|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Intern IDM Java-task  Guide | | A close up of a sign  Description automatically generated | | |
|  |  |
|  |  |
| V.3 | |
| TESTED ON   * SYBASE ASE 16   Last Updated on   * 29.07.2022 | | |  | User Guide The document outlines the user experience delivered by the product, which is used to create and manage employee access and HR data. |

Contents

[Changes 2](#_Toc110001339)

[1.Overview 2](#_Toc110001340)

[1.1 What Are Roles And Privileges 3](#_Toc110001341)

[1.2 HTTP Requests 3](#_Toc110001342)

[2 Core Information 4](#_Toc110001343)

[3.Functionality Overview 6](#_Toc110001344)

[3.1 Show 6](#_Toc110001345)

[3.2 Add 9](#_Toc110001346)

[3.4 Modify 12](#_Toc110001347)

[3.5 Delete 14](#_Toc110001348)

[3.6 Manage Access 15](#_Toc110001349)

# Changes



# 1.Overview

This IDM project is a REST based product which is used to simplify the organization of employee identities and managing roles and privileges. With the current release the following functionalities are available:

* Creation and management of employees
* Creation and management of roles and privileges
* Assigning, managing and removing access from the employees

The product can be accessed using the following URL:

[*http://nwj01:50000/nv\_onboarding/api/v1/rest*](http://nwj01:50000/nv_onboarding/api/v1/rest/)

## 1.1 What Are Roles And Privileges

* A role can store other privileges in it providing a lot of access at one time, it can also store other roles in it
* A privilege is a certain type of access providing access only for certain things, you can’t assign other roles or privileges to a privilege
* An ONLY privilege is a special type of privilege which provides access for certain types of systems. You can’t have access to other privileges from the same system if you don’t have the respective ONLY privilege

## 1.2 HTTP Requests

HTTP requests are used throughout all the application. Everything is access with an HTTP request.

**Here is a list of all the HTTP requests used:**

* GET - The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.
* PUT - The PUT method replaces all current representations of the target resource with the request payload.
* POST - The POST method submits an entity to the specified resource, often causing a change in state or side effects on the server.

# 2 Core Information

**The first time the system is started its very important to first run the initalLoad request. That way the database is populated with the data from the HR system.**

All commands are sent as REST requests with Json body. The response will again be in JSON

**Example of a user body with all his attributes:**

Text

Description automatically generated with low confidence

name- Indicates the user’s name

lastName - Indicates the user’s last name

title- Indicates the user’s title

dateOfBirth - Indicates the user’s date of birth where the correct format is **dd/mm/yyyy**

department - Indicates the user’s department

phoneNumber - Indicates the user’s phone number where the correct format is **000-0000-00**

managerId - Indicates the user’s manager id where the managerId can’t be the users own id

language - Indicates the user’s language

uniqueId - Indicates the user’s unique identifier where this field is automatically generated by the system in this format **IDM000N**

email - Indicates the user’s email address where this field is automatically generated by the system in this format <IDM username>@ROIABLE.com

gender - Indicates the user’s gender

position - Indicates the user’s position

hireDate- Indicates the user’s hire date where the correct format is **dd/mm/yyyy**

isDeleted - Indicates if the user is deleted. There are two options:

true – the user is deleted

false – the user is not deleted

isManager - Indicates if the user is manager. There are two options:

true – the user is a manager

false – the user is not a manager

assignments – Shows all the access that is connected to the user, but only the one that is not deleted

**Example of a role body with all his attributes:**

Graphical user interface, text, application

Description automatically generated

name – Indicates the display name (user friendly name) of the role

description – Indicates what is the role used for

isDeleted – Indicates if the role is deleted. There are two options:

true – the role is deleted

false – the role is not deleted

uniqueName - Indicates the role’s uniqueName. That field is also autogenerated using this format: **ROLE:<NAME>**

assignedTo - Shows a list of all the objects that this role is assigned to. Can be users or other roles.

asignments – Shows all the access the role has. Can be users or other roles.

**Example of a privilege body with all its attributes:**

Graphical user interface, text, application, Word

Description automatically generated

name – Indicates the display name (user friendly name) of the privilege

description – Indicates what is the privilege used for

isDeleted – Indicates if the privilege is deleted. There are two options:

true – the role is deleted

false – the role is not deleted

uniqueName - Indicates the privilege’s uniqueName. That field is also autogenerated using either of those formats: **PRIV:<SYSTEM>:<NAME> / PRIV:<SYSTEM>:<NAME>:ONLY**

assignedTo - Shows a list of all the objects that this role is assigned to. Can be users or other roles.

# 3.Functionality Overview

## 3.1 Show

Display will provide all the information assigned to the certain object and show it.

Show User

In order to get information for existing user you need to get the unique id and create a GET request on this url: [http://nwj01:50000/nv\_onboarding/api/v1/rest/showUser/{uniqueId}](http://nwj01:50000/nv_onboarding/api/v1/rest/showUser/%7buniqueId%7d)

You can also see all the access a user has in the “assignments” field when displaying him.

**Example request and response for user IDM0003:**

A picture containing graphical user interface

Description automatically generated

Here in the assignments field only the access that is not deleted is shown.

**In order to get information for every existing user in the system you need to create a GET request on this url:**

<http://nwj01:50000/nv_onboarding/api/v1/rest/showUsers>

Graphical user interface, text, application, website

Description automatically generated

Show Role

In order to get information for and existing role you need to get the unique name and create a GET request on this url: [http://nwj01:50000/nv\_onboarding/api/v1/rest/showAssignemntObject/{uniqueName}](http://nwj01:50000/nv_onboarding/api/v1/rest/showAssignemntObject/%7buniqueName%7d)

You can also see all the access a role has in the “assignments” field when displaying him.

**Example request and response for role ADMIN:**

Graphical user interface, text, application, email

Description automatically generated

In order to get information for every existing role in the system you need to create a GET request on this url:

http://nwj01:50000/nv\_onboarding/api/v1/rest/showRoles

Show Privilege

In order to get information for and existing privilege you need to get the unique name and create a GET request on this url: [http://nwj01:50000/nv\_onboarding/api/v1/rest/showAssignemntObject/{uniqueName}](http://nwj01:50000/nv_onboarding/api/v1/rest/showAssignemntObject/%7buniqueName%7d)

You can also see all the access a privilege has in the “assignments” field when displaying him

**Example request and response for privilege PRIV:Sys1:ONLY**

Graphical user interface, text, application, email

Description automatically generated

In order to get information for every existing user in the system you need to create a GET request on this url:

http://nwj01:50000/nv\_onboarding/api/v1/rest/showPrivileges

## 3.2 Add

Add User

To add a user, you need to create a POST request on this url: <http://nwj01:50000/nv_onboarding/api/v1/rest/addUser>

Here are all the attributes a user can have:

Table

Description automatically generated

The fields that are shown as mandatory always must be added otherwise the user will be invalid, while the other fields marked with no are optional.

**Example of an addUser request:**

Text

Description automatically generated

managerId - Indicates the user’s manager id where the managerId can’t be the user’s own id and it must be a reference to an existing manager in the system.

isManager - Indicates if the user is manager. There are two options:

true – the user is a manager

false – the user is not a manager

**First user in the system:**

When we add a user for the first time the system will automatically assign his managerId to be his own since there are no other users in the system. That could be changed later

Add Privilege

To add a privilege, you need to create a POST request on this url:

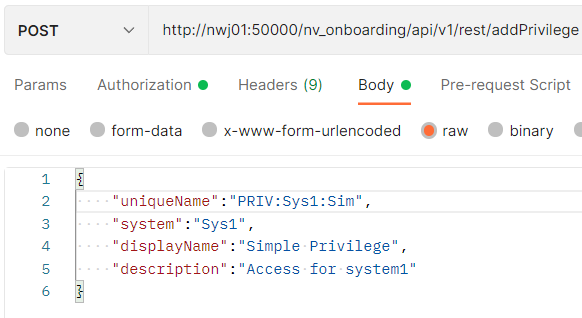
<http://nwj01:50000/nv_onboarding/api/v1/rest/addPrivilege>

Here are all the attributes a privilege can have:

Table

Description automatically generated

**Example of an addPrivilege request:**



**ONLY privilege:**

For each system there should be one special privilege: PRIV:<SYSTEM>:ONLY. A user should have this privilege in order to have access to other roles and privileges from the respective system. No other role or privilege from that system can be assigned if the user doesn’t have the ONLY privilege for that system.

This is an example for creating an ONLY privilege:

Graphical user interface, text, application

Description automatically generated

Add Role

To add a role, you need to create a POST request on this url:

<http://nwj01:50000/nv_onboarding/api/v1/rest/addRole>

Here are all the attributes a privilege can have:

Table

Description automatically generated

**Example of an addRole request:**

Graphical user interface, text, application, email

Description automatically generated

## 3.4 Modify

Modify User

To modify a user, you need to create a PUT request on this URL:

[http://nwj01:50000/nv\_onboarding/api/v1/rest/modifyUser/<uniqueId](http://nwj01:50000/nv_onboarding/api/v1/rest/modifyUser/%3cuniqueId)>

In the field<uniqueId> in the URL you put the unique Id of the user you want to modify,

Where you can’t modify a deleted user or a non-existing one.

**Example of a modifyUser request:**

**Graphical user interface, text

Description automatically generated**

Only these fields shown here are modifiable. None of them can be null.

dateOfBirth - Indicates the user’s date of birth where the correct format is **dd/mm/yyyy**

phoneNumber - Indicates the user’s phone number where the correct format is **000-0000-00**

managerId - Indicates the user’s manager id where the managerId can’t be the users own id

Modify Role

To modify a role, you need to create a PUT request on this URL:

[http://nwj01:50000/nv\_onboarding/api/v1/rest/modifyRole/<uniqueName](http://nwj01:50000/nv_onboarding/api/v1/rest/modifyRole/%3cuniqueName)>

In the field<uniqueName> in the URL you put the unique Id of the roleyou want to modify,

Where you can’t modify a deleted role or a non-existing one.

**Example of a modifyRole request:**

**Graphical user interface, text, application, Word

Description automatically generated**

Only these fields shown here are modifiable. None of them can be null.

displayName – this is a user-friendly name for the role

description – this fields describes what’s the purpose of the role

Modify Privilege

To modify a privilege, you need to create a PUT request on this URL:

[http://nwj01:50000/nv\_onboarding/api/v1/rest/modifyPrivilege/<uniqueName](http://nwj01:50000/nv_onboarding/api/v1/rest/modifyPrivilege/%3cuniqueName)>

In the field<uniqueName> in the URL you put the unique Id of the privilege you want to modify,

Where you can’t modify a deleted privilege or a non-existing one.

**Example of a modifyPrivilege request:**

**Graphical user interface, text, application, Word

Description automatically generated**

Only these fields shown here are modifiable. None of them can be null.

displayName – this is a user-friendly name for the privilege

description – this fields describes what’s the purpose of the privilege

## 3.5 Delete

Delete User

To delete a user, you need to create a PUT request on this URL:

[http://nwj01:50000/nv\_onboarding/api/v1/rest/deleteUser/<uniqueId](http://nwj01:50000/nv_onboarding/api/v1/rest/deleteUser/%3cuniqueId)>

In the field<uniqueId> in the URL you put the unique Id of the user you want to delete.

Where you can’t delete an already deleted user or a non-existing one.

When you delete a user, you will get this message:



When you delete an already deleted user, you will get this message:



When you delete a non-existing user, you will get status code 404 NOT FOUND and an empty body as a return

Delete Role

To delete a role, you need to create a PUT request on this URL:

[http://nwj01:50000/nv\_onboarding/api/v1/rest/deleteAssignemntObject/<uniqueName](http://nwj01:50000/nv_onboarding/api/v1/rest/deleteAssignemntObject/%3cuniqueName)>

In the field<uniqueName> in the URL you put the unique Id of the role you want to delete.

Where you can’t delete an already deleted role or a non-existing one.

When you delete a role, you will get this message:



When you delete an already deleted role, you will get this message:



When you delete a non-existing role, you will get status code 404 NOT FOUND and an empty body as a return

Delete Privilege

To delete a privilege, you need to create a PUT request on this URL:

[http://nwj01:50000/nv\_onboarding/api/v1/rest/deleteAssignemntObject/<uniqueName](http://nwj01:50000/nv_onboarding/api/v1/rest/deleteAssignemntObject/%3cuniqueName)>

In the field<uniqueName> in the URL you put the unique Id of the privilege you want to delete.

Where you can’t delete an already deleted privilege or a non-existing one.

When you delete a privilege, you will get this message:



When you delete an already deleted privilege, you will get this message:



When you delete a non-existing privilege, you will get status code 404 NOT FOUND and an empty body as a return

## 3.6 Manage Access

Assign Access

To assign access to an object you need to make a POST request on this URL:

<http://nwj01:50000/nv_onboarding/api/v1/rest/assignAccess>

**This is an example of an assignAccess body:**

**Text, letter

Description automatically generated**

uniqueId – this field can be either the unique id of a user or the unique name of a role. It can’t be a privilege because you can’t assign access to privileges

access – this field can either be the unique name of a role or a user. It can’t be a user because you can’t assign user as an access

fromDate – Indicates when the access will start. It must follow the following pattern **dd/mm/yyyy,**

fromDate can’t be bigger or equal to toDate.

toDate – Indicates when the access will end. It must follow the following pattern **dd/mm/yyyy,**

toDate can’t be smaller or equal to fromDate

**You can’t:**

1. Assign access to invalid object

2. Assign invalid access

3. Assign access to the same object

3. Assign user as an access

4. Assign ONLY privileges to a role

5. Assign access with invalid date format

6. Assign access to a deleted object

7. Assign a privilege to a user without the ONLY privilege

Modify Access

To assign access to an object you need to make a PUT request on this URL:

<http://nwj01:50000/nv_onboarding/api/v1/rest/modifyAccess>

**Example of a modifyAccess body:**

Text

Description automatically generated

Remove Access

To remove access to an object you need to make a PUT request on this URL:

[http://nwj01:50000/nv\_onboarding/api/v1/rest/removeAccess/<uniqueId>/<accessToRemove](http://nwj01:50000/nv_onboarding/api/v1/rest/removeAccess/%3cuniqueId%3e/%3caccessToRemove)>

* <uniqueId> - this field can only be a user or a role, because you can’t remove access from a privilege
* <accessToRemove> - this field can only be a role or a privilege, because you can’t remove user from access

**Example of a remove access request:**

****

****

**Example of remove access request for two objects with to relation between them:**



**Example of remove access request for an invalid object:**

****