

API-CMDB Integration

This document is presented as a general guide to integration with Aranda CMDB. Operations are presented in detail including request data, parameters, responses, and error.

General

General

Below are the generalities that must be taken into account for the correct functioning of the Aranda CMDB API.

- Aranda CMDB Web Console Supported Version 8.4.3 and Database Version 8.0.62
- Microsoft .NET Framework version 4.5 or higher.
- Operations on the API must be performed with users who have the corresponding permissions assigned from Aranda PROFILE.
- Using the API requires available licenses, which will behave the same as if a specialist were logged into the console.
- Creating CIs using the API complies with the same licensing rules for CIs that are available, as if they were created from the console.
- The CI's registered through the integration will be created in the project that is configured by default in the "Parameter Settings>Default Project" path within the CMDB WEB console, this can be modified.
- The project must have the maps between CI category and type configured in the CMDB WEB console.
- The configured categories must have associated states and their respective flow.
- The CMDB WEB console in the "Settings Settings>Site for Authorization" menu must have the authentication site path configured.
- Querying, editing, deactivating, and reactivating can be done by the IC ID or by the GuID code.
- The following fields now receive text parameter in creation or editing, if the value that is sent already exists it is associated with the same ID that it has in the database, in case it does not exist, it is created and associated with a new ID: CurrentStatus, ProgrammedStatus, Manufacturer, Provider, ProfileCI, CostCenter, Unit, Brand, Model, Building, City, Company, Country, Department, Floor, Site, and NumberDepartment.

Description of Operations

Description of operations

This section details the consumption of the services corresponding to each method. Details such as URI, type of operation, parameters (required and optional) and their types, responses, codes, and error messages.

Session Management

Session Management

This section describes the operations related to the session management (user authentication, token renewal, and logout).

Login

Details of the request:

- URI: api/v8.6/user/login
- Type: POST
- Required headings: Content-Type: application/json

Parameters:

Name	Guy	Obligatory	Description
USERNAME	Text	Yes	User who will log in.
PASSWORD	Text	Yes	Password corresponding to the user.
LANGUAGEID	Number	No	Language in which the session will be recorded. Possible options are: 1. English 2. Spanish 3. Portuguese If it is not provided, it is recorded in English by default.
CONSOLEID	Number	No	
CONSOLEVERSION	Text	No	The version of the console from which you sign in.
DEVICEIMEI	Text	No	IMEI of the mobile device from which you are logging in.
DEVICEBRANDID	Number	No	Brand of the device from which you are logging in. Possible options are: 1. Android 2. iOS
DEVICENAME	Text	No	Name of the mobile device from which you sign in.
DEVICEREGISTRATIONID	Text	No	RegistrationId of the mobile device, used for push messages.
APPLICATIONNAME	Text	No	Name of the external application that consumes the service.
FOREIGN	Text	No	It is not required as long as the name of the application is not submitted. Possible options are: 1. True (if application is foreign) 2. False (If application is not foreign)

Body of the petition:

The request consists of a field-value json array with possible parameters as required. Example for login with CMDB:

```
[
  {
    "Field": "username",
    "Value": "USUARIO_CMDB"
  },
  {
    "Field": "password",
    "Value": "CONTRASEÑA_DEL_USUARIO"
  }
]
```

Example for External Application Login:

```
[
  {
    "Field": "username",
    "Value": "USUARIO_CMDB"
  },
  {
    "Field": "password",
    "Value": "CONTRASEÑA_DEL_USUARIO"
  },
  {
    "Field": "applicationname",
    "Value": "Nombre de aplicación externa"
  },
  {
    "Field": "foreign",
    "Value": "true"
  }
]
```

Answer:

As an answer, a json object is obtained, with the following information:

```
[
  {
    "Field": "userId",
    "Value": "####"
  },
  {
    "Field": "sessionId",
    "Value": ""###AAABBBCCC###"
  },
  {
    "Field": "result",
    "Value": "True"
  }
]
```

Error messages:

Code	HTTP status	Error Message
400	BadRequest	InvalidUserName
400	BadRequest	InvalidPassword
400	BadRequest	InvalidConsoleId
400	BadRequest	InvalidConsoleVersion
400	BadRequest	InvalidDeviceImei
400	BadRequest	InvalidDeviceBrandId
400	BadRequest	InvalidDeviceName
400	BadRequest	InvalidLangId
400	BadRequest	DeviceNamelsRequired
400	BadRequest	InvalidForeignValue
400	BadRequest	InvalidApplicationName
500	InternalServerError	FailureOnLogin

Session Renewal

Details of the request:

- URI: api/v8.6/session/renew
- Type: POST
- Required headings:
- Content-Type: application/json
- Authorization: TOKEN

Parameters:

Body of the petition:

Answer:

There is no response body, the validation of the result of the operation can be performed by analyzing The HTTP code obtained: 200 OK for a successful operation.

Error messages:

Logging Out

Details of the request:

- URI: api/v8.6/user/logout
- Type: POST
- Required headings: Content-Type: application/json
- Content-Type: application/json
- Authorization: TOKEN

Parameters:

It is only required to make a request by sending the authentication token as an http header.

Body of the petition:

Empty.

Answer:

There is no response body, the validation of the result of the operation can be performed by analyzing the http code obtained: 200 OK for a successful operation.

Error messages:

Code	HTTP status	Error Message
400	BadRequest	InvalidToken
500	InternalServerError	FailureCloseSession

IC Management

IC Management

This section describes the operations related to handling CI (create, edit, query, delete, and reactivate).

CI Creation

Details of the request:

- URI: api/v8.6/ci/add
- Type: POST
- Required headings:
- Content-Type: application/json
- Authorization: TOKEN

Parameters:

Name	Guy	Obligatory	Description
Name	Text	Yes	Name of the IC.
ProjectId	Number	No	Project ID.
Guid	Text	No	External ID of the IC. This field is required only if the CI comes from an integration.
Type	Number	No	Type of IC. This field is required only if the CI comes from an integration. The options are: 1. Application 2. BusinessProcess 3. Catalog 4. Computer 5. Hub 6. Monitor 7. Phone 8. Printer 9. Router 10. Server 11. Service 12. SoftwareProduct 13. SystemSoftware 14. Tablet 15. Laptop
CategoryID	Number	No	Category id. If the Type field is sent in the request, this field should not be entered.
AuthorId	Number	No	Author’s ID. If it is not sent, the user token associated with the token used in the request is obtained.
Description	Text	No	Description of the Ci.
Building	Text	No	Name of the building.
City	Text	No	Name of city.
Company	Text	No	Company name.
CostCenter	Text	No	Cost center.
Country	Text	No	Name of the country.
CurrentStatus	Text	No	Name of the current state.
Department	Text	No	Name of the department, e.g. Development.
DepreciationMethod	Number	No	Id of the depreciation method (1: Straight line, 2: Sum of digits).
Floor	Text	No	Floor name, e.g. 25-Basement
GroupId	Number	No	ID of the IC grouping.
Impact	Number	No	Impact of IQ on the organization.
Manufacturer	Text	No	Name of the IC manufacturing company.
Model	Text	No	IC model.
DepartmentNumber	Text	No	Department number.
Price	Decimal	No	IC price.
CIProfile	Text	No	IC Profile.
ProgrammedStatus	Text	No	Name of the future state of the IC.
Provider	Text	No	Name of the IC provider company.

ResidualValue	Decimal	No	Value to be saved from the price of the IC.
Alias	Text	No	User responsible for the CI, can go in the following formats: Domain\User or Usiario@Dominio
Site	Text	No	Place.
Unit	Text	No	IC unit of measurement e.g.: inches
UsefulLife	Number	No	Years of useful life of the IC, for the calculation of depreciation
Brand	Text	No	Brand.
AdditionalField:			
Type	Number	No	Additional field types are: 1.Text 2. Date 3. List 4. Long text 5. Check box 6. Number
Value	-	No	Information to be entered in the additional field, depends on the type of data that the additional field expects
Id	Number	No	Additional field ID

Remarks:

- The CITotal field is not part of the parameters, as it is automatically calculated by the tool and is for internal use.
- The request is case sensitive, so you should leave the parameter names as they are in the parameter table.
- Keep in mind that, when creating correspondences, the category associated with the type must have configured states and transitions.

Body of the petition:

The request consists of a field-value json array with possible parameters as required. Example:

```
{
  "Name": "Nombre del CI",
  "CategoryId": 19,
  "GroupId": -1,
  "CurrentStatus": "En uso",
  "AuthorId": 3,
  "ProgrammedStatus": "Estado Programado Prueba",
  "Model": "GXP1200",
  "Alias": " Usuario@Dominio ",
  "AdditionalField": [
    {
      "Id": ###,
      "Type": #,
      "Value": "Valor"
    }
  ]
}
```

Answer:

As a response, a unique identifier of the CI created is obtained.

Error messages:

Code	HTTP status	Error Message
400	BadRequest	InvalidEventName
400	BadRequest	InvalidUserId
400	BadRequest	InvalidGuld
400	BadRequest	InvalidType
400	BadRequest	InvalidApplicationName
400	BadRequest	InvalidApplicationId
400	BadRequest	InvalidCild
400	BadRequest	InvalidProjectIdByDefault
400	BadRequest	InvalidSubcategoryId
400	BadRequest	InvalidStateName
400	BadRequest	InvalidCompany
400	BadRequest	InvalidIdSettingGeneral
400	BadRequest	CatalogDescriptionIsRequired
400	BadRequest	InvalidCategoryId
404	NotFound	NotFoundStatesOfCMDB
500	InternalServerError	FailureAddCi
500	InternalServerError	FailureAddCiOrGetIdCi

CI Consultation

Details of the request:

- URI: `api/v8.6/ci/{id}?isGuld={isGuld}`
- Type: GET
- Required headings:
- Content-Type: `application/json`
- Authorization: TOKEN

The URI structure of the request is implemented as follows:

- If it's by CI: `api/v8.6/ci/IdCI?isGuld=False`
- If by Guld: `api/v8.6/ci/Guld?isGuld=True`

Parameters:

Name	Data type	Obligatory	Description
Id	Text	Yes	ID of the CI or Guld to be queried
IsGuld	Boolean	Yes	In case the Id that is sent is Guld the value will be “true”, if it corresponds to the Id of a CI then it will be “false”

Answer:

In response, it will return a json object, with the following information:

```
{
  "AcceptDate": null,
  "AdditionalField": [
    {
      "Alias": "ADD_STR_1",
      "Caption": "Prueba característica en 80",
      "Id": 92,
      "IdLookField": 0,
      "Mandatory": false,
      "Order": 0,
      "Type": 1,
      "Value": "Característica en 80",
      "ValueLookField": null,
      "Visible": false
    },
    {
      "Alias": "ADD_STR_2",
      "Caption": "Home",
      "Id": 213,
```

```
        "IdLookField": 0,
        "Mandatory": false,
        "Order": 0,
        "Type": 1,
        "Value": "prueba additional home",
        "ValueLookField": null,
        "Visible": false
    },
    {
        "Alias": "ADD_STR_3",
        "Caption": "Aranda",
        "Id": 217,
        "IdLookField": 0,
        "Mandatory": false,
        "Order": 0,
        "Type": 6,
        "Value": null,
        "ValueLookField": null,
        "Visible": false
    },
    ],
    "Address": null,
    "ApplicationName": null,
    "AssetTag": null,
    "AuthorId": 2017,
    "BackgroundCurrentStatus": "255.255.255",
    "BarCode": null,
    "Brand": "Nueva marca prueba 2",
    "BrandId": "71",
    "Building": null,
    "BuildingId": null,
    "CategoryId": 9,
    "CheckOutDate": null,
    "CheckinDate": null,
    "CiProfile": null,
    "CiProfileId": null,
    "CiTotal": 0,
    "City": "Panama Prueba crear 6",
    "CityId": null,
    "Company": "Disc Guadalajara",
    "CompanyId": null,
    "CostCenter": null,
    "CostCenterId": "5495",
    "Country": "Mexico",
    "CountryId": null,
    "CurrentStatus": "Estado Prueba 2",
    "CurrentStatusId": 718,
    "Department": "Development",
    "DepartmentId": null,
    "DepartmentNumber": null,
    "DepartmentNumberId": null,
    "DepreciationMethod": null,
    "Description": "Prueba catalogo crear centro costo id",
    "Floor": null,
    "FloorId": null,
    "ForegroundCurrentStatus": "0.0.0",
    "Foreign": false,
    "GroupId": -1,
    "Guld": "Guid_154_iyuhd_03052016_1652",
    "Id": 5238,
    "ImageFileName": null,
    "Impact": null,
    "LicenseNumber": null,
    "Manufacturer": "Prueba Manufacturer",
    "ManufacturerId": 929,
    "MapFilename": null,
    "Model": "GXP1200",
    "ModelId": "12",
    "Name": "AddCi_03052016_1652",
    "Price": null,
    "ProfileId": 0,
    "ProgrammedDate": null,
    "ProgrammedStatus": "Estado Programado Prueba",
    "ProgrammedStatusId": 717,
    "ProjectId": null,
    "Provider": "Company Prueba 2",
    "ProviderId": 929,
    "RFID": null,
    "ReponsibleDate": null,
    "ResidualValue": null,
    "ResponsibleId": null,
    "Serial": null,
    "Site": null,
    "SiteId": null,
    "Size": null,
    "StateId": 718,
    "Type": null,
    "Unit": null,
    "UnitId": 1,
    "UsefulLife": null,
    "UserIdModifier": null
}
```

Error messages:

Code	HTTP status	Error Message
400	BadRequest	InvalidCild
404	NotFound	NotFoundCild
500	InternalServerError	FailureGetDetailCi

CI Edition

Details of the request:

- URI: `api/v8.6/ci/{id}?isGuld={isGuld}`
- Type: PUT
- Required headings:

- Content-Type: application/json
- Authorization: TOKEN

The URI structure of the request is implemented as follows:

- If it’s by CI: api/v8.6/ci/IdCI?isGuld=False
- If by GuID: api/v8.6/ci/GuID?isGuld=True

Parameters

Code	Data type	Obligatory	Description
Id	Text	Yes	IC ID or Guld to edit
IsGuld	Boolean	Yes	In case the Id that is sent is Guld the value will be “true”, if it corresponds to the Id of a CI then it will be “false”

Additionally, it is possible to send the values you want to edit as parameters in the body of the request. These values correspond to the same values available in the creation process.

Body of the petition:

Example:

```
{
  "Description": "AAABBBCCDDDEEE",
  "AdditionalField": [
    {
      "Id": 486,
      "Type": 1,
      "Value": "Windows 15"
    },
  ],
}
```

Answer:

Value booleano indicating the result of the operation.

Error messages:

Code	HTTP status	Error Message
400	BadRequest	InvalidToken
400	BadRequest	InvalidUserId
400	BadRequest	InvalidCiData
400	BadRequest	InvalidCiId
400	BadRequest	InvalidCategoryId
400	BadRequest	InvalidProjectIdByDefault
400	BadRequest	InvalidSubcategoryId
500	InternalServerError	FailureParseCi
500	InternalServerError	FailureUpdateCi

Clear CI

Details of the request:

- URI: api/v8.6/ci/{id}?isGuld={isGuld}
- Type: DELETE
- Required headings:
- Content-Type: application/json
- Authorization: TOKEN

The URI structure of the request is implemented as follows:

- If it’s by CI: api/v8.6/ci/IdCI?isGuld=False
- If by GuID: api/v8.6/ci/GuID?isGuld=True

Parameters:

Name	Data type	Obligatory	Description
Id	Text	Yes	CI ID or Guld to be deleted
IsGuld	Boolean	Yes	In case the Id that is sent is Guld the value will be “true”, if it corresponds to the Id of a CI then it will be “false”

Body of the petition

Empty

Answer:

Value booleano indicating the result of the operation.

Error messages:

Code	HTTP status	Error Message
400	BadRequest	InvalidToken
404	BadRequest	InvalidUserId
404	BadRequest	InvalidCild
500	InternalServerError	FailureDeleteCi

Reactivate CI

Details of the request:

- URI: api/v8.6/ci/{ci}/reactivate?isGuld={isGuld}
- Type: PUT
- Required headings:
- Content-Type: application/json
- Authorization: TOKEN

The URI structure of the request is implemented as follows:

- If per CI: api/v8.6/ci/IdCI/reactivate?isGuld=False
- If by Guld: api/v8.6/ci/Guld/reactivate?isGuld=True

Parameters:

Name	Data type	Obligatory	Description
Id	Text	Yes	CI ID or Guld to be reactivated.
IsGuld	Boolean	Yes	In case the Id that is sent is Guld the value will be “true”, if it corresponds to the Id of a CI then it will be “false”

Body of the petition

Empty

Answer:

Value booleano indicating the result of the operation.

Error messages:

Code	HTTP status	Error Message
400	BadRequest	InvalidToken
404	BadRequest	InvalidUserId
404	BadRequest	InvalidCild
500	InternalServerError	FailureReactivateCi

Handling Additional Fields

Additional Fields

See additional fields

Details of the request:

- URI: api/v8.6/additionalfield
- Type: POST
- Required headings:
- Content-Type: application/json
- Authorization: TOKEN

Parameters:

Name	Data type	Obligatory	Description
ProjectId	Number	No	Id of the Ci project.
Cild	Text	No	ID of the CI, if application is foreign it receives its foreign id.
CategoryId	Number	No	Id of the Ci category.
TypeId	Number	No	Type of Ci.

Body of the petition:

Example:

```
{
  "TypeId": "14"
}
```

Answer:

```
[
  {
    "Alias": "ADD_STR_1",
    "Caption": "Prueba característica en 80",
    "Id": 92,
    "IdLookField": 0,
    "Mandatory": false,
    "Order": 0,
    "Type": 1,
    "Value": null,
    "ValueLookField": null,
    "Visible": false
  },
  {
    "Alias": "ADD_STR_2",
    "Caption": "Home",
    "Id": 213,
    "IdLookField": 0,
    "Mandatory": false,
    "Order": 0,
    "Type": 1,
    "Value": null,
    "ValueLookField": null,
    "Visible": false
  },
  ...
]
```

Error messages:

Code	HTTP status	Error Message
400	BadRequest	InvalidToken
404	BadRequest	InvalidUserId
400	BadRequest	InvalidCild
400	BadRequest	InvalidProjectIdByDefault
404	NotFound	NotFoundCild
404	NotFound	NotFoundTypeld
500	InternalServerError	FailureGetListAdditionalFields

Editing Additional Fields

Details of the request:

- URI: `api/v8.6/additionalfield/{cild}?isGuld={isGuld}`
- Type: PUT
- Required headings:
- Content-Type: `application/json`
- Authorization: TOKEN

The URI structure of the request is implemented as follows:

- If it's for CI: `api/v8.6/additionalfield/IdCI?isGuld=False`
- If by Guld: `api/v8.6/additionalfield/GulD?isGuld=True`

Parameters:

Name	Data type	Obligatory	Description
Id	Text	Yes	ID of the CI or Guld to be edited
IsGuld	Boolean	yes	In case the Id that is sent is Guld the value will be “true”, if it corresponds to the Id of a CI then it will be “false”

Note: In the body of the request, the Id that corresponds to the additional field to be edited is sent, this can be obtained by consulting additional fields described in the previous section.

Body of the petition:

Example:

```
[
  {
    "Id": 494,
    "Value": "501"
  },
  ...
]
```

Answer:

There is no response body, the validation of the result of the operation can be performed by analyzing the http code obtained: 200 OK for a successful operation.

Error messages:

Code	HTTP status	Error Message
400	BadRequest	InvalidToken
404	BadRequest	InvalidUserId
400	BadRequest	NotFoundCild
400	BadRequest	InvalidCild
400	NotFound	InvalidCategoryId
400	NotFound	InvalidDateValue
400	NotFound	InvalidType
400	NotFound	InvalidDataElement
400	NotFound	InvalidBooleanValue
400	NotFound	InvalidNumericValue
500	InternalServerError	FailureUpdateAditionalFields

Scenarios

Basic Scenarios

The following describes the steps required to perform some of the most common operations.

Sign in

1. Call login service by sending the required parameters.
2. Get the response from the service.
3. Store data such as user ID and authorization token as they will be required for subsequent services.

Renew Session

1. Call the session renewal service by sending the authorization token that is stored.
2. If the service’s response is successful, the token is still valid.
3. If the service’s response is not successful, it is necessary to obtain a new token according to what is indicated in the login scenario.

Create an IC

1. Obtain an authentication token as indicated in the login or renew session scenarios, as appropriate.
2. Consume the creation service by sending the authentication header and request body with the data corresponding to the case as appropriate.
3. To manage, as appropriate, the data obtained as a result of the consumption of the service.

Get CI Detail

1. Obtain an authentication token as indicated in the login or renew session scenarios, as appropriate.
2. Consume service of obtaining CI detail by sending the data corresponding to the case to be consulted.
3. To manage, as appropriate, the response obtained from consuming the service.

Updating an IC

1. Obtain an authentication token as indicated in the login or renew session scenarios, as appropriate.
2. Consume the CI editing service by sending authentication header and request body with the data to be edited.
3. To manage, as appropriate, the data obtained as a result of the consumption of the service.

Additional Fields

1. Obtain an authentication token as indicated in the login or renew session scenarios, as appropriate.
2. Consume the CI editing service by sending authentication header and request body with the data to be edited.
3. The additional fields both in creation and in editing mandatorily require the Id of the additional field, type of field and the information that is going to be entered in it, these values can be known by consulting additional fields.
4. To send information to an additional field from the creation or editing of a CI, this field must exist for the type (category) of CI to be managed, otherwise the action cannot be executed successfully.
5. The creation of additional fields is done only from the Aranda CMDB administration console web and Windows version.