



Version 24.0



Publication Date: January, 2024

Prepared by the DROPS Software Documentation Team

North America & LATAM 1 N. State St, 15th Floor Chicago, IL USA 1-603-371-9074 1-603-371-3256 (support calls only) sales-us@arcadsoftware.com

EMEA (HQ)

55 Rue Adrastée – Parc Altaïs 74650 Chavanod/Annecy France +33 450 578 396 sales-eu@arcadsoftware.com Asia Pacific

5 Shenton Way #22-04 UIC Building Singapore 068808 sales-asia@arcadsoftware.com



Preface

Document Purpose

This document is intended to guide DROPS Server Administrators through admin-level server management procedures.

This document is intended to outline the basic principles of DROPS' Public REST API and how to use it.

Intended Audience

This document is intended for REST API users.

Related Documentation

Related Documentation
ARCAD SSL Configuration Guide
DROPS Configuration Guide
DROPS Datasheet
DROPS Installation Guide
DROPS Release Notes
DROPS Script Reference Guide
DROPS User Guide

Table 1: Related Documentation

Publication Record

Unless stated otherwise, all content is valid for the most current version of Public REST API listed as well as every subsequent version.

Product Version	Document Version	Publication Date	Update Record
≥ 24.0	2.1	January, 2024	No functional changes
23.3	2.0	November, 2023	No functional changes
23.2	1.9	July, 2023	No functional changes
23.1	1.8	April, 2023	No functional changes

Table 2: DROPS Operational Guide Publication Record



Contents

Preface	2
Contents	
1 The DROPS architecture & workflow	
2 DROPS' Public REST API	7
2.1 Basic principles	7
2.2 Using DROPS' Public REST API to create releases	
2.2.1 Opening the release	
2.2.2 Creating the import instance	
2.2.3 Running the import instance	12
2.2.4 Reading the result of the import instance execution	13
2.3 Using DROPS' Public REST API to deploy	14
2.3.1 Creating the deployment instance	15
2.3.2 Running the transfer instance, then the deployment instance.	
2.3.3 Rolling back	
2.3.4 Tracking execution instances	24





Figure 1: The DROPS functional diagram

The DROPS architecture is based on the DROPS Server, which is a Java application that offers a REST interface on port 5254, by default. The various *clients* (fat client, thin client, APIs, etc.) communicate with the server based on the REST interface.

The DROPS Server drives deployments through the DROPS Agent, installed on the target servers. The agents are accessed either through a REST interface on the default port 8230 if they are installed manually, or via SSH when using the "agentless" solution (no installation on the target server that is reachable via ssh).

Deployments are made in 3 phases in DROPS:

- 1. Declaring the applications and provisioning of the releases to deploy.
- 2. Declaring the technical and logical targets.
- 3. Orchestrating the deployment of the releases to the target (logical) environments.

Public REST API v24.0 1 The DROPS architecture & workflow | Operational Guide



Figure 2: The DROPS workflow

Provisioning the releases to deploy

The administrator, or users with the *release configuration* roles, must first declare the **applications** that are a logical set of **components**.

For each component a set of **import strategies** is defined in order to collect or extract the different artifacts included in the component subset of the application. An **import instance** is the selective execution of the different import strategies of the components of the application.

The result of the execution of an import instance is the creation or the update of a **release**, which is the basic element that will be deployed by DROPS during the last phase.

Defining technical and logical targets

The administrator, or the users with the *environment configuration* roles, must first declare the **infrastructures items**, which are the technical targets (DB, application servers, file systems, third systems, etc.) to which will be deployed the artifacts of the different components of the application.

The infrastructure items are grouped into **environments** that are the logical targets (UAT, TEST, PRE-PROD, PROD, etc.) for deployment during the last phase. It is possible to group environments into **environment groups** which allows simultaneous deployment on a set of environments.

For each environment, **deployment processes** are published for the applications allowed in these environments. A deployment process is a deployment scheme in which components of the application are published to an infrastructure item using a list of actions to be executed. An environment can contain several deployment schemes for the same application, the latter implementing all or part of the available infrastructure items and components using specific action lists. Actions can interact with artifacts, technical targets, host systems and their file systems, and also third-party systems (via APIs, interfaces, etc.).

Orchestrating deployments

The last phase of operations is the implementation and orchestration of deployments. This is to select a logical target and an available deployment scheme for that target that therefore selects an application. For this application the selection of the release to deploy completes the creation of a **deployment instance**



which is the actual deployment process on the logical target. The deployment instance is then executed which results in the deployment of the artifacts to the target system(s). The execution also produces statistics that can be retrieved in reports.

2 DROPS' Public REST API

2.1 Basic principles

DROPS' Public REST API relies on conventional CRUD methods for persistence of data. By default, authentication on the API is done using basic HTTP authentication. The API is available in XML ("Accept: application/xml ") or JSON ("Accept: application/json ") and can generate HTML when browsing with a browser.

Third-party systems that want to automate DROPS will interact primarily during DROPS' operation phases, namely the creation of releases and the deployment of releases.

All the entities and their metadata managed by the API can be consulted via the URL *http://<server>:<port>/metadata* and the GET method.

6	•	🗆 loca	lhost		\times	+ ~
~	\rightarrow	0	ŵ	0	localh	M\$5252/metadata
	22cml list> + <0 + <0 + <0 + <0 + <0 + <0 + <0 + <0	version ntity na ntity na ntity na ntity na ntity na ntity na ntity na ntity na ntity na ntity na extri < catrii + catri	="1.0" e ime="co ime="te ime="te ime="te ime="te ime="te ime="te ime="co ime="co ime="te idata> bute nam bute nam bute nam bute nam bute nam	ncoding mplate stallLog ategyT apauth' vironm hedule ployme lease" to hee"Par te="Par te="Co	=="IS(ationI Type" " type eentR eentR entIn type= rent & ened ative: reme ked" sed" lease sed"	I-8859-1"?> BMRule" type="configurationIBMiRule" date="2019-09-17T08:36:25,0389Z" version="1" domain="jdbc:rm4os"> type="templateType" date="2019-09-17T08:36:25,0411Z" version="1" domain="jdbc:rm4os"> type="templateType" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os"> type="template" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os"> tet" type="strategyTemplate" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os"> type=templateType" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os"> tet" type=temvironmentRole" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os"> tet" type=temvironmentRole" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os"> terruption" type="deploymentInterruption" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os"> terruption" type="deploymentInterruption" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os"> teleaset=2019-09-17T08:36:25,0418Z' version="1" domain="jdbc:rm4os"> teleaset=2019-09-17T08:36:25,0418Z' version="1" domain="jdbc:rm4os"> terruption" type="deploymentInterruption" date="2019-09-17T08:36:25,0418Z' version="1" domain="jdbc:rm4os"> teleaset=2019-09-17T08:36:25,0418Z' version="1" domain="jdbc:rm4os"> teleaset=2019-09-17T08:36:25,0418Z' version="1" domain="jdbc:rm4os"> teruption" type="deploymentInterruption" date="ca
	-	< </td <td><visibl /metadat ibute > bute nan metadata <col/> <visibl /metadat</visibl </visibl </td> <th>e>true a> ne="Apj colrani RLS_AP e>true a></th> <th>plicat k="5" P_ID <th>xle> ion" type="application" length="1" mandatory="false" listable="true" readonly="false" code="application"> > ble></th></th>	<visibl /metadat ibute > bute nan metadata <col/> <visibl /metadat</visibl </visibl 	e>true a> ne="Apj colrani RLS_AP e>true a>	plicat k="5" P_ID <th>xle> ion" type="application" length="1" mandatory="false" listable="true" readonly="false" code="application"> > ble></th>	xle> ion" type="application" length="1" mandatory="false" listable="true" readonly="false" code="application"> > ble>
		+ <attri + <attri + <attri + <attri - <attri - <attri< td=""><td>bute nam bute nam bute nam bute nam bute nam bute nam</td><th>ne="abs ne="Clo ne="isn ne="Cor ne="Rel n colrank</th><th>solute sed t ode nmer ease</th><th>order" type="string" length="255" mandatory="false" listable="true" readonly="false" code="absoluteorder"> he" type="date" mandatory="false" listable="true" readonly="false" code="closuredate"> ype="boolean" length="1" mandatory="false" listable="true" readonly="false" code="sinde"> it" type="string" length="4000" mandatory="false" listable="true" readonly="false" code="comment"> Description" type="string" mandatory="false" listable="true" readonly="false" code="true" code;="code="comment"></th></attri<></attri </attri </attri </attri </attri 	bute nam bute nam bute nam bute nam bute nam bute nam	ne="abs ne="Clo ne="isn ne="Cor ne="Rel n colrank	solute sed t ode nmer ease	order" type="string" length="255" mandatory="false" listable="true" readonly="false" code="absoluteorder"> he" type="date" mandatory="false" listable="true" readonly="false" code="closuredate"> ype="boolean" length="1" mandatory="false" listable="true" readonly="false" code="sinde"> it" type="string" length="4000" mandatory="false" listable="true" readonly="false" code="comment"> Description" type="string" mandatory="false" listable="true" readonly="false" code="true" code;="code="comment">
entity	lis	t car	i be c	consi	ulte	a via the URL http:// <server>:<port>/data/<entity></entity></port></server>

	6	•	release List	× + 、	/								
	÷	\rightarrow	0 @	O localhost 5252	/data/release								
ŝ	#	parent	creat	tiondate	relativeorder	isclosed	number	application	absoluteorder	isnode	comment	text	is
	10		Thu May 04 12	:26:04 CEST 2017			1.0.0.0	3		0		WRKITM 17050300	0
	11		Thu May 04 13	25:43 CEST 2017			2.0.0.0	3		0		WRKITM 17050001	0
	12		Thu May 04 16	08:36 CEST 2017			3.0.0.0	3		0		WRKTKT4 17050002	0
	13		Thu May 04 17	:37:33 CEST 2017			4.0.0.0	3		0		emergency	0
	14		Thu May 04 17	:50:47 CEST 2017			5.0.0.0	3		0		emergency 2	0
	15		Thu May 04 19	02:39 CEST 2017			5.0.0.1	3		0		xevxev	0
2	16		Thu May 04 19	03:49 CEST 2017			5.0.0.2	3		0		Description_to_change	0

or the URL *http://<server>:<port>/data/<entity>/<id>* for a particular entity ID.



🔁 <table-cell-columns> 🖂 release Data</table-cell-columns>	× + ~
$\leftrightarrow \rightarrow \circ \ \omega$	O localheat 5252/Sata/release/10
release #10	
Modification: Thu May 04 1	2-33-26 CEST 2017
 privati indiate: Thu Ma relativeceder: NULL indecked: NULL incleased: NULL inschenet: NULL insocher 0 application: 3 absolutereder: NULL insocher 0 comment: NULL insarchived: 0 states: NULL order: 1 	y 04 12:26:04 CEST 2017

Access to a complete documentation that gives a description of all the CRUD operations available on DROPS' Public REST API. To do so, add /*drops/api/doc* to your DROPS Server URL. The schema of each entity is also dynamically documented and referenced when they are used.

If you created customized web services, you can also add specifications for them. To do so, add in the **drops.restapi** bundle and in the *specs/custom* folder, a dedicated JSON file that contains only the **paths** node for each one of them.

A Swagger web interface is also served by the DROPS Server, that needs authentication. To access the interface, add */swaggerui* to your DROPS Server URL.

🕕 Note

It is necessary to install Maven to use this feature, as the Swagger UI is fetched by a Maven dependency.

The classic CRUD methods apply to entities:

- GET: fetch the entity list or a particular entity instance
 - GET http://server:5252/data/release reading the list of entities releases
 - GET http://server:5252/data/release/10 read release entity with identifier 10

- **POST**: create a new entity. The creation parameters are in the body of the request
 - POST http://server:5252/data/release

*body: increment=0&text=Automated+release&application=3&number=*GEN* – creation of a new empty release in the application with ID 3, the description of the release will be "Automated release", the release number will be generated automatically by incrementing the major number of the current counter

Note The new generated ID is available in the answer.

- PUT: update an entity
 - *PUT http://server:5252/data/release/10?text=New+description* update description of the release with ID 10
- DELETE: delete an entity
 DELETE http://server:5252/data/release/10 delete the release entity with ID 10

It is possible to filter the result lists with lists of criteria:

 GET http://localhost:5252/data/release?criteria=<and><equals attribute="application" value="3"/><equals attribute="number" value="1.0.0.0"/></and>

get the list of releases whose number is 1.0.0.0 and for the linked application with ID 3 $\,$

🖷 🖷 🗖 release List	× + . ~	
ϵ \rightarrow \circ $rac{1}{2}$	localhost:5252/data/release?criteria=%3Cand%3E%3Cequals%20attribute=%22application%22%20value=%223%22/%	3E963
# parent creation 10 Thu May 04 12:2	ndate relativeorder isclosed number application absoluteorder isnode comment text	7050

2.2 Using DROPS' Public REST API to create releases

The creation of a release is done in several phases:

- Opening/Creating the release
- Creating the import instance and add import strategies
- Running the import instance
- Retrieving the result of the import instance

2.2.1 Opening the release

If the release already exists then the ID can be retrieved using a GET request, but if the release doesn't exist yet, it must be created. This is a POST request that includes the necessary parameters in the body of the release entity.



			2.0
POST	~	http://localhost:5252/data/release	Send request
Headers >			
Basic auth >			
Request body~			
	Туре	Custom	×
tesponse (0.226s) -	http://local	host.5252/data/release	
200 ok			
Headers >			

- **increment**: the part of the release number to increment when the release number is automatically generated
- text: description of the release
- release: the ID of the application this release relates to
- number: number of the release to be created or *GEN for automatic generation using the increment part

The HTTP response 200 confirms that the release was created successfully. The ID of the new release is available in the response body.

2.2.2 Creating the import instance

First of all, you have to create the import instance on the previously-created release using a POST request on the **importExecution** entity or by using the **/import** endpoint.

Public REST API v24.0 2 DROPS' Public REST API | Operational Guide

POST	http://localhost:5252/import	Send request
Headers >		
Basic auth >		
Request body~		
Туре	Custom	~
release=6324resetrelease	=1sexecdeploy=0sfiltertype=2	
Response (0.024s) - http://local	host.5252/import	
200 ok Headers >		
<pre>{ "importExecution": { "id": "816", "reserrelease": "1", "release": "632", "initializationonly: "0 "startdate": "2019-09-17 "execdeploy": "0", "filtertype": "2", "result": "2", "status": "0", "deleted": "0", "statussesage": { "id": "1", "level": "2", "status": "0", "code": "100-EXC-CRT-I "textievel2": "The new "textievel2": "The new "textievel2": "Import "details": { "count": "0", } } }</pre>	", T14:01:00,05052", 01", ∴ Import Process Instance with the ID-\$1816 has been created.", Process Instance has been created successfully.",	

- release: ID of the release this import instance relates to
- resetrelease: 0 or 1 to empty the release content before executing the import
- **execdeploy**: automatically execute a deployment after executing the import. (extra parameters may be needed)
- filtertype: filter to apply to this import (extra parameters may be needed)
 - 0: do not apply filter
 - 1: file filter
 - 2: custom filter

Most of the time filters are not needed or custom filters are used. To use custom filters, we must add the import strategies of the different components and the possible import parameters.

Public REST API v24.0 Operational Guide | 2 DROPS' Public REST API

POST	~	http://localhost:5252/	/data/importExecutionComponent		Send request
Headers >					
Basic auth~					
admin			•••••		Show pass
Request body~					
	Туре	Custom		~	
Response (0.034s) -	http://loca	ilhost 5252/data/importExecut	tionComponent		
200 ок					
200 oK Headers >					

- **component**: ID of the component this filter relates to
- importstrategy: ID of the import strategy to use for this component
- **execution**: ID of the import instance
- externalversion: parameter for the import strategy

The HTTP response 200 confirms that the import instance was created successfully. The ID of the new import execution component is available in the response body.

At this stage the import instance is now configured and ready to run.

2.2.3 Running the import instance

The import instance execution is launched using a POST request on the entity or by using the **/import** endpoint.

Public REST API v24.0 2 DROPS' Public REST API | Operational Guide

POST ~	http://localhost:5252/import/816		
Headers >			
Basic auth~			
admin		•••••	
Request body~			
Type	Custom		~
Response (0.074s) - http://local	host:5252/import/816		
200 ок Headers >			

• **initializationonly**: optional parameter in the body to tell DROPS to launch only import strategies typed as initialization one.

The HTTP response 200 confirms that the import instance execution was launched successfully.

2.2.4 Reading the result of the import instance execution

The execution status of the import instance is available for consultation via a GET request.

Public REST API v24.0 Operational Guide | 2 DROPS' Public REST API

GET Y	http://localhost:5252/data/	importExecution/016	Send reque
Headers >			
Basic auth~			
admin		•••••	Show pa
200 ок			
Headers >			

The HTTP response 200 returns the status and result of the import instance execution.

The different statuses are:

- 0: Prepared
- 1: In progress
- 2: Completed

The different types of results are:

- 0: Succeeded
- 1: Failed
- 2: Not available

🕕 Note

The import execution status is prepared or running until the status is completed. When the status is completed an import execution result is set.

2.3 Using DROPS' Public REST API to deploy

Deploying a release can be done in two ways:

- Transfer of deliverables and deployment at once
- Provisioning deliverables and launching the deployment instance asynchronously

2.3.1 Creating the deployment instance

The deployment instance is created by a POST request to create the deployment instance.



Request		2.0.+
POST ¥	http://demodrops:5252/deploy	Send request
Headers >		
Basic auth≚		
admin		Show password?
Request body~		
Туре	Custom	~
process=7&envi	ronment=16status=16transactionid=TID\$(time)6release=1	
Passance (0.196c)	http://domodrone.5252/doploy	
Response (0. 1005)	- mp.rdemourops.5252/deproy	
<pre>Headers > { "deployment": { "ideployment": { "id": "212", "release": "1", "scheduled": "0", "useinternallog": "result: "2", "instancetype": "1", "process": "7", "rollbackonfail": "revival": "8", "multirelease": " "preparedate": "2 "status": "8", "transactionid": "deleted": "8", "statusmessage": "id": "1", "statusmessage": "id": "1", "status:"8", "textLevel1": "2", "status": "8", "textLevel1": "2", "status": "8", "textLevel1": "2", "status": "8", "count": "8", " "details": { "count": "8", " } } } }</pre>	<pre>"0", "0", "0", "0", "0", "10001457424", 11D001457424", { S-CRT-I01", The new instance with the ID=212 has been created.", Deployment process instance has been created successfully.",</pre>	

- **process**: ID of the process to apply
- **environment**: ID of the target environment

- transactionid: template to apply for generation
- **release**: ID of the release to deploy

The HTTP response 200 confirms that the deployment instance was created successfully. The ID of the new deployment instance is available in the response body.

Once the instance is created and the instance ID read, the instance is launched via a POST or PUT request on the entity or by using the **/deploy** endpoint.



PUT http://demodrops:5252/deploy/212 Send request Headers > Basic auth* admin admin admin Show password? Request body > Response (0.244s) - http://demodrops.5252/deploy/212 200 ok Headers > { ("one closed") """, "reaction" """, "reaction" """, "", "reaction" """, "", "reaction" "", "", "", "reaction" "", "", "reaction" "", "", "", "reaction" "", "", "reaction" "", "", "", "", "", "", "", "", "", "	Request		20+
Headers > Basic auth~ <pre>dmin</pre>	PUT v http://demodrops:5252/	deploy/212	Send request
Basic auth~ <pre>sdmin</pre>	Headers >		
<pre>admin</pre>	Basic auth~		
<pre>Request body> Response (0.244s) - http://demodrops:5252/deploy/212 200 oK Headers></pre>	admin	•••••	Show
<pre>Request body> Response (0.244s) - http://demodrops:5252/deploy/212 200 ok Headers > { fingLopemt*1 { if if</pre>			password?
<pre>Response (0.244s) - http://demodrops:5252/deploy/212 200 oK Headers > { for 1227, rowset: 1; rowset:</pre>	Request body >		
<pre>Response (0.2445) - http://demodrops.5252/deproj/212 200 ok Headers ></pre>	.	- 24.0	
<pre>Paders> /// *********************************</pre>	Response (0.244s) - http://demodrops:5252/deplo	yl/212	
<pre>{ "deployment": { "ide": 1212", "ide": 1212", "ide": 1212", "ide": 1213", "ide": 12</pre>	200 ok Headers >		
<pre>], "resolves-to": "java.util.Collections\$UrmodifiableList", "class": "java.util.Collections\$UrmodifiableRandomiccessList"), "transactionid": "TID001457424") </pre>	<pre>{ "deployment": { "id": "212", "achebuled": "0", "release": "1", "startdate": "2019-00-23707:21:16,04742", "result": "2", "executer": "2", "executer": "2", "executer": "3", "reduals: "0", "reduals: "0", "reduals: "0", "reduals: "0", "nultirelease": "0", "proparentia": "2019-00-23707:14:57,04252", "startdate": "2019-00-23707:14:57,04252", "starts: "1", "language": { "default": "0", "proparentia": "2019-00-23707:14:57,04252", "starts: "1", "language": { "default": "0", "proparentia": "2019-00-23707:14:57,04252", "starts: "1", "language": { "default": "0", "proparentia": "2019-00-23707:14:57,04252", "starts: "1", "language": { "default": "0", "proparentia": "2019-00-23707:14:57,04252", "starts: "1", "language": { "default": 0, "item": [("item": [("item:: [("item:: [], "reference:: "/c", "class:: "java.util.concurrent.CopyOnerite), { "resolves-to": "java.util.collections\$Urmodifiables: }), "rensuctionid": "TED005457424" }) // "rementerice: "TED005457424" }) // "rementerice: "TED005457424") // "rementerice: "TED005457424" </pre>	ArrayList" ArrayList" WowAccessList"	

The result and monitoring of the deployment instance is done with a GET request on this same instance.



Request		2 o +
GET V	http://demodrops:5252/deploy/212	Send request
ieaders >		
Basic auth≚		
admin		Show password?
Response (0.091s	- http://demodrops/5252/deploy/212	parties.
Response (0.091s 200 ok	- http://demodrops.5252/deploy/212	

The deployment execution status is prepared or running until it is completed or canceled. When the status is completed a deployment execution result is set.

The different statuses are:

- 0: Prepared
- 1: In progress
- 2: Completed
- 3: Canceled
- 8: Suspended

The different types of results are:

- 0: Succeeded
- 1: Failed
- 5: Succeeded with warnings
- 6: Not available

2.3.2 Running the transfer instance, then the deployment instance

This mode starts with creating a deployment instance just like in the previous deployment mode. This deployment instance must be "transformed" into a Transfer instance (**transfer-only** in DROPS). This operation is done using a PUT request on the new instance.



Request			2 o +
PUT ¥	http://192.160.99.101:5252/da	ata/deployment/231	Send request
Headers >			
Basic auth ≤			
admin	••••	•	Show password?
Request body~			
Туре	Custom	>]
instancetype=2	scheduled=04schedulelinked=0		
Response (0.055s)	http://192.168.99.101.5252/data/deplo	ymen//231	
Response (0.055s)	http://192.168.99.101:5252/data/deplo	yment/231	
Response (0.055s) 204 No Content	http://192.168.99.101.5252/data/depio	yment/231	

The parameters are set in the request body:

- instancetype:
 - 2: transfer only
 - 3: execute only
- scheduled: 0 or 1 if schedule is used



The instance can now be executed using POST or PUT. Only the provisioning (the transfer of deliverables) will be executed.

The execution of the transfer instance can be followed via a GET method, as usual, on the deployment instance. Statuses and results are available in the response body.

Now, you have to create the deployment instance that will execute the installation phase of the deployment. As a reminder, the previous deployment instance only corresponds to the provisioning phase, that is to say, the transfer of the artifacts.

This creation is done using a new POST request on the deployment instance where the value of the instance type field is 3 and the execution mode field is set to 1.

Public REST API v24.0 2 DROPS' Public REST API | Operational Guide

Request				2.0.4
POST V	http://192.168	.99.101:5252/dep	loy/231	Send request
Headers >				
Basic auth~				
admin		•••••		Show
Request body v				password
Туре	Custom			~
.,,,,,,				
		A A PARAL AL AL MARK		
Response (0.102s)	 http://192.168.99.1 	01:5252/deploy/231		
Response (0.102s)	 http://192.168.99.1 	01:5252/deploy/231		
Response (0.102s)	- http://192.168.99.1	01:5252/deploy/231		
Response (0.102s) 200 ок	- http://192.168.99.1	01:5252/deployr231		
200 oK Headers >	- http://192.168.99.1	01:5252/deploy/231		
200 OK Headers >	- http://192.168.99.1	01:5252/deploy/231		
200 OK Headers >	- http://192.168.99.1	01:5252/deploy/231		
Response (0.102s) 200 ok Headers > ("deployment"; { "sd"; "235",	- http://192.168.99.1	01:5252/deploy/231		
Response (0.102s) 200 OK Headers > ("deployment" (("id" ("235", "scheduled" (")")	- http://192.168.99.1	01.5252/deploy/231		
Response (0.102s) 2000 OK Headers > { "deployment": { "scheduled": "@" "release": "1", "useichermallog"	- http://192.168.99.1	01.5252/deploy/231		
Response (0.1023) 200 OK Headers > { "deployment" ("sd" "235", "scheduled": "0" "release": "1", "useinternallo": "20	- http://192.168.99.1	01.5252/deploy/231		
Response (0.1023) 200 OK Headers > ("deployment": { "acheable": "a" "reclease": "2", "ussistermallog" "result": "e",	- http://192.168.99.1	e",		
Response (0.102s) 200 OK Headers > { "deployment": { "advalued": "2", "scheduled": "2", "scheduled": "2", "sackternallog" "tartdate": "2", "esucite": "2", "sackternallog"	- http://192.168.99.1	01.5252/deploy/231		
Response (0.102s) 200 OK Headers > ("deployment": { "deployment": { "deployment": { "scheduled": "W" "release": "L", "scheduled": "W" "release": "L", "scheduled": "L", "instructure": "L", "instructure": "L", "collariesenfall"	- http://192.168.99.1	01.5252/deploy/231		
Response (0.102s) 2000 OK Headers > { "deployment"; { "ide"; "35", "125", "scheduled"; "9" "result"; "9", "startder"; "2", "startder"; "2", "startder"; "2", "startder"; "2",	- http://192.168.99.1 , ; "1", ;========:::::::::::::::::::::::::::::	e",		
Response (0.1023) 200 OK Headers > { "deployment": { "defloyment": { "release": "1", "sasinternallog" "result": "2", "instructype": "rolliackonfail" "process": "2", "process": "2", "process: "2", "process: "2", "process: "2", "process: "2", "process: "2", "process: "2", "process: "2", "process: "2", "process: "2", "process	- http://192.168.99.1 , "1", s0-24787:11:29,0006 "2", ; "9",	01.5252/deploy/231		
Response (0.102s) 200 OK Headers > { "deployment": { "drif": "253", "acheduled": "27, "esuiction": "27, "resuccer": "27, "rotancetope": "acheduled": "27, "rotancetope": "acheduled: "acheduled": "acheduled": "acheduled: "acheduled": "acheduled: "acheduled": "acheduled: "acheduled: "acheduled: "acheduled: "acheduled: "acheduled: "acheduled: "acheduled: "acheduled: "acheduled: "a	- http://192.168.99.1 , "1", 59-00-24707:11:29,0006 "2", ; "9",	01.5252/deploy/231		
Response (0.102s) 200 OK Headers > ("deployment": { "deployment": [%] "release": "%] "release": "%] "result": "%", "antrotector": "%", "retarctor": "%", "retarctor": "%", "retarctores": "%", "reclass": "%", "medicaless": "%",	- http://192.168.99.1 , ; "x", 29-00-24107:11:29,0024 "2", ; "9",	01.5252/deploy/231		
Response (0.102s) 2000 OK Headers > { "deployment": { "deployment": ["result": "0", "result": "0", "result": "0", "result": "0", "result": "0", "matticelesse": 1", "resident": "1", "resident": 1", "resident": "1", "resident": "1", "resident": 1", "resident": 1	- http://192.168.99.1 , "1", 59-00-24707:11:29,0006 "2", ; "9", -00-24707:11:44,00622"	e",		
Response (0.1023) 200 oK Headers > { "deployment": { "ideployment": { "advebled": "0" "releast": "1", "useinternallog" "tartoter: "20, "executer": "21, "result": "0", "executer": "21, "result": "0", "executer": "21, "result: "0", "executer": "21, "result: "0", "executer": "21, "result: "0", "executer": "21, "result: "0", "executer": "21, "result: "0", "executer": "21, "result: "1", "result: "1",	- http://192.168.99.1 , "1", se-ee-24787:11:29,0006 "2", 1"8", "8", -89-24787:11:44,00622" 1"8",	01.5252/deploy/231		
Response (0.102s) 200 OK Headers > ("deployment": { "id": 233", "scheduled": "9", "redease": "20, "redease": "21, "rediate": "20, "matinetppe": "proparet": "21, "rediate": "2009 "scheduleInkeet" "proparet": "2009 "scheduleInkeet" "proparet": "2009 "scheduleInkeet" "proparet": "2009 "scheduleInkeet" "proparet": "2009 "scheduleInkeet" "proparet": "2009	- http://192.168.99.1 - http://192.168.99.1 - "1", - "0", - "0",	<pre>c1.5252/deploy/2.31 c1.5252/deploy/2.31 c3.5 c3.5 c3.5 c3.5 c3.5 c3.5 c3.5 c3.5</pre>		
Response (0.1023) 200 OK Headers > ("deployment": { "deployment": { "deployment": ["scheduled": "@" "realesse": "1", "santenerslig" "startdate": "2", "realesse": "1", "realesse": "1", "realesse": "2", "realesse": "2", "matterelsse": "2", "matterelsse": "2", "matterelsse": "2", "matterelsse": "2", "matterelsse": "2", "matterelsse": "2", "matterelsses": "2", "", "", "", "", "", "", "",	- http://192.168.99.1 , "1", 29-00-24787:11:29,0026 "2", 1 "0", "0-0-24787:11:44,00622" "0", 2020-00-24787:02:17,03	01.5252/deploy/231		
Response (0.102s) 200 OK Headers > { "deployment": { "deployment": { "deployment": ["result": "B" "result": "B' "result": "B' "res	- http://192.168.99.1 , '1";, 59-09-24787:11:29,0006 "2", '8", -09-24787:11:44,00622" '1"e", -09-24787:12:29,0005 "100000217306"	<pre>c",</pre>		
Response (0.102s) 200 OK Headers > { "deployment": { "id": 253", "achealued": "0", "realesse": "1", "realesse": "1", "realesse": "1", "realesse": "1", "realesse": "1", "realesse": "2", "realesse": "20, "realesse": "20, "realesse: "20, "re	- http://192.168.99.1 , "1", 54-00-24707:11:29,0006 "2", ; "0", "0", -00-24707:11:44,00822" ; "0", 5259-00-24707:02:17,00	01.5252/deploy/231 E", , 262",		

The response contains a **child** field, which is the ID of the new linked deployment instance that corresponds to the installation phase of the deployment.

The execution of this new instance can be followed, as usual, with a GET request.



057			
UCI Y	http://192.168.9	9.101:5252/data/deployment/232	Send request
Headers >			
Basic auth∽			
admin			Show password?
200 oK Headers > { "deployment": { "ddr": "232", "ddre": "233", "parent": "233", "scheduled": "27" "scheduled": "27"	24707:28:15,01732",		

The status and result fields have the same meaning as before.

2.3.3 Rolling back

Rollback is only possible on a deployment instance that has already been executed. To implement the rollback, you need the ID of the deployment instance to revert. In the case of a two-part deployment, transfer + installation, you also need the deployment instance ID corresponding to the ID of the child install instance.

To implement the rollback, you must create a deployment instance using the **/rollback** endpoint and the POST method.



Request			
POST V	http://192.168.99.	101:5252/rollback/232	Send request
Headers >			
Basic auth Y			
admin		•••••	Show password?
Request body~			
Туре	Custom		~
executer=1			
assonse (0.08s) -	http://102.162.00.101-526	Okallhack/939	
response (o.oos) -	map of the tool and to take	ALL ON DECKE DE	
200 ок			
leaders >			
icolocia z			
(
"deployment") {			
"perent": "231",			
"scheduled": "#",			
"release": "1",			
"useinternallog"			
	- r -,		
"rollbackinstance	"1", "1"233",		
"rollbackinstance "startdate": "201	"1", "1"233", 9-09-24107:27:59,00962",		
"rollbackinstance "startdate": "200 "result": "0",	"1"233", 9-00-24707127150,00062",		
"rollbackinstano "startdate": "200 "result": "0", "executer": "1",	""", "1 "233", 9-89-24787127159,80962",		
"rollbackinstance "startdate": "200 "result": "0", "executer": "1", "instancetype": " "rollbackonfail"	''''''''''''''''''''''''''''''''''''''		
"rollbackinstano "startdate": "20 "result": "0", "executer": "1", "instancetype": " "rollbackonfail", "process": "2",	"1"; "1"233", 9-00-04787:27:59,89962", "9", "9",		
<pre>"rollbackinstance "startdate": "20" "result": "0", "esecuter": "1", "instancetype": " "rollbackonfail": "process": "7", "result": "0",</pre>	"1", "1"233", 9-00-34787127159,00062", 3", "9",		
<pre>"rollbackinstance "startiste": "De "result: "0", "escult: "0", "instancetype": " "rollbackonfail" "process": "7", "revival": "0", "esclutetaee";</pre>	177, 172337, 8-80-20187127159,000027, 37, 1791, 97,		
<pre>"rollbackinetarco "startiste": "Det "result": "Det "result": "D", "secuter": "T", "starcetype": " "rollbackonfail" "process": "T", "multinetarce": "T", "preparer": "T",</pre>	"T", " 1237", #-00-24787127159,00042", "", "9",		
"Hilbackinstance" "sartdate": "De "resulter": "A", "secuter": "L", "setancetype": " "relibackonfail" "process": "L", "multirelesse": "2009 "reducter": "2009	11., 1237, 3-86-34787127159,888827, 37, 197, 87, 86-34787128113,8827327,		
"reliactiontance Intertate": "De Insult": "G', "executer": "1", "intimationful" "process": "7", "redual": "6", "multiviesse": "1", "enduate": "2020 "schedulliser"	110, 1237, 10-80-24787127159,899827, 21, 1197, 81, 60-24787128125,857327, 1197,		
"Hilbackinstance "startdate": "be "nesuler": "a", "secoter": "a", "startortype": "Process": "a", "neutoinsta": "b', "neutoinsta": "b', "propared': "b', "proparedite": "b', "proparedite": "b', "proparedite": "b', "proparedite": "b', "proparedite": "b', "proparedite": "b', "proparedite": "b', "proparedite": "b', "proparedite": "b',	12., 1237, 3-80-24787(27)59,899827, 37, 197,		
"hellbackinstance "hearder": "De- "nesole": "G", "escole": "1", "sotancetype": " "hellbackonfell" "process": "G", "heltinbass": "G", "heltinbass": "C", "sothedullried": " "properede": "Compared "schedullried": "Schedullried": "Schedullried: "Schedu	-10, -1237, 8-88-34787127159,800827, 37, 197, 87, 89, 89, 40-34787128115,8827327, 197, 818-80-34787127159,804827, 1981-80-34787127159,804827,		
"rollackinters" Intertates" Tak Intertates" Tak Interceptor Interc	12., 1237, 4-80-24787:27:59,899627, 37, 197, 87, 89,-04787:28:15,867327, 197, 89-06-24787:27:59,864327, 197, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:59,864327, 198-06-24787:27:57:57,864327, 198-06-24787:27:57,875,864327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,884327, 198-06-24787:27:57,57,57,57,57,57,57,57,57,57,57,57,57,5		
"vollaskintes" (20) "tatetäs" (20) "result" (20) "istatoritye" (20) "sistanoitye" (20) "redual" (20) "redual" (20) "sistanoitye" (20) "sistanoitye" (20) "sistanoitye (20) "si	""", """, """, """, "", "", "", "", "",		
"reliablichter Thartder" 100 Thartder" 100 Thartder Theoder Theoder Theoder Theoder Theoder Theoder Theoder Theodel T	11., 1237, 3-85-34787127159,888821, 31, 197, 87, 89, 89, 89, 197, 19		
"vollaskintes" (20) "tatetäsi" (20) "vessit" (7) "settenetype" (7) "vollaskontäi" (7) "vollaskontäi" (7) "vessit" (7) "valtivääss" (1) "setessiä (20) "tateut (2) "tateut (2)	 "1", "233", -00-24707127159,00002", "9", 00-24707128153,05732", "9", 00-24707128153,05732", "9", 10", 		
"vollaskinteen" "tatetdes" 198 "result" 197, "sectors" 197, "sistanceste" 197, "rolaskinte", "process", "process", "process", "schedublisse", "process", "schedublisse", "property 197, "schedublisse", "schedublisse	-10, (1237), 8-08-24787(27)59,000021, 31, 10,		
"vollaskinisten istartden" 100 Inselit" (%) Inselit" (%	(", "233", a-86-34787(27)59,80082", 3", "9", 80-34787(28)(5,85782", "9", 80-34787(28)(5,85782", "9", 80-36-34787(27)(59,86482", "french Language", [
"vollaksimise" tak "textelet" tak "vesdit" "", "sectorelyn" 1 "potesticityn" 1	(10, 1237), 10-00-24707(27):59,000027, 10, 10, 10, 10, 10, 10, 10, 10		
"reliablichterier" "htterteist" 198 "result": 197, "result": 197, "reliablicht": 198, "reliablicht": 197, "reliablicht": 198, "reliablicht": 198, "reliablicht": 198, "reliablicht": 198, "reliablicht": 198, "reliablicht": 198, "reliablicht": 198, "steisbelicht": 198, "stei	(10, 1237), 3-08-24787(27)59,000021, 31, 107, 107, 107, 107, 107, 108-06-24787(27)59,004221, 107,006-24787(27)59,004221, 170006-24787(27)59,00421, 170006-247875,00000, 170006-24785,000000,00000,00000,00000,00000,0		
"vollaskiniste istartiskini "Be Inselitin "P", Inselitin "P", Inselitin "P", Inselitin "P", Inselitin "P", Inselitin "P", Inselitin "P", Inselitin" "Properties" Inselitin" "Properties" Inselitin" "Properties" Inselitin" "Properties" Inseliting" (Inseliting") (Inselition" I	(10, 1233), =-80-34787(27)59,000021, 31, 107, 97, 90-34787(28)55,007321, 107, 90-34787(28)59,004821, 107, 107, 108, 109,		
"vollaskintes" '20 "tatetdes" '20 "vesult" "", "sectorelyse" ' "ollaskontes" '7', "vollaskontes" '70 "rokalt" '70, "sectore" '70, "sec	 11., (1.233°), (3.468-24787);27:59,00002°, (1.269-24787);27:59,00002°, (1.269,0002°), (1.269,000°		
"rollacidnices" '20 "tat-tdes" '20 "result" '3", "sectores" '3", "rollacionis", "rollacionis", "rollacionis", "propers", '3", "rollacionis", "propers", '3", "rollacionis", "propers", '3", "rollacionis", "propers", '3", "rollacionis", "propers", '3", "rollacionis", "propers", '3", "rollacionis", "propers", "statum", "stat	<pre>""""""""""""""""""""""""""""""""""""</pre>		
"vollaskinteen" "tatetden" 100 "result" 10", "sectorer" 11", "settorerype": "vollaskintel", "process", 100 "propert" 12", "endet", 100 "scheduleisket", "scheduleisk	<pre>""""""""""""""""""""""""""""""""""""</pre>	meritawyny.List"	
"vollaskintes" '20 "terdets" '20 "vesdit" "1", "sectorsys" '1", "sectorsys" '1", "vollaskontai" '20 "vollaskontai" '20 "schedulines",	<pre>""""""""""""""""""""""""""""""""""""</pre>	netteknayulat"	
<pre>"validacidnicad" " Tarteidar" Tar "result") "a", "nescult") "a", "instancester" 12", "result") "a", "result" "a", "result""a", "result" "a", "result" "a", "result""a", "result""a", "result""a", "result""a", "result""a", "</pre>	<pre>""""""""""""""""""""""""""""""""""""</pre>	neriterray.List"	
"vollaskinteen" "tarteks" 'Be "result" 'B', "sectors" 'S', "sectors" 'S', "reduct" 'B', "reduct" 'B', "reduct" 'Bes "schedulsise" 'Bes "s	<pre>""""""""""""""""""""""""""""""""""""</pre>	narita/way.List"	
<pre>"vollastintes" '20 "tatetdes" '20 "result's "0", "sectoredres" '20 "result's "0", "sectoredres" '20, "results "0", "sectoredres" '20, "sector</pre>	<pre>""""""""""""""""""""""""""""""""""""</pre>	neritekrayılışt"	
<pre>"vollaskinteen" "startden" "startden" "startden" "vollaskinteen" "vollaskinteen" "vollaskinte" "vollaskinte" "propert" "propert"" "propert"" "propert"" "propert"" "propert"" "proper</pre>	<pre>""""""""""""""""""""""""""""""""""""</pre>	neritekrayılışt"	
"vollaskintes" "tartdes" 'Be "result'n 'B', "sectors" 'S', "sectorspe": "vollaskontai" "proces" 'S', "extendedines" "properties" "scheduldiset" "properties" "scheduldiset" "sched	<pre>""""""""""""""""""""""""""""""""""""</pre>	naribakway.List"	
"vollastinisti" 120 "transist" 120 "result" 14", "nesoter" 14", "socores" 14", "process" 17", "reduct" 120, "process" 17", "reduct" 120, "schedultise", 120, "schedulti	<pre>""""""""""""""""""""""""""""""""""""</pre>	naritakrrayılıst" maritakrrayılıst" modifiablesist",	
<pre>"villascinites" " The result's " 'A' " result's " 'A' " result's " 'A' " result's 'A' " res</pre>	<pre>""""""""""""""""""""""""""""""""""""</pre>	naritakrvayılıst" maritakrvayılıst" madifiablenist", ablahandamuccesulut"	
<pre>"vollascinteers" "startdes" "startdes" "startdes" "startdes" "startdes" "startdeste" "vollasciontal" "process" "prodest" "prodest" "prodest" "prodest" "prodest" "prodest" "prodest" "prodest" "startdeste" "prodest" "prodes</pre>	<pre>""""""""""""""""""""""""""""""""""""</pre>	Haribak-Hay,List" Haribak-Hay,List" Hodificable.cot", ablehandow.coss.list"	
<pre>"values.in" take to the second s</pre>	<pre>""""""""""""""""""""""""""""""""""""</pre>	neritekrnajulat" neritekrnajulat" modifiablevlet", ablahandowiccesulat"	
<pre>"vollastimized "tateddas" 'De "result's 'O", "mesouter's 'D", "instanceyse's 'D' "process', 'D' "propers' 'D', "medual's 'D', "propers' 'D', "mesouter's 'D' "propers' 'D', "mesouter's 'D' "steadilises" "propersites" 'D' "steadilises" "steadilises" "steadilises" "tates" "tates" "tates" "tates" " "tates" " "tates" " "tates" " "trendenter' " 'D' "tates" 'D' "tates"'' 'D' 'D' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot' 'Trensolves-fot'</pre>	<pre>""""""""""""""""""""""""""""""""""""</pre>	naritakrvayılıt" naritakrvayılıt" modifiablesist", ablahandomiccessilıt"	

The response contains a **rollbackinstance** field whose value is the ID of the new deployment instance created for the rollback part of the deployment.

This ID is used to follow the progress of the rollback instance using a GET request on the deployment entity.



GET 🗸	http://192.168.99.101:6252/data/deployment/233	Send request
Headers >		
Basic auth Y		
admin		Show password?
200 ck Headers > (

The status and result fields have the same meaning as before.

2.3.4 Tracking execution instances

Monitoring the execution of a deployment instance is done using a GET request:

• either on the deployment instance entity in order to have the complete information about the instance,

Request	20+
027 V http://192.140.99.101:5252/dets/deployment/233	Send request
Headers >	
Basic auth Y	
admin	Determine (
Response (8.037a) - http://102.158.00.101.5252/data/deployment223	
200	
<pre>{ "indopent": { "indopent": { "indopent": I; "indopent: I; Indopent: I; Indopent: I; Indopent: I</pre>	

• or on the endpoint /deploy to have only the follow-up of execution.



OET w http://192.168.99.101/9282/deploy/283	Send request
Headers >	
Basic auth *	
*****	Show password?
Response (8.854s) - http://102.158.99.101.5252/deploy/233	
Response (2 054s) - http://102.1168.09.101.5252http/kg/233 200 cx: Headers >	

The same logic applies to the import process instances using the /import endpoint.