: ReleaseApiDto

		stageState=pending, succeeded .	
GET	/step-status/{stepId}	Retrieves the status of a specific step id	Response: ResponseData

POST REST API V2 Calls

These are the POST API calls for V2.

Method	URL	Description	DTO Used
POST	/artifact-details	Retrieves the details of an artifact type. This call is **only** relevant for **RA 4.7.** For v5.0 and above, please use /artifact-version-details	Request: ArtifactBasicApiDto Response: ArtifactsApiDto
POST	/artifact-status	Retrieves the status of an artifact type. This call is **only** relevant for **RA 4.7.** For v5.0 and above, please use /artifact-version-version	Request: ArtifactBasicApiDto Response: ArtifactsApiDto
POST	/create-artifact	Create an artifact based on supplied parameters. This method only applies to **RA 4.7.** For v5.0 and above, please use /create-artifact-package-xml	Request: ArtifactBasicApiDto Response: ArtifactsApiDto
POST	/create-release	Creates a release from an existing template. This method only applies to **RA 4.7.** For v5.0 and above, please use /run-deployment-plan. This particular call will create a release from existing templates, and you will need to provide the JSON body the key values for the template and release details.	Request: CreateReleaseApiDto Response: ResponseData
POST	/release-status	Retrieves the status of a deployment based off the application, environment, release, and version names,	Request: ReleaseBasicApiDto Response: ReleaseStatusApiDto

		as well as releaseId which are passed as parameters. The results are different for V1 and V2 support.	
POST	/run-release	Runs a prepared deployment, which will be updated with deployment details. This is controlled by providing the application, environment, release, and version names. This call is only relevant in **RA 4.7. .** For v5.0 and above, please use /run-deployments.	Request: RunReleaseApiDto Response: ResponseData
POST	/run-template	This call will either create, update, or run a release based on the supplied template id, release name, environment name, and version number. This call is only relevant in **RA 4.7.** For v5.0 and above, please use /run-deployment-plan	Request: RunTemplateDto Response: ResponseData
POST	/schedule-release	Schedules a release to execute at a given date/time	Request: ScheduleReleaseApiDto Response: ResponseData
POST	/step-status	Retrieves the status of a step by providing the step id via a JSON body	Request: StepApiDto Response: ResponseData
POST	/stop-release	Stops a running a deployment based off the provided application, release, and environment names, as well as the release id	Request: ReleaseBasicApiDto Response: ResponseData
POST	/update-release	Updates a deployment using XML entries to describe the changes. This is accomplished by supplying the application, environment, release, and version names. This call is only relevant in **RA 4.7. .** For v5.0 and above, please use /load-manifest	Request: UpdateReleaseApiDto Response: ResponseData

REST API V3 Calls

This section will list the available GET/POST for version 3 REST API calls. Please keep in mind that these calls can only be used starting with Release Automation 5.0.x onwards. Another item to keep in mind is that all the version 3 API calls will require a JSON body as a request to perform the API call.

GET REST API V3 Calls

These are the GET API calls available for "V3".

Method	URL	Description	DTO Used
GET	/applications/{appId}/projects	Retrieves all the active projects associated with the provided application id	Response: ProjectApiDto
GET	/applications/{appId}/projects/{projectId}	Retrieves the specified active project for the provided application id	Response: ProjectApiDto
GET	/applications/{appId}/projects/{projectId}/deployment-plans	Retrieves all the deployment plans associated with the provided application and project ids	Response: DeploymentPlanResponseApiDto
GET	/applications/{appId}/projects/{projectId}/deployment-plans/{deploymentPlanId}	Retrieves the specific deployment plan associated with the provided application and project ids. In addition the status of all the environment, in which the deployment was executed	Response: DeploymentPlanResponseApiDto

POST REST API V3 Calls

These are the POST API calls available for "V3".

Method	URL	Description	DTO Used
POST	/add-artifact-package-to-deployment	Add an artifact package to an existing deployment.	Request: CreateDeploymentPlanApiDto

			Response: DeploymentPlanResponseApiDto
POST	/artifact-version-details	Retrieves the details of an artifact type. The valid request in the JSON body will be: Artifact version id Artifact definition id and version name App, artifact-type, and artifact-definition name, as well as the artifact version.	Request: ArtifactVersionApiDto Response: ArtifactsApiDto
POST	/artifact-version-status	Retrieves the upload status of an artifact type.	Request: ArtifactVersionApiDto Response: ArtifactStatusApiDto
POST	/assign-new-artifact-package-to-deployment-plan	Creates and adds an artifact package to an existing deployment plan	Request: CreateArtifactForDeploymentPlanDto Response: ResponseDataApiDto
POST	/create-artifact-package	Creates an empty artifact package for a given application id	Request: CreateArtifactPackageApiDto Response: ResponseData
POST	/create-artifact-package-xml	Creates a new artifact package based off the information provided by the XML file for a given application id	Request: ArtifactPackageUploadDto Response: ResponseData
POST	/create-artifact-version	Creates an artifact definition version based off definition id or application id	Request: CreateArtifactVersionApiDto Response: ResponseData
POST	/create-deployment-plan	Creates a deployment plan from an existing deployment template	Request: CreateDeploymentPlanApiDto

			Response: DeploymentPlanResponseApiDto
POST	/create-project	Creates a new project for a specific application id	Request: CreateProjectApiDto Response: ResponseData
POST	/delete-artifact-version	Deletes an artifact definition version	Request: DeleteArtifactVersionApiDto Response: ResponseData
POST	/get-artifact-package-content	Retrieves the artifact definition versions for a specific artifact package	Request: ArtifactPackageApiDto Response: ArtifactNameDTOList
POST	/get-artifact-versions	Retrieves the artifact versions for a specified artifact definition	Request: ArtifactPackageApiDto Response: ArtifactNameDTOList
POST	/get-environment-parameter	Retrieves the value of a parameter per provided environment id	Request: EnvironmentParamaterApiDto Response: ResponseEnvironmentParameterApiDto
POST	/load-manifest	Loads a manifest file to a specified deployment plan	Request: LoadManifestApiDto Request: ResponseDataApiDto
POST	/load-tokens	Loads token values to a specified environment id	Request: LoadTokenApiDto Response: ResponseDataApiDto
POST	/run-deployment-plan	Creates a deployment plan from an existing deployment template name	Request: RunDeploymentPlanApiDto Response: DeploymentPlanResponseApiDto

POST	/run-deployments	Creates a deployment from an existing deployment plan, or the user can create or run the deployment in the environments provided	Request: DeploymentApiDto Response: DeploymentResponseApiDto
POST	/update-environment-parameter	Retrieves the value of an environment parameter based off the environment id, parameter path, and parameter value	Request: FullEnvironmentParameterApiDto Response: ResponseDataApiDto
POST	/upgrade-agents	This call will upgrade the RA agents based off the following options: • All – all agents in the RA installation • Agents_by_ES – All agents under control by an Execution Server • Specific_agents – A specific agent	Request: AgentUpgradeApiDto Response: AgentUpgradeApiResponse

REST API V4 Calls

This section will list the available GET/PUT/POST/DELETE for version 4 REST API calls. Please keep in mind that these calls can only be used starting with Release Automation 5.5.2 onwards. Another item to keep in mind is that all the version 4 API calls will require a JSON body as a request to perform the call.

GET REST API V4 Calls

These are the GET API calls available for "V4".

Method	URL	Description	DTO Used
GET	/applications/{appId}/categories	Gets the template categories in an application.	Response: TemplateCategoryDTOList
GET	/applications/{appId}/categories/{categoryId}/templates	Gets the deployment templates of a template category	Response: DeploymentTemplateApiDtoList

GET	/applications/{appId}/categories/{categoryId}/templates/{templateId}/exportXML	Export the XML metadata of a deployment template	Response: ResponseDataApiDto

PUT REST API V4 Calls

These are the PUT API calls available for "V4".

Method	URL	Description	DTO Used
PUT	/applications/{appId}/categories/{categoryId}	Updates a template category	Response: ResponseDataApiDto
PUT	/applications/{appId}/categories/{categoryId}/templates/{templateId}	Updates a deployment template	Response: ResponseDataApiDto
PUT	/applications/{appId}/categories/{categoryId}/templates/{templateId}/steps/{stepId}/update-dependencies	Updates the dependencies of a deployment template's step	Response: ResponseDataApiDto

POST REST API V4 Calls

These are the POST API calls available for "V4".

Method	URL	Description	DTO Used
POST	/applications/{appId}/categories/{categoryId}/get-properties	Get the template properties of a given template category	Response: CategoryPropertyDTOList
POST	/applications/{appId}/categories/{categoryId}/get-steps	Gets all the steps of a template category	Response: CategoryStepDTOList
POST	/applications/{appId}/categories/{categoryId}/properties	Creates a template property	Response: ResponseDataApiDto
POST	/applications/{appId}/categories/{categoryId}/properties/{propertyId}	Update a template property	Response: ResponseDataApiDto
POST	/applications/{appId}/categories/{categoryId}/steps	Create a step in a template category	Response: ResponseDataApiDto
POST	/applications/{appId}/categories/{categoryId}/steps/{stepId}	Updates a template category step	Response: ResponseDataApiDto
POST	/applications/{appId}/categories/{categoryId}/templates	Create a template	Response: ResponseDataApiDto

POST	/applications/{appId}/categories/{categoryId}/templates/{templateId}/get-steps	Gets deployment template steps	Response: TemplateStepDTOList
POST	/applications/{appId}/categories/{categoryId}/templates/{templateId}/steps	Attaches a category step to a template	Response: ResponseDataApiDto
POST	/deployments/{deploymentId}/update-approval	Changes a deployment manual approval status	Response: ResponseDataApiDto

DELETE REST API V4 Calls

These are the DELETE API calls available for "V4".

Method	URL	Description	DTO Used
DELETE	/applications/{appId}/categories/{categoryId}	Deletes a template category	Response: ResponseDataApiDto
DELETE	/applications/{appId}/categories/{categoryId}/properties/{propertyId}	Deletes a template property	Response: ResponseDataApiDto
DELETE	/applications/{appId}/categories/{categoryId}/steps/{stepId}	Deletes a template category step	Response: ResponseDataApiDto
DELETE	/applications/{appId}/categories/{categoryId}/templates/{templateId}	Deletes a deployment template	Response: ResponseDataApiDto
DELETE	/applications/{appId}/categories/{categoryId}/templates/{templateId}/steps/{stepId}	Detaches a step from a deployment template	Response: ResponseDataApiDto

Copyright Notice

Useful Links

https://wiki.ca.com/ca-release-automation/5-5-2/en/reference/rest-api-reference

https://wiki.ca.com/display/RA55/CA+Release+Automation+Home

MSDN Definition of DTO

Data Transfer Object Definition

http://www.getpostman.com/docs

https://www.base64decode.org/

Appendix A – Special Query Parameters

For the exporting of processes and releases to a PDF file, you will need to pass the following parameters as part of the URL.

Parameter	Type	Description	Required
reportName	String	Report name	False
reportDescription	String	Report description	False
Type	String	Type filter – The values to use are: raw, data, or raw+data	True
Period	Integer	Period	False
Applications	List	Application filter	False
Environments	List	Environment filter	False
Categories	List	Categories filter	False
Processes	List	Processes filter	False
Users	List	Users filter	False
statusType	List	Status type filter	False
Templates	List	Template filter	False
Versions	List	Versions filter	False
Statuses	List	Statuses filter	False

The correct usage for a "/export/processes"

http://<RA_Server>:8080/datamanagement/a/api/export/processes?type=raw+data&reportName=example&reportDescription=example+of+export-processes&period=100

The correct usage for a "/export/releases

http://<RA_Server>:8080/datamanagement/a/api/export/releases?type=raw+data&reportName=example&reportDescription=example+of+export-releases&period=100

Appendix B – DTO Parameters

AgentUpgradeApiDto

Contains the data about which computers should be upgraded.

Parameter	Type	Description	Required
nodeIds	List	Collection of node-ids of the agents or execution servers to be used in the upgrade.	false
upgradeType	UpgradeType	The upgrade can be one of the following types: • *ALL* : All the agents in the system are upgraded, the nodeIds collection is ignored. • *AGENTS_BY_ES* : A list of execution-servers is provided in nodeIds collection. All of the agents on the execution-servers are upgraded. • *SPECIFIC_AGENTS* : A list of agents is provided in nodeIds collection. Only those agents are upgraded.	false

ArtifactBasicApiDto

The DTO contains the fields for specifying an artifact.

* Specify an artifact in one of the following ways:

- Artifact ID {artifactId}
- Application ID {applicationId}, Artifact name {artifact} and Artifact version {artifactVersion}
- Application name {application}, Artifact name {artifact} and Artifact version {artifactVersion}

Parameter	Type	Description	Required
application	String	Application name	true *
applicationId	Long	Application ID	true *
artifact	String	Artifact name	true *
artifactId	Long	Artifact ID	true *

Parameter	Type	Description	Required
version	String	Artifact version	true *

ArtifactPackageApiDto

Contains the ID or path of an artifact-package.

* Specify the location in one of the following ways: One has to be present.

- Artifact-package ID {artifactPackageId}
- Application name {application}, artifact-package name {artifactPackage}

Parameter	Type	Description	Required
application	String	Application name	true *
artifactPackage	String	Artifact package name	true *
artifactPackageId	Long	Artifact package ID	true *

ArtifactPackageUploadDto

Contains the xml of an artifact package.

Parameter	Type	Description	Required
applicationId	Long	The id of application to create the package at	true
xml	String	XML	true

ArtifactVersionApiDto

Contains the location of a specific artifact version

* Specify the location in one of the following ways:

- Artifact-definition ID {artifactDefinitionId}, Artifact-version {version}
- Application name {application}, artifact-type name {artifactType}, artifact-definition name {artifactDefinition},Artifact-version {version}
- Artifact-id {artifactId}

Parameter	Type	Description	Required
artifactId	Long	Artifact ID	true *
version	String	Artifact version	true *
application	String	Application name	true *
artifactDefinition	String	Artifact definition name	true *
artifactDefinitionId	Long	Artifact definition ID	true *
artifactType	String	Artifact type name	true *

ArtifactVersionsApiDto

Contains the location of artifact version(s)
* TSpecify the location in one of the following ways:

- Artifact-definition ID {artifactDefinitionId}
- Application name {application}, artifact-type name {artifactType}, artifact-definition name {artifactDefinition}

Parameter	Type	Description	Required
application	String	Application name	true *
artifactDefinition	String	Artifact definition name	true *
artifactDefinitionId	Long	Artifact definition ID	true *
artifactType	String	Artifact type name	true *

CreateArtifactApiDto

The DTO contains data to create an artifact.
* Specify a required application in one of the following ways:

- Application name {application}
- Application ID {applicationId}

Parameter	Type	Description	Required
application	String	Application name	true *
artifact	String	Artifact name	true

Parameter	Type	Description	Required
allowArtifactModifications	Boolean	True - a different file can be deployed on the same release settings. False - Only one file is allowed to be executed in the process.	false
applicationId	Long	Application ID	true *
description	String	The description of the artifact (entered by the user).	false
getterType	String	Artifact retrieval method {Agent Local File, Remote File (renamed 'FTP' in 4.2 and renamed back to 'Remote file' in 4.7), SVN, HTTP, SSH, TFS, Repository, FTP}	true
properties	Map	The properties of the getter type used.	true
server	String	Represents server cluster and is either single server id or server group id. This is determined by serverGroup property. if serverGroup==true then serverId is ServerGroup id, i.e. Category id. In this case the actual file getter will be selected out of the server group.	true
serverGroup	Boolean	Indicates whether the above server (artifact getter) is single server or a server group	true
shouldUploadNow	Boolean	should the artifact be stored in the repository now	false
storeInRepository	Boolean	should the artifact be stored in the repository on first use	false
version	String	Artifact version	false

CreateArtifactForDeploymentPlanDto

Contains the xml of an Artifact Package and deployment plan details.

* Specify the deployment plan in one of the following ways:

- Deployment plan ID {DeploymentPlanId}
- Deployment plan name {DeploymentPlan}. In this case additional build version {build} and project {project} names are required

** Specify the application in one of the following ways:

- Application ID {applicationId}
- Application name {application}

Parameter	Type	Description	Required
applicationId	Long	The id of application to create the package at	true
xml	String	XML	true
application	String	Application name	true**
build	String	Build name	true*
deploymentPlan	String	Deployment plan name	true*
deploymentPlanId	Long	Deployment plan id	true*
project	String	Project name	true*

CreateArtifactPackageApiDto

Contains the name, description and application of artifact package
* Specify the application in one of the following ways:
Note: One of the following parameters has to be present.

Application ID {applicationId}

Application Name {application}

Parameter	Type	Description	Required
application	String	Application name	true *
applicationId	Long	Application ID	true *
description	String	Artifact-package description	false
name	String	Artifact-package name	true
description	String	Description of the attribute	false
name	String	Name of the attribute	false
id	Long	ID of the attribute	false

CreateArtifactVersionApiDto

The DTO contains data to create an artifact-version.
* To specify an artifact-definition (which is mandatory) there are two possibilities:

- Artifact-definition ID {artifactDefinitionId}

- Application ID {applicationId}
- Artifact-definition name {artifactDefinition}
- Artifact-type name {artifactType}

Parameter	Type	Description	Required
applicationId	Long		true
artifactDefinition	String	Artifact-definition name	true *
artifactDefinitionId	Long	The ID of Artifact-definition	true *
artifactType	String	Artifact-type name	true *
allowArtifactModifications	Boolean	True - a different file can be deployed on the same release settings. False - Only one file is allowed to be executed in the process.	false
applicationId	Long	Application ID	true *
description	String	The description of the artifact (entered by the user).	false
getterType	String	Artifact retrieval method {Agent Local File, Remote File (renamed 'FTP' in 4.2 and renamed back to 'Remote file' in 4.7), SVN, HTTP, SSH, TFS, Repository, FTP}	true
properties	Map	The properties of the getter type used.	true
server	String	Represents server cluster and is either single server id or server group id. This is determined by serverGroup property. if serverGroup==true then serverId is ServerGroup id, i.e. Category id. In this case the actual file getter will be selected out of the server group.	true
serverGroup	Boolean	Indicates whether the above server (artifact getter) is single server or a server group	true
shouldUploadNow	Boolean	should the artifact be stored in the repository now	false
storeInRepository	Boolean	should the artifact be stored in the repository on first use	false
version	String	Artifact version	false

CreateDeploymentPlanApiDto

The required arguments in the JSON object are the deployment plan, build, project, deployment template, application***, and template category* names.

* Specify a template category 1. in one of the following ways:

- Template Category ID {templateCategoryId}
- Template Category Name {templateCategory}

*** Application could be supplied by name or id

Parameter	Type	Description	Required
application	String	Application name	true***
applicationId	Long	Application id	true***
artifactPackage	String	Artifact package name. If package name supplied, the package will be assigned to the deployment plan	false
artifactPackageAsXML	String	Contains the xml of Artifact Package	false
build	String	Build version name	true
deploymentPlan	String	Deployment plan name	true
deploymentPlanDescription	String	Deployment plan description	false
deploymentPlanType	DeploymentPlanType	Deployment plan type. Can be MINOR, MAJOR or EMERGENCY. If the type is not supplied, the plan will be marked MAJOR.	false
deploymentTemplate	String	Deployment template name	true
manifest	String	Manifest xml. If supplied	false
project	String	Project name. A project will be created if not exists	true
properties	Map	A map of {name, value} property pairs. If supplied, will update the property values for the deployment plan. note - property name must exist in the template.	false
templateCategory	String	Template category name	true *
templateCategoryId	Long	Template category id	true *
timeout	Long	Timeout in seconds (default is 180). If a there is an pre init stage & it's running exceeds the timeout, the result of the	false

Parameter	Type	Description	Required
		rest call will fail although the pre init stage won't be stopped	

CreateProjectApiDto

The DTO contains the attributes of a project.To create a project, you will need to specify the application id {applicationId}, the name of the project {name}, and optionally the description {description}. The id {id} of the created project is returned.

Parameter	Type	Description	Required
applicationId	Long	Application Id to which this project is associated with.	true
description	String	Project description	false
name	String	Project name	true

CreateReleaseApiDto
The DTO contains the data to create a release from a template, individualize a template, define release parameters, and to validate the steps of the environment before running.

* Specify a template in one of the following ways:br>
- Template ID {templateId}
- Template Name {template} and Application Name {application}

Parameter	Type	Description	Required
asynch	Boolean	True/false. Used only if releaseStepToPerform is RunDeploy. True - http call returns after the run deployment command is triggered, False - http call returns after the deployment run finished. Default is false.	false
description	String	Deployment description	false
doStepsValidation	Boolean	True - In case one step (or more) does not mach the specified environment, the operation will fail. False - All steps that wrap processes that are not assigned to the deployment environment, will be removed from the template category automatically.	false
environment	String	Environment name	true
release	String	Deployment name	true

Parameter	Type	Description	Required
releaseStepToPerform	String	Steps to perform after creating the deployment {DoNothing, RunInitOnly, RunDeploy. Default is RunInitOnly}	false
releaseType	String	Deployment type {Minor,Major,Emergency. Default is Minor}	false
timeout	Long	Timeout in second. Used only if releaseStepToPerform is RunDeploy and asynch is False. If a deployment run exceeds the timeout, the deployment is stopped automatically.	false
version	String	Deployment version. Can be null	false
application	String	Application name	true *
template	String	Template name	true *
templateId	Long	Template ID	true *

DeleteArtifactVersionApiDto

The DTO contains data to delete an artifact-version.

Specify an artifact version to delete either directly by the artifact version id or provide all of:

- Application ID {applicationId} or ApplicationName {applicationName}
- Artifact-definition name {artifactDefinition}
- Artifact-type name {artifactType}
- Artifact-version name {version}

Parameter	**Type**	**Description**	**Required**
applicationId	Long	Application ID	false
applicationName	String	Application name	false
artifactDefinition	String	Artifact-definition name	false
artifactType	String	Artifact-type name	false
id	Long	Artifact version id	false
version	String	Artifact version	false

DeploymentApiDto

The required arguments in the JSON object are the deployment plan id**, deployment, and application names.

* Specify the environments in one of the following ways:

- Environment Names list {environments}

- Environment Ids list {environmentIds} (default)

** Supply the deployment plan in one of the following ways:
- Deployment plan ID {DeploymentPlanId}
- Deployment plan name {DeploymentPlan}. In this case additional build version {build} and project {project} names are required

*** The application can be supplied by name or id

Parameter	Type	Description	Required
applicationId	Long	Application id	true***
deployment	String	Deployment name	true
deploymentDescription	String	Deployment description	false
environmentIds	List	List of Environment ids the deployment should run on	true *
environments	List	List of Environment names the deployment should run on	true *
stageToPerform	String	Execute The stage after deployment has been created. All The stages preceding will be executed. {None, Initialization, Validation, Approval-Gate, Distribute-Execution-Server, Distribute-Agent, Deployment, Post-Deployment} Note - If None has been selected, then the deployment will only be created. Default - will run all the stages.	false
application	String	Application name	true**
build	String	Build name	true*
deploymentPlan	String	Deployment plan name	true*
deploymentPlanId	Long	Deployment plan id	true*
project	String	Project name	true*

EnvironmentParameterApiDto
The DTO contains the fields that describe a parameter for a specific environment.

* Specify an environment in one of the following ways:
- Environment ID {environmentId}
- Environment name {environment} and Application Id {applicationId}
- Environment name {environment} and Application name {application}

Parameter	Type	Description	Required
application	String	Application Name	true *
applicationId	Long	Application ID	true *
environment	String	Environment Name	true *
environmentId	Long	Environment ID	true *
parameterPath	String	Parameter Path	true

FullEnvironmentParameterApiDto
The DTO contains the fields to update the value of an environment parameter

* Specify an environment in one of the following ways:
- Environment ID {environmentId}
- Environment name {environment} and Application Id {applicationId}
- Environment name {environment} and Application name {application}

Note: The value needs to be in a JSON format. i.e. \" instead of ". (same as in update parameter)

Parameter	Type	Description	Required
application	String	Application Name	true *
applicationId	Long	Application ID	true *
environment	String	Environment Name	true *
environmentId	Long	Environment ID	true *
parameterPath	String	Parameter Path	true
serverTypes	Map	A Map of Server Types and correlated values	false
arrayValue	List	Value of an Array Parameter	false
loopValue	List	Value of an Loop Folder Parameter	false
simpleValue	String	Value of a Simple Parameter	false

LoadManifestApiDto
Contains the manifest xml and deployment plan details.

* Specify the deployment plan in one of the following ways:
- Deployment plan ID {DeploymentPlanId}
- Deployment plan name {DeploymentPlan}. In this case additional build version {build} and project {project} names are required

** Specify the application in one of the following ways:

- Application ID {applicationId}
- Application name {application}

Parameter	Type	Description	Required
applicationId	Long	Application id	true**
manifest	String	Manifest XML file	true
application	String	Application name	true**
build	String	Build name	true*
deploymentPlan	String	Deployment plan name	true*
deploymentPlanId	Long	Deployment plan id	true*
project	String	Project name	true*

LoaDTOkenApiDto

Contains the token values for an environment.

Parameter	Type	Description	Required
environmentId	Long	Environment id	true**
manifest	String	Manifest XML file	true

ReleaseBasicApiDto

The DTO contains the fields for specifying a release.

* Specify a deployment in one of the following ways:

- Deployment ID {releaseId}
- Deployment name {release}, Deployment version {deploymentVersion}, Application name {application} and Environment name {environment}

Parameter	Type	Description	Required
application	String	Application name	true *
environment	String	Environment name	true *
release	String	Deployment name	true *
releaseId	Long	release ID	true *
version	String	Deployment version. Can be null	false *

RunDeploymentPlanApiDto

To run a deployment plan, create the deployment plan and the deployments.The fields below are required in order to create both.

* Specify a template category in one of the following ways:
    - Template Category ID {templateCategoryId}
    - Template Category Name {templateCategory}

** Environments may be provided in one of the following ways:
    - Environment Names list {environments}
    - Environment Ids list {environmentIds} (default)

*** The application could be supplied by name or id.

Parameter	Type	Description	Required
deployment	String	Deployment name	true **
deploymentDescription	String	Deployment description	false
deploymentStageToPerform	String	Execute The stage after deployment has been created. All The stages preceding will be executed. {None, Initialization, Validation, Approval-Gate, Distribute-Execution-Server, Distribute-Agent, Deployment, Post-Deployment} Note - If [None] has been selected, then the deployment will only be created. If no deploymentStageToPerform is specified - it will run all the stages.	false
environmentIds	List	List of environment ids the deployment should run on	true **
environments	List	List of environment names the deployment should run on	true **
application	String	Application name	true***
applicationId	Long	Application id	true***
artifactPackage	String	Artifact package name. If package name supplied, the package will be assigned to the deployment plan	false
artifactPackageAsXML	String	Contains the xml of Artifact Package	false

Parameter	Type	Description	Required
build	String	Build version name	true
deploymentPlan	String	Deployment plan name	true
deploymentPlanDescription	String	Deployment plan description	false
deploymentPlanType	DeploymentPlanType	Deployment plan type. Can be MINOR, MAJOR or EMERGENCY. If the type is not supplied, the plan will be marked MAJOR.	false
deploymentTemplate	String	Deployment template name	true
manifest	String	Manifest xml. If supplied	false
project	String	Project name. A project will be created if not exists	true
properties	Map	A map of {name, value} property pairs. If supplied, will update the property values for the deployment plan. note - property name must exist in the template.	false
templateCategory	String	Template category name	true *
templateCategoryId	Long	Template category id	true *
timeout	Long	Timeout in seconds (default is 180). If a there is an pre init stage & it's running exceeds the timeout, the result of the rest call will fail although the pre init stage won't be stopped	false

RunReleaseApiDTO
The DTO contains the data to run a deployment, individualize a release and define the timeout, and to run asynchronously or not.

* Specify a deployment in one of the following ways:
- Deployment ID {releaseId}
- Deployment name {release}, Deployment version {deploymentVersion}, Application name {application} and Environment name {environment}

Parameter	Type	Description	Required
asynch	Boolean	True/false. If true, http call returns after the run release command is triggered, and not after the deployment run finished. Default is false, http call returns after deployment	true

Parameter	Type	Description	Required
runOnlyInit	Boolean	True - Run only initialization. False - Run the complete deployment. Default is False.	false
timeout	Long	Timeout in second. If a deployment run exceeds the timeout, the deployment is stopped automatically	true
application	String	Application name	true *
environment	String	Environment name	true *
release	String	Deployment name	true *
releaseId	Long	release ID	true *
version	String	Deployment version. Can be null	false *

RunTemplateApiDTO

The DTO contains the data to run a template which includes: create release, update release, and run release.

* Specify a template in one of the following ways:
  - Template ID {templateId}
  - Template Name {template} and Application Name {application}

Parameter	Type	Description	Required
description	String	Deployment description	false
doStepsValidation	Boolean	True - In case one step (or more) does not mach the specified environment, the operation will fail. False - All steps that wrap processes that are not assigned to the deployment environment, will be removed from the template category automatically.	false
environment	String	Environment name	true
properties	Map	{map of {key,value}}	false
release	String	Deployment name	true
releaseType	String	Deployment type {Minor,Major,Emergency. Default is Minor}	false
version	String	Deployment version. Can be null	false
application	String	Application name	true *
template	String	Template name	true *
templateId	Long	Template ID	true *

ScheduleReleaseApiDTO
The DTO contains the data to schedule a release, specify a release, and define the scheduled date, time and duration.

* Specify a release in one of the following ways:
- Deployment ID {releaseId}
- Deployment name {release}, Deployment version {releaseVersion}, Application name {application} and Environment name {environment}

Parameter	Type	Description	Required
estimatedDurationMinutes	Long	The duration in minutes in this format: mmmm	true
scheduledDate	String	The date in this format: dd/mm/yy	true
scheduledTime	String	The time in this format: HH:mm	true
application	String	Application name	true *
environment	String	Environment name	true *
release	String	Deployment name	true *
releaseId	Long	release ID	true *
version	String	Deployment version. Can be null	false *

StepApiDto
The DTO contains just the step ID.

Parameter	Type	Description	Required
stepId	Long	Step ID	true

TemplateBasicApiDto
The DTODTO contains the fields for specifying a template.

* Specify a template in one of the following ways:
- Template ID {templateId}
- Template Name {template} and Application Name {application}

Parameter	Type	Description	Required
application	String	Application name	true *
template	String	Template name	true *
templateId	Long	Template ID	true *

UpdateReleaseApiDto
The DTODTO contains the data to update a release, specify a release, and the xml that describes the updates.

* Specify a release in one of the following ways:
- Deployment ID {releaseId}
- Deployment name {release}, Deployment version {releaseVersion}, Application name {application} and Environment name {environment}

Parameter	Type	Description	Required
xml	String	Xml that describes the needed updates	true
application	String	Application name	true *
environment	String	Environment name	true *
release	String	Deployment name	true *
releaseId	Long	release ID	true *
version	String	Deployment version. Can be null	false *

AgentUpgradeApiResponse
The result of the agent-upgrade request. The possible values are:

- BLOCKED: Another upgrade process is running.
- ERROR: More data can be found in logs.
- SUCCESS

Parameter	Type	Description
class	Class	
declaringClass	Class	

ApplicationApiDto
The application DTODTO contains the application id, application name, and description.

Parameter	Type	Description
description	String	Description of the attribute
name	String	Name of the attribute
id	Long	ID of the attribute

ArtifactNameDtoList
Contains a list of artifact-versions, and each has some of the following properties:

- id - The id of artifact-version
- version - The name of artifact-version
- description - The description of artifact-version
- definitionName - The name of artifact-definition this artifact-version belongs to
- typeName - The name of artifact-type this artifact-version belongs to

Parameter	Type	Description
list	List	The list of artifact-version data holders

ArtifactStatusApiDto
Retrieves the status data of the artifact data. If the request fails the data contains information about the failure.

Parameter	Type	Description
description	String	Description of the Response/Artifact
id	Long	Success - Artifact ID
result	Boolean	Result of the requsest
status	String	Success - Status of the artifact

ArtifactsApiDto
The DTODTO contains all the available data on an artifact.

Parameter	Type	Description
allowArtifactModifications	Boolean	True - a different file can be deployed on the same release settings. False - Only one file is allowed to be executed in the process.
deleted	Boolean	Indicates if the artifact version is archived.
description	String	The description of the artifact (entered by the user).
getterType	String	Artifact retrieval method {Agent Local File, Remote File (renamed 'FTP' in 4.2 and renamed back to 'Remote file' in 4.7), SVN, HTTP, SSH, TFS, Repository, FTP}
properties	Map	The properties of the getter type used.
server	String	Represents server cluster and is either single server id or server group id. This is determined by serverGroup property. if serverGroup==true then serverId is ServerGroup id, i.e.

Parameter	Type	Description
		Category id. In this case the actual file getter will be selected out of the server group.
serverGroup	Boolean	Indicates whether the above server (artifact getter) is single server or a server group
shouldUploadNow	Boolean	should the artifact be stored in the repository now
storeInRepository	Boolean	should the artifact be stored in the repository on first use
application	String	Application name
applicationId	Long	Application ID
artifact	String	Artifact name
artifactId	Long	Artifact ID
version	String	Artifact version

DeploymentPlanResponseApiDto

The DTODTO contains the values of a deployment plan.

Parameter	Type	Description
application	String	Application name
archived	Boolean	
artifactPackage	String	Artifact package name
build	String	Build version name
deploymentPlan	String	Deployment plan name
deploymentPlanId	Long	Deployment plan id
deploymentResults	DeploymentResponseApiDto[]	Deployments created and/or run from the deployment plan
deploymentTemplate	String	Deployment template name
deploymentsStatus	DeploymentStatusApiDto[]	The Deployments status
description	String	Description in case of the failure
project	String	Project name
templateCategory	String	Template category name
description	String	Description of the operation
result	Boolean	Result of the requsest

DeploymentResponseApiDto
The DTODTO contains the values of a deployment.

Parameter	Type	Description
envId	Long	Environment id of the of the deployment
envName	String	Environment name of the deployment
id	Long	ID of entity
description	String	Description of the operation
result	Boolean	Result of the requsest

EnvironmentApiDto
The environment contains the environment name, environment id, application name, application id, and description.

Parameter	Type	Description
applicationId	long	Application ID
applicationName	String	Application name
description	String	Description of the attribute
name	String	Name of the attribute
id	Long	ID of the attribute

ProjectApiDto
The DTODTO contains the attributes of a project.

Parameter	Type	Description
applicationId	Long	Application Id to which this project is associated with.
archived	Boolean	Status of the project. Non-active/deleted projects are marked archived.
description	String	Project description
id	Long	Project Id
name	String	Project name

ReleaseApiDto
The DTODTO contains all the available release data.

Parameter	Type	Description
applicationId	long	Application ID
applicationName	String	Application name
endTime	String	End time
environmentId	long	Environment ID
environmentName	String	Environment name
releaseStatus	String	Release Status - since V2
stage	String	Current stage of the release
stageState	String	Currnet state of the stage of the release
startTime	String	Start time
status	String	Release status - v1 only
templateId	long	Template ID
templateName	String	Template name
version	String	Release version
description	String	Description of the attribute
name	String	Name of the attribute
id	Long	ID of the attribute

ReleaseStatusApiDto

The DTODTO is divided into two sections: Success and Failure, and not all fields appear in both sections.

- Success - all the fields contain data linked to the release
- Failure - all the fields contain data regarding the reason for failure

Note: There are differences between version1 and version2.

Parameter	Type	Description
description	String	Success - general status and progress of the release, Failure - reason of failure
id	Long	Success - Release ID
releaseErrors	List	Failure - Errors of the stages and their rellevant steps
releaseStatus	String	Success - V2: Release Status {Active/Succeeded/Failed/Canceled}
result	Boolean	Result of the requsest

Parameter	Type	Description
stage	String	Success - V2: Current stage of the release {Initialization/Approval-Gate/Deployment/Post-Deployment}
stageState	String	Success - V2: Status of the current stage: {Pending/Running/Paused/Running-With-Errors/Succeeded/Failed/Canceled}
status	String	Success - V1: Current state of the release: {Active/Finished/Failed/Canceled}

ResponseData

This is the general DTO that is returned to the user. The DTO data is concurrent with the success or the failure of the request.

Parameter	Type	Description
id	Long	ID of entity
description	String	Description of the operation
result	Boolean	Result of the requsest

ResponseDataApiDto

This is the general DTO that is returned to the user. The DTO data is concurrent with the success or the failure of the request.

Parameter	Type	Description
description	String	Description of the operation
result	Boolean	Result of the requsest

ResponseEnvironmentParameterApiDto

The DTO contains the fields that describe the value of parameter.

Note: The rest call is from version 3 only.

Parameter	Type	Description
description	String	Description of the operation
result	Boolean	Result of the requsest
parameterPath	String	Parameter Path
serverTypes	Map	A Map of Server Types and correlated values
arrayValue	List	Value of an Array Parameter

Parameter	Type	Description
loopValue	List	Value of an Loop Folder Parameter
simpleValue	String	Value of a Simple Parameter

TemplateApiDto

The DTO contains the available data for a template.

Parameter	Type	Description
applicationId	long	Application ID
applicationName	String	Application Name
description	String	Description of the template
status	String	Template status
description	String	Description of the attribute
name	String	Name of the attribute
id	Long	ID of the attribute