# Table of Contents

Table of Contents 1

Revision Log 3

Intro 4

Fusion REST alignment 6

BaseResource/ManagedResource 6

Standard Headers 7

Request Headers 7

Response Headers 8

General Resource Controller Behavior 9

Versioning 9

Standard Formats 10

Data Classes & Documentation 10

Resource URIs, Categories & Types 10

authn, authz, session mgmt, trusted component, local user management 11

osd-apx 12

os-deployment-apxs URIs 12

osd-build-plans 13

os-deployment-build-plans URIs 13

Osd-cfg-files 14

os-deployment-install-cfgfiles URIs 14

osd-facility 14

os-deployment-facility URIs 14

osd-ilos 16

os-deployment-ilos URIs 16

osd-index 17

osd-jobs 19

os-deployment-jobs URIs 19

osd-server-scripts 20

os-deployment-server-scripts URIs 20

osd-ogfs-scripts 21

os-deployment-ogfs-script URIs 21

osd-servers 22

os-deployment-servers URIs 22

osd-settings 23

os-deployment-settings URIs 23

osd-zip-files 24

os-deployment-install-zips URIs 24

Altair\_IC Security Categories/Actions & Roles 25

Categories/Actions 25

Roles 25

# Revision Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Comments** |
| 0.1 |  | C. Boeker | Sent out to middleware group for input. |
| 0.2 | 2/14/2012 | C. Boeker | Updated based on middleware group input. |
| 0.3 | 2/21/2012 | C. Boeker | Many changes to align with Fusion REST spec v0.84 |
| 0.4 | 3/7/2012 | C. Boeker | Added uri/category/type table – updated names from osd-\* to os-deployment-\*.  Added runAsSuperUser and serverChanging to server script |
| 0.5 | 3/8/2012 | C. Boeker | fixed a mistake where I had 2 categories & 1 type for server & ogfs scripts instead of vice-versa in the compiled table (the rest of the doc was correct) |
| 0.6 | 3/26/2012 | C. Boeker | Added first-pass iLO resource controller info |
| 0.7 | 10/20/2012 | C. Boeker | Updated with latest info |

# Intro

Altair\_IC will be developed using an HTML5 UI based on the Fusion MVC model talking to a REST back end on top of SA foundation. The REST layer will be “thick” in that it will expose resources that are driven by the needs of the UI (the items that need to be visible to the user) rather than the internal classes of the SA java client (the interface we will be using to drive SA foundation). Conversely we will have the goal of using the SA-provided data as much as possible to avoid extensive recombination of SA data.



The resources/resource controllers needed are:

|  |  |  |
| --- | --- | --- |
| **Category** | **Resources** | **Description** |
| **Index** | Index | We will need to implement as much of the Fusion Index api as is needed to interface with the UI. We want to see if we can provide acceptable search functionality by doing searches of the SA db via the java client rather than building a separate index as we have servers getting inserted into the db at PXE boot time. We may bypass REST purity by having our Index service search the SA db directly and then construct the objects/uris for the resources without going through the other resource controllers (but using common code). |
| **Configuration/Settings** | Settings  Facility | This includes system configuration (ip address, etc.), SA settings, PXE configuration, etc. Certain SA custom attributes are set against the default facility and inherited by all servers. |
| **Servers** | Servers  iLO | We need to expose servers as visible to/described by SA. We also need access to adding a server by iLO information which uses an iLO resource controller to drive the SA iloManager interface. |
| **OS Build Plans** | OS Build Plans  Scripts  Zip Files  Configuration files  APX | We don’t need a DRM-like composite catalog in Altair so it seems to make sense to follow the SA organization of these objects rather than forcing them into the DRM catalog.  Capture & deploy of configuration files are accomplished as build plan steps and we may have a need to access/run APXes for erase disk, nic personalization, and other functionality we add through that mechanism. |
| **Jobs** | Jobs | We want the richness of information and output available about running SA jobs so we won’t look at shoehorning this into something that looks like the CIC task tracker.  Performing a deployment may best be thought of as creating a job given one or more servers and an OSBP or SW Policy, so rather than being an action on a server or an OSBP/SWPolicy it is creation of a new job with those things as inputs. |
| **Other** (leveraged from Atlas) | Authentication  Authorization  Local User Management  Backup/Restore  Patch/Upgrade  Logging  . . . | All other RMs are leveraged from Atlas. |
|  |  |  |
|  |  |  |

These interfaces will be fleshed out as we go. We’ll want to keep them as simple and straightforward as possible.

# Fusion REST alignment

Altair-IC has the goal of staying as compliant with the Fusion REST approach as possible. To that end this spec is meant to sync with the Fusion REST spec currently available at <http://fusionwiki.vse.adapps.hp.com/fusionwiki/DecisionsAndGuidelines>

There are some aspects of the Fusion REST spec that will not be supported in Altair-IC 1.0:

* State Change Message Bus (SCMB) support
* Schema (/rest/<resource-type>/schema) support – although the URI should be present with a 501 (Unsupported) response
* Validation support (/rest/<resource-type>/validator & /rest/<resource-type>/<resource-id>/validator) – URIs should return 501
* /rest/activities (Task Tracker)
* single attribute GET/PUT

# BaseResource/ManagedResource

All our resources will be derived from the Fusion BaseResource class. That will keep Altair\_IC in sync with the way Fusion resources are designed. The BaseResource class defines several elements that will be inherited by all Altair\_IC resources:

|  |  |  |
| --- | --- | --- |
| **Element** | **Type** | **Description** |
| uri | String | The canonical URI of the resource |
| type | String | Identify the type of the resource. For example “blade”, “profile”, etc. |
| created | String | Last modification time in ISO 8601 format |
| modified | String | Last modification time in ISO 8601 format |
| healthStatus | String | Valid states not defined in BaseResource definition at this time |
| resourceState | String | Valid states not defined in BaseResource definition at this time |
| category | String | The resource category used both in the UI (index category) and authorization. |
| etag | String | This is an opaque string used to identify whether a resource has changed. In the Altair-IC implementation the etag will be the modified timestamp. |
| name | String | Required of all Fusion managed resources. Not from BaseResource – explicitly add to all Altair-IC resources. |
| description | String | Required of all Fusion managed resources. Not from BaseResource – explicitly add to all Altair-IC resources. |

Definitions in this doc are “logical” definitions, not implementation. For example the class may have getUri and setUri methods defined but no uri member – that’s an implementation detail.

# Standard Headers

## Request Headers

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Header** | **Value** | **Description** |
| Req | Accept | application/json[;version=1.0] | application/json is the only acceptable media type. Version will change over releases and if left off the latest version will be assumed. Return 406(Not Acceptable) if not able to support the requested version. |
| Req | auth | auth-token | The authorization token returned from a login request |
| Req | Content-Type | application/json[;version=1.0] | application/json is the only acceptable content-type. Version will change over time and if left off the latest version will be assumed. Return 406 (Not Acceptable) if incorrect. |
| Opt | Accept-Charset | UTF-8 | per Fusion REST spec this is the only acceptable charset. Return 406 (Not Acceptable) if incorrect. |
| Opt | Accept-Language | en;en-us;… | language(s) accepted for response. Return 406 (Not Acceptable) if no requested languages are supported. If not provided use appliance locale. |
| Req for PUT | If-Match | <etag> | PUT/update will only proceed if etag matches that of resource. Return 412 (Precondition Failed) if it does not match, 403 (Forbidden) if it is not present. If-Match:\* will force overwrite. |
| Opt for GET | If-None-Match | <etag> | If etag matches that of resource return 304 (Not Modified) otherwise return resource. |

## Response Headers

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Header** | **Value** | **Description** |
| Req | ETag | <etag> | Not included on multiple GETs (etags are in the body of each resource). For Altair-IC the etag is the modified timestamp. |
| Req | Vary | Accept | Ensures caches map responses to the specified Accept header. |
| Req for POSTs that create new URI resources | Location | <URI> | required on POSTs that create a new URI-addressable resource |
| Req on auth failures | WWW-Authenticate | <challenge> | Return 401 (Unauthorized) when no auth token.  For Altair-IC this should just mean throwing the correct exception when no auth token present and letting the framework handle the rest. |
| Req for responses w/bodies | Content-Type | application/json[;version=1.0] | application/json is the only acceptable type.  If a version was specified in the GET request header and can be matched then that same version is put into this header, otherwise 406 (Not Acceptable) is returned.  For PUT/POST if a newer version is requested then the response is 415 (Unsupported Media Type). |

# General Resource Controller Behavior

* GET – returns a resource or a list of resources
  + see If-None-Match headers for conditional GETs and return values
  + multi-GETs all support total, count, start, sort and filters parameters as per Fusion REST spec
  + multi-GETs do not have etags in the response header, they are in the body of each returned resource
* PUT – update an existing resource
  + if read-only attributes are in body of request then return 403 (Forbidden) if they don’t match values in resource
  + see If-Match header for ETag usage during update
* DELETE – delete an existing resource
  + Successful DELETEs return 204 (No Content)
* POST – create a new subordinate resource
  + Successful synchronous creates return 201 (Created) and set the Location response header with the URI of the newly created resource
  + Successful asynchronous creates (creating os-deployment-jobs to run build plans or scripts) return 202 (Accepted) and the body is an os-deployment-jobs resource

All request returns that are not in the 200-range have as their body an ErrorMessage entity following the Fusion model.

# Versioning

# Standard Formats

This section details any common formats used in all interfaces. Where possible Altair-IC will follow Fusion format standards.

|  |  |  |
| --- | --- | --- |
| **category/type** | **format** | **comments** |
| date/time | Fusion date/time format | all date/time objects should use the standard Fusion date/time format |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Data Classes & Documentation

Data class definitions are contained in the Altair Javadoc that is available on the Jenkins build system linked into the Sonar builds.

Post Altair 1.0 the REST apis will be published and the documentation for it will be generated from annotations in the code.

# Resource URIs, Categories & Types

The category for a resource will eventually be both the UI & authorization category for the resource type (once Atlas brings those into alignment). The type is what is reported in the resource object.

|  |  |  |
| --- | --- | --- |
| **Resource** | **Category/Type** | **URI** |
| osd settings | os-deployment-settings | /rest/os-deployment-settings |
| servers | os-deployment-servers | /rest/os-deployment-servers |
| ilos | os-deployment-ilos | /rest/os-deployment-ilos |
| build plans | os-deployment-build-plans | /rest/os-deployment-build-plans |
| server scripts | os-deployment-scripts | /rest/os-deployment-server-scripts |
| ogfs scripts | os-deployment-scripts | /rest/os-deployment-ogfs-scripts |
| zip files | os-deployment-install-zips | /rest/os-deployment-install-zips |
| configuration files | os-deployment-install-cfgfiles | /rest/os-deployment-install-cfgfiles |
| jobs | os-deployment-jobs | /rest/os-deployment-jobs |
| apxs | os-deployment-apxs | /rest/os-deployment-apxs |
| Facilities | os-deployment-facility | /rest/os-deployment-facility |

# authn, authz, session mgmt, trusted component, local user management

These components will be leveraged from Foundation Services. Altair-IC will add Altair-specific roles to the db as well as add Altair-specific categories & actions (CRUD+-only for actions) to the list of supported actions.

# osd-apx

APXes are not tied into the Altair index.

## os-deployment-apxs URIs

The index used is the SA apx id.

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-apxs/<apx id> | Std headers | OSDAPX in JSON format | Get detailed information for an apx |
| GET /rest/os-deployment-apxs?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | List of OSDAPX entries as a paginated collection in JSON format | Get the list of APXs on the system |
| PUT /rest/os-deployment-apxs/<apx unique name>/run | Std headers  OSDAPXRunData body | 202 (Accepted)  Body: job id | Run an apx by name |
| PUT /rest/os-deployment-apxs/erasedisk | Std headers  String[] serverUris body | 202 (Accepted)  Body: job id | Run the erase disk apx on the specified list of servers |
| PUT /rest/os-deployment-apxs/personalizeserver | Std headers  OSDPersonalizeServerData[] body | 202 (Accepted)  Body: job id | Run the network personalization apx on the specified servers |
| PUT /rest/os-deployment-apxs/iloadd | Std headers  OSDAddIloData body | 202 (Accepted)  Body: job id | Run the ilo addition apx |

Standard URIs that are not supported:

PUT /rest/os-deployment-apxs/<apx id>

PUT /rest/os-deployment-apxs

POST /rest/os-deployment-apxs

DEL /rest/os-deployment-apxs/<apx id>

GET /rest/os-deployment-apxs/schema

POST /rest/os-deployment-apxs/validator

POST /rest/os-deployment-apxs/<apx id>/validator

# osd-build-plans

These are the OS build plans available in the SA database. The index used is the SA OS build plan id.

## os-deployment-build-plans URIs

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-build-plans/<osbp-id> | Std headers | OSDBuildPlan in JSON format | Get detailed information for an osbp |
| GET /rest/os-deployment-build-plans?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | Paginated collection of OSDBuildPlan in JSON format | Get detailed information for a list of osbps |
| PUT /rest/os-deployment-build-plans/<osbp-id> | Std headers  OSDBuildPlan body of post (JSON) | Body: updated OSDBuildPlan JSON | Update an osbp in the SA db |
| PUT /rest/os-deployment-build-plans/<osbp-id>/run | Std headers  String[] serverIds | 202 (Accepted)  Body: job URI JSON | Run an OSBP against a list of servers |
| PUT /rest/os-deployment-build-plans/<osbp-id>/personalizeandrun | Std headers  OSDPersonalizeServerData[] body | 202 (Accepted)  Body: job URI JSON | Run an OSBP against a list of servers with personalization |
| POST /rest/os-deployment-build-plans | Std headers  OSDBuildPlan (JSON) body | Body: OSDBuildPlan JSON | Add an osbp in the SA db |
| DEL /rest/os-deployment-build-plans/<osbp-id> | Std headers | HTML response only | Delete a single osbp from the SA db |

Standard URIs that are not supported:

PUT /rest/os-deployment-build-plans

GET /rest/os-deployment-build-plans/schema

POST /rest/os-deployment-build-plans/validator

POST /rest/os-deployment-build-plans/<osbp-id>/validator

# Osd-cfg-files

These are the configuration files available in the SA database. The index used is the SA configuration file id.

## os-deployment-install-cfgfiles URIs

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-install-cfgfiles /<cfgfile-id> | Std headers | OsdCfgFile in JSON format | Get detailed information for a config file |
| GET /rest/os-deployment-install-cfgfiles?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | Paginated collection of OsdCfgFile in JSON format | Get detailed information for a list of config files |
| PUT /rest/os-deployment-install-cfgfiles /<cfgfile-id> | Std headers  OsdCfgFile body of post (JSON) | Body: updated OsdCfgFile JSON | Update a cfg file in the SA db |
| POST /rest/os-deployment-install-cfgfiles | Std headers  OsdCfgFile (JSON) body | Body: OsdCfgFile JSON | Add a cfg file in the SA db |
| DEL /rest/os-deployment-install-cfgfiles/<cfgfile-id> | Std headers | HTML response only | Delete a single cfg file from the SA db |

Standard URIs that are not supported:

PUT /rest/os-deployment-build-plans

GET /rest/os-deployment-install-cfgfiles/schema

POST /rest/os-deployment-install-cfgfiles/validator

POST /rest/os-deployment-install-cfgfiles/<osbp-id>/validator

# osd-facility

Facilities are not tied into the Altair index.

## os-deployment-facility URIs

The index used is the SA facility id. Altair uses a single facility with id assumed to be 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-facility/<facility id> | Std headers | OSDFacility in JSON format | Get detailed information for a facility |
| GET /rest/os-deployment-facility?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | Paginated collection of OSDFacility in JSON format | Get the list of facilities on the system (should be 1 facility) |
| PUT /rest/os-deployment-facility/<facility id> | Std headers  OSDFacility body | 200 (OK)  Body: OSDFacility | Update facility data (facility name and custom attributes) |

Standard URIs that are not supported:

PUT /rest/os-deployment-facility

POST /rest/os-deployment-facility

DEL /rest/os-deployment-facility/<facility id>

GET /rest/os-deployment-facility/schema

POST /rest/os-deployment-facility/validator

POST /rest/os-deployment-facility/<facility id>/validator

# osd-ilos

iLOs are modeled using the iLOManager in SA. This resource controller exposes the iLOManagers in SA. The index is the SA iLOManager index.

## os-deployment-ilos URIs

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-ilos/<ilo id> | Std headers | OSDIlo in JSON format | Get detailed information for a iLO |
| GET /rest/os-deployment-ilos?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | Paginated collection of OSDIlo in JSON format | Get detailed information for a list of iLOs |
| POST /rest/os-deployment-ilos | Std headers  Optional header: addstyle=old  OSDIlo in body of post (JSON) | 202 (Accepted)  Body: OSDUri | Add an iLO in the SA db.  Default method is to use addIlo apx but header can indicate to use SA add ilo that does not bring the server into maintenance. |
| DEL /rest/os-deployment-ilos/<ilo-id> | Std headers | HTML response only | Delete a single ilo from the SA db |

Standard URIs that are not supported - all return 501 (Unsupported):

PUT /rest/os-deployment-ilos/

GET /rest/os-deployment-ilos/schema

POST /rest/os-deployment-ilos/validator

POST /rest/os-deployment-ilos/<server-id>/validator

# osd-index

The Altair-IC index service will support the REST interface defined by the Foundation Services REST interface. The URIs consumed by the UI will be fully supported. Any other URIs will be optional.

Although the Foundation Service Index service will move to being event-based (the UI will register for and receive events when the index changes) we have made the request to continue to support the polled approach currently in place as there is no eventing in the java client library. Supporting an event-drive index approach will require internal polling of the SA db via the java client library which can be put off until such time as Altair-IC consumes the Foundation Services index service implementation.

The osd-index web service will reside in the osd-services .war file and use web container mappings to redirect the base index URIs to the osd-index service.

All resource controllers will provide a utility function for the index service that provides the list of requested resources in IndexResource format. This function will be leveraged directly by the osd-index code (not via a REST call, not exposed as a REST api) to fulfill index requests for that resource category. This will encapsulate the code that understands the details of a specific resource type into a single project albeit at the cost of each resource controller using IndexResource. Initially these utility functions will be simple but will eventually expand to cover all the possible qualifications of an index request (pagination, search, filtering, etc.).

GET /index/rest/index/resources?<params>

Params:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Values | Required? | Desc |
| start | Integer | Y | Starting index of returned data |
| count | Integer | Y | Number of items to return (-1 means all) |
| category | String (osdserver, etc.) | N  (but effectively Y as Altair doesn’t support not having it in a meaningful way) | Category of data to return |
| Sort | String  <sort attribute>:<ascending|descending> | N | Sort type – default is ascending by name |
| userQuery | String | N | See the ICsp user guide or online help on the search field for a full description – this parameter gets filled with an escaped copy of that string.  The words in the search field are searched for as ANDed search terms. If phrases are in double quotes then those phrases are searched for exact matches of the quoted string(s). Only a subset of all fields are searchable (specified in the user guide). |

Index URIs not supported:

GET /rest/index/suggestions

GET /rest/index/trees

GET /rest/index/trees/\*\*

# osd-jobs

These are the SA jobs available in the SA database. The index used is the SA job id.

## os-deployment-jobs URIs

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-jobs/<job-id> | Std headers | OSDJob data in JSON format | Get detailed information for a job |
| GET /rest/os-deployment-jobs?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | Paginated collection of OSDJob in JSON format | Get detailed information for a list of jobs |

Standard URIs that are not supported:

DEL /rest/os-deployment-jobs

POST /rest/os-deployment-jobs/<job-id>

PUT /rest/os-deployment-jobs

GET /rest/os-deployment-jobs/schema

POST /rest/os-deployment-jobs/validator

POST /rest/os-deployment-jobs/<job-id>/validator

# osd-server-scripts

These are the server scripts available in the SA database. The index used is the SA server script id.

## os-deployment-server-scripts URIs

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-server-scripts/<script-id> | Std headers | OSDServerScript in JSON format | Get detailed information for a script |
| GET /rest/os-deployment-server-scripts?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | Paginated collection of OSDServerScript in JSON format | Get detailed information for a list of scripts |
| DEL /rest/os-deployment-server-scripts/<script-id> | Std headers | HTML response only | Delete a single script from the SA db |
| POST /rest/os-deployment-server-scripts | Std headers  OSDServerScript in body of post (JSON) | Body: OSDServerScript in JSON | Add a script in the SA db |
| PUT /rest/os-deployment-server-scripts/<script-id> | Std headers  OSDServerScript in body of post (JSON) | Body: Updated OSDServerScript in JSON | Update a script in the SA db |

Standard URIs that are not supported:

PUT /rest/os-deployment-server-scripts

GET /rest/os-deployment-server-scripts/schema

POST /rest/os-deployment-server-scripts/validator

POST /rest/os-deployment-server-scripts/<script-id>/validator

# osd-ogfs-scripts

These are the OGFS scripts available in the SA database. The index used is the SA OGFS script id.

## os-deployment-ogfs-script URIs

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-ogfs-scripts/<script-id> | Std headers | OSDOGFSScript in JSON format | Get detailed information for a script |
| GET /rest/os-deployment-ogfs-scripts?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | Paginated collection of OSDOGFSScript in JSON format | Get detailed information for a list of scripts |
| DEL /rest/os-deployment-ogfs-scripts/<script-id> | Std headers | HTML response only | Delete a single script from the SA db |
| POST /rest/os-deployment-ogfs-scripts | Std headers  OSDOGFSScript in body of post (JSON) | Body: added OSDOGFSScript JSON | Add a script in the SA db |
| PUT /rest/os-deployment-ogfs-scripts/<script-id> | Std headers  OSDOGFSScript in body of post (JSON) | Body: updated OSDOGFSScript JSON | Update a script in the SA db |

Standard URIs that are not supported:

PUT /rest/os-deployment-ogfs-scripts

GET /rest/os-deployment-ogfs-scripts/schema

POST /rest/os-deployment-ogfs-scripts/validator

POST /rest/os-deployment-ogfs-scripts/<script-id>/validator

# osd-servers

These are the servers in the SA database. The index used is the SA server id.

## os-deployment-servers URIs

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-servers/<server-id> | Std headers | OSDServer in JSON format | Get detailed information for a single server |
| GET /rest/os-deployment-servers?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | Paginated collection of OSDServer in JSON format | Get detailed information for a list of servers |
| DEL /rest/os-deployment-servers/<server-id> | Std headers | HTML response only | Delete a single server from the SA db |
| POST /rest/os-deployment-servers | Std headers  OSDServer in body of post (JSON) | Body: added OSDServer resource | Add a server in the SA db |
| PUT /rest/os-deployment-servers/<server-id> | Std headers  OSDServer body of post (JSON) | Body: updated OSDServer resource | update a server in the SA db |

Standard URIs that are not supported - all return 501 (Unsupported):

GET /rest/os-deployment-servers/schema

POST /rest/os-deployment-servers/validator

POST /rest/os-deployment-servers/<server-id>/validator

# osd-settings

This controller handles Altair settings that are stored in the OS (DHCP configuration, PXE configuration) or SA (MatrixUser credentials). It also provides a convenient RM to host tool download (media server setup, winpe generation) and file upload (winpe upload) URIs.

## os-deployment-settings URIs

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-settings/<setting-id> | Std headers | OsdSetting in JSON format | Get detailed information for a single setting |
| GET /rest/os-deployment-settings?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | Paginated collection of OsdSetting in JSON format | Get detailed information for a list of settings |
| PUT /rest/os-deployment-settings/<setting-id> | Std headers  OsdSetting body of post (JSON) | Body: updated OsdSetting resource | update a resource in the SA db |
| POST /rest/os-deployment-settings/WinPE | Std headers  file=<uploaded multipart file> | OSDWinPE | Upload a WinPE image to the appliance |

Standard URIs that are not supported - all return 501 (Unsupported):

DEL /rest/os-deployment-settings/<setting-id>

POST /rest/os-deployment-settings

GET /rest/os-deployment-settings/schema

POST /rest/os-deployment-settings/validator

POST /rest/os-deployment-settings/<server-id>/validator

# osd-zip-files

These are the zip files available in the SA database. The index used is the SA zip software object id.

## os-deployment-install-zips URIs

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Input** | **Output** | **Desc** |
| GET /rest/os-deployment-install-zips/<zip-file-id> | Std headers | OSDZipFile in JSON format | Get detailed information for a zip file |
| GET /rest/os-deployment-install-zips?start=<start>&count=<count> | Std headers  Start of 0 and count of -1 gives all | Paginated collection of OSDZipFile in JSON format | Get detailed information for a list of zip files |

Standard URIs that are not supported:

DEL /rest/os-deployment-install-zips

PUT /rest/os-deployment-install-zips

POST /rest/os-deployment-install-zips/<zip-id>

GET /rest/os-deployment-install-zips/schema

POST /rest/os-deployment-install-zips/validator

POST /rest/os-deployment-install-zips/<zip-id>/validator

# Altair\_IC Security Categories/Actions & Roles

## Categories/Actions

These are the security categories needed for controlling access to Altair\_IC resources. Actions in general have been reduced to CRUD+ (create, read, update, delete, use) by Atlas/Fusion/Pulsar and Altair\_IC will follow that approach. The latest REST spec indicates that categories will be used for both UI grouping and authorization so we need one category per UI entity type.

Categories:

* os-deployment-servers
* os-deployment-ilos
* os-deployment-device-groups
* os-deployment-build-plans
* os-deployment-install-cfgfiles
* os-deployment-server-scripts
* os-deployment-ogfs-scripts
* os-deployment-install-zips
* os-deployment-jobs
* os-deployment-admin-tasks
* os-deployment-settings

Actions (for all Categories):

* create
* read
* update
* delete
* use

## Roles

Where it makes sense we will add additional categories/actions to foundation service roles. The roles needed for Altair\_IC and the proposed rights to the Altair\_IC categories and actions are as follows:

**Infrastructure administrator** – full rights to all appliance settings and all SA features

**server administrator** – can manage servers including server groups and run scripts & build plans. This is an Altair addition to the base Atlas roles.

**backup administrator** – can perform backup/restore operations. Used to not put general use credentials in backup/restore scripts as backup/restore functionality is only via REST api in Atlas 1.0/Altair 1.0.

**read only** – can read any information on the appliance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Infrastructure administrator | server administrator | Read only | Backup administrator |
| os-deployment-servers | CRUD+ | CRUD | R | - |
| os-deployment-ilos | CRUD+ | CRUD | R | - |
| os-deployment-device-groups | CRUD+ | CRUD | R | - |
| os-deployment-build-plans | CRUD+ | R | R | - |
| os-deployment-scripts | CRUD+ | R | R | - |
| os-deployment-install-zips | CRUD+ | R | R | - |
| os-deployment-jobs | CRUD+ | R | R | - |
| os-deployment-install-cfgfiles | CRUD+ | CRUD | R | - |
| Os-deployment-settings | CRUD+ | R | R | - |

.