How can we help?

Categories

- Overview
- FAQs
- Software Guides
- Hardware Guide
- Integration
  - Spintly Platform Integrations
    - Mobile Docs
    - Spintly Integration Overview
    - Prerequisite for backend api Integration
    - Prerequisite for mobile sdk Integration
    - Steps to get started
    - Type 1 Integration
    - Steps to get up and running (Type 1 integration)
    - Examples of how to use type 1 with your application
    - Type 2 integrations
    - Steps to get up and running (Type 2 integration)

Home / Integration / Spintly Platform Integrations

Type 4 integration (Whitelabeling)

Categories

Type 4 integration (Whitelabeling)

Table of Contents

Note

Make sure you have gone through the Spintly Integration Overview section, as that section will help you understanding the spintly platform and the types of integration

Examples of how to use type 2 with your

Use Case for Integration

In whitelabbeling, the client dosent have to do any development, In this integration spintly can whitelabel its app and spintly website with their logo and client theme color, the same can be logo can be put to out devices and gateways also. Also whichever integration mention before ie, type1, type2, and type3, will be available for the clients also.

There are two ways you approcahes you can go with

1. Whitelabelling the spintly websites and apps , but cloud resides in spintly
2. Whitelabelling the spintly websites and apps, but you can have the spintly platform deployed to your own aws account

To get started please contact spintly help and support team link

embedding white labeling

Was this article helpful?

Yes No

Give feedback about this article

Related Articles

Door Schedules

Organization Settings

Can I give specific door accesses to a user?

Spintly
Amplify the Simplicity

Industries

Multifamily / Residential
Co-Working
Education
Healthcare
Hospitality
Manufacturing
Fitness & Wellness

Solutions

Access Control
Visitor Management
Tenant Management
Time & Attendance Management

About

About Us
Partners
FAQ
Privacy Policy

Terms & Conditions

Resources

Case Studies

Blogs

Whitepapers

Webinars

Press

Data Sheets

Contact Us

Sales: sales@spintly.com

Tech Support: support@spintly.com

Get a Quote

Type 5 integration

Type 5 integration

Note

Make sure you have gone through the Spintly Integration Overview section, as that section will help you understanding the spintly platform and the types of integration. Please note the Type 5 integration will always go hand in hand with Type 3 integration. Also the below functionality are already available in Spintly partner website and Spintly partner app. The functionality which are available in Spintly partner website is available via apis and functionalities which are available in spintly partner app are available as a sdk to the client. Please note if you are doing this integration you will need both the app and website, this integration can be used in below combination

1. Use Spintly Partner apis to create/delete devices gateways etc and use Spintly partner app to configure the devices
2. Use Spintly Partner apis to create/delete devices gateways etc and incorporate Spintly partner sdk in your app to configure the devices

Use Case for Integration

This integration is normally used when an exisiting access control client wants to integrate spintly devices with thier platform, this integration normaly goes hand in hand with type 3 integration

Live example:

This solution in combination with type 3 integration is already live and implemented by lot of our indian customers namely godrej

The features to be offered in this integration

Features	Available

SPINTLY SAAMS API	NO
SPINTLY ACCESS SDK	NO
SPINTLY SAAMS APP	NO
SPINTLY SAAMS WEBSITE	NO
SPINTLY PARTNER WEBSITE/APP	YES
SPINTLY PARTNER SDK/API	YES

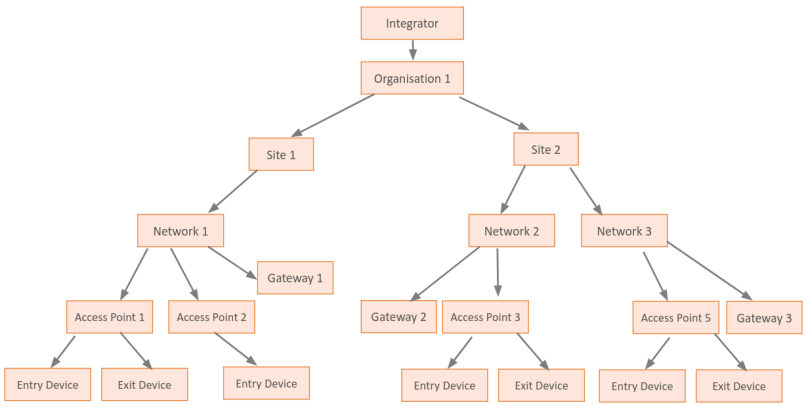
Steps to get up and running

As mentioned above Type 5 integration will always go hand in hand with Type 3, Type 5 integration cannot star

Please refer the following link to get up and running along with type 6 integration

Spintly Apis

Please keep the below diagram in mind while going through the apis

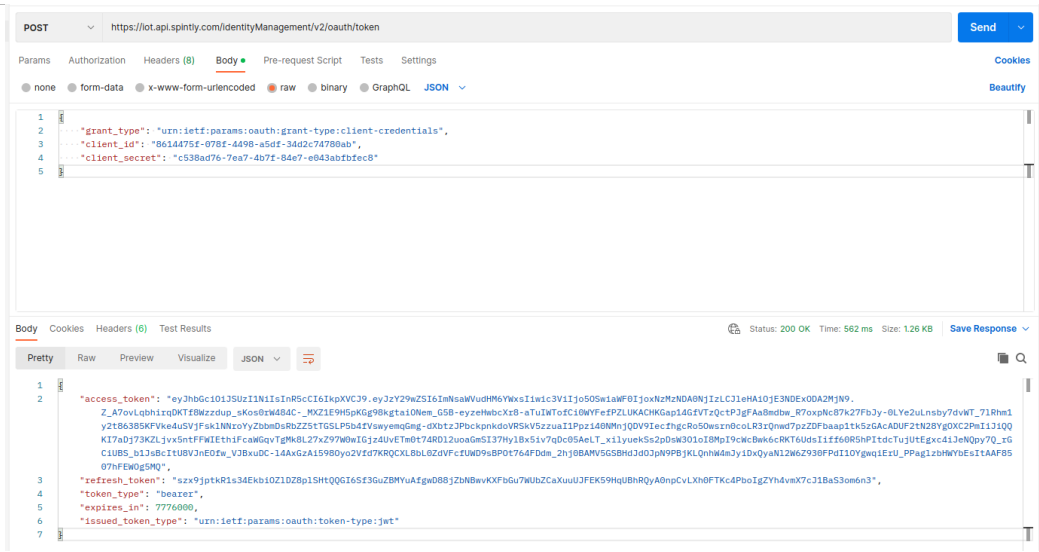


Following are the apis

1. Oauth apis
2. Create Organisations
3. Create Site
4. Delete Site
5. Create Network
6. Delete Network
7. Add Access point
8. Delete Access point
9. Add gateway

1) Ouath Api

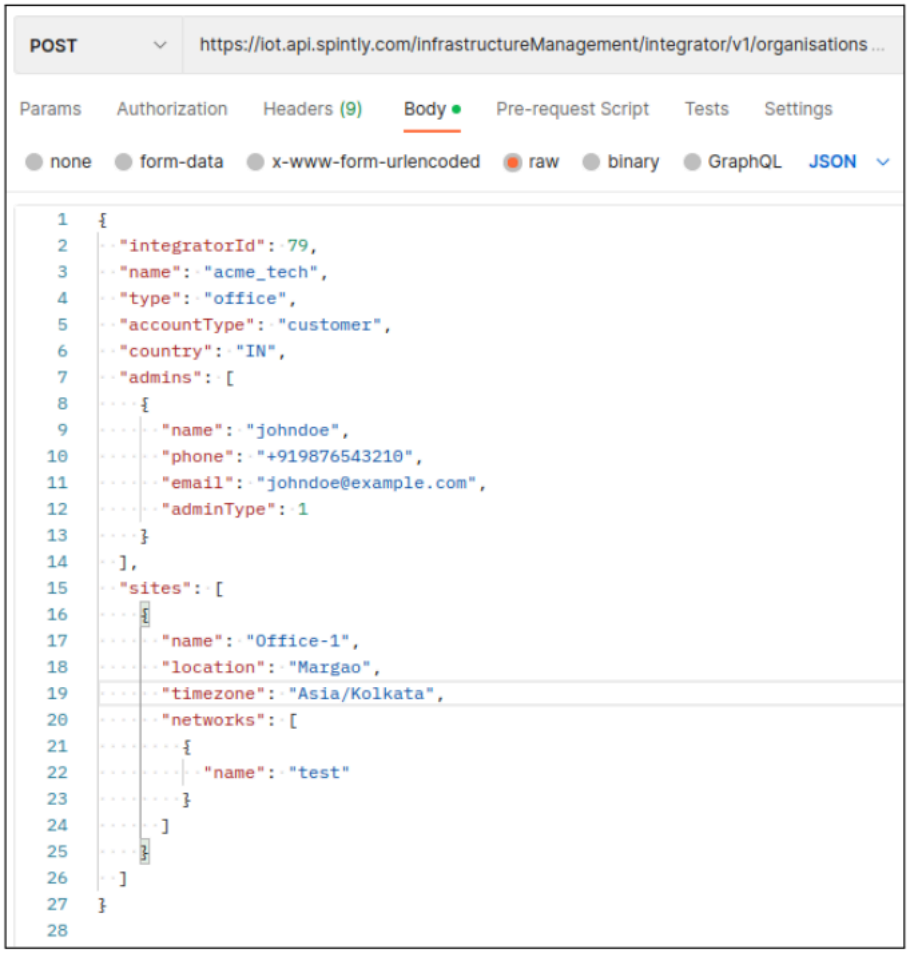
Spintly uses oauth 2.0 client credential flow for authorization purpose, here in the api the client just has to put a clientid and clientSecret to get an access token, this access token can then be used to access other apis of Spintly.



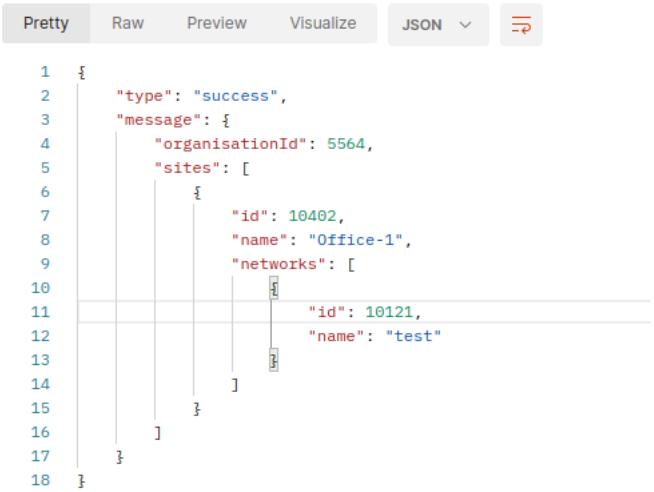
2) Create Organisation

Spintly uses oauth 2.0 client credential flow for authorization purpose, here in the api the client just has to put a clientid and clientSecret to get an access token, this access token can then be used to access other apis of Spintly. As shown in the spintly resources chart, when you create an organistion a site and a network under the site is created

REQUEST BODY

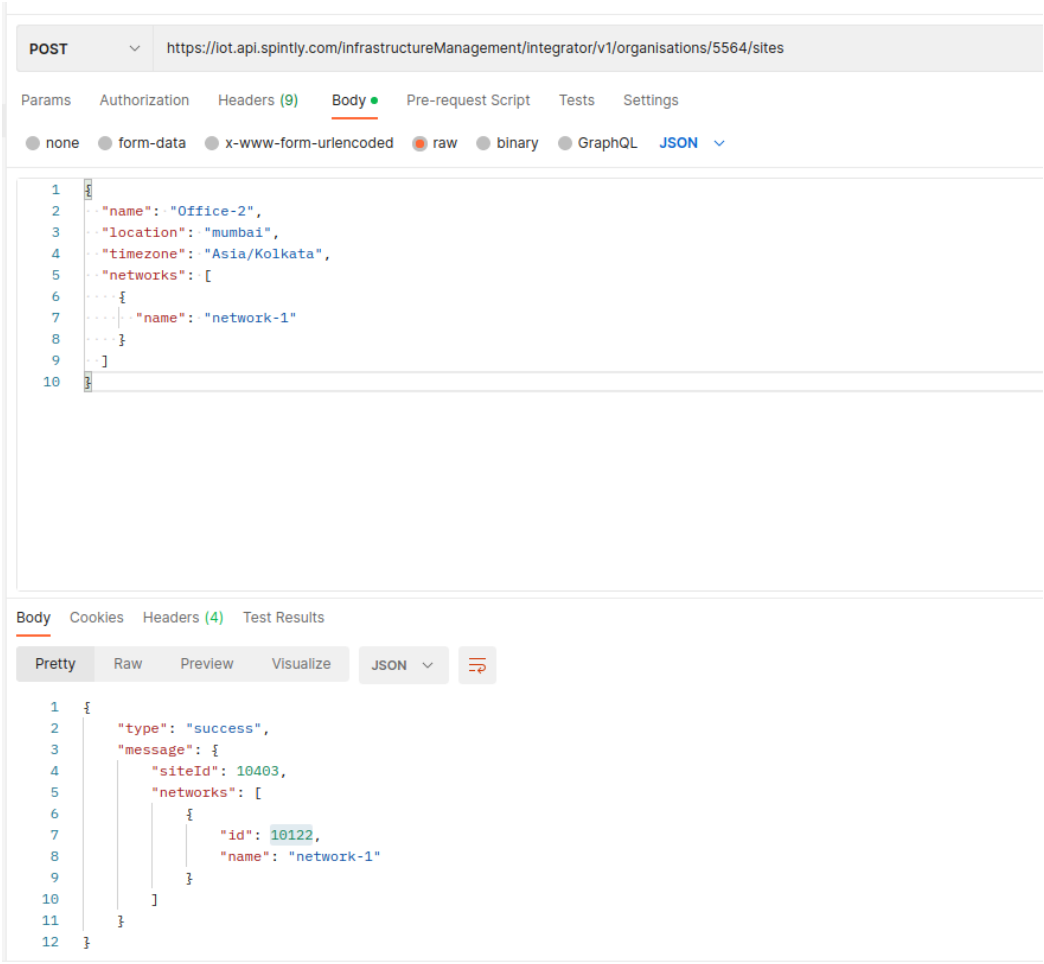


RESPONSE BODY



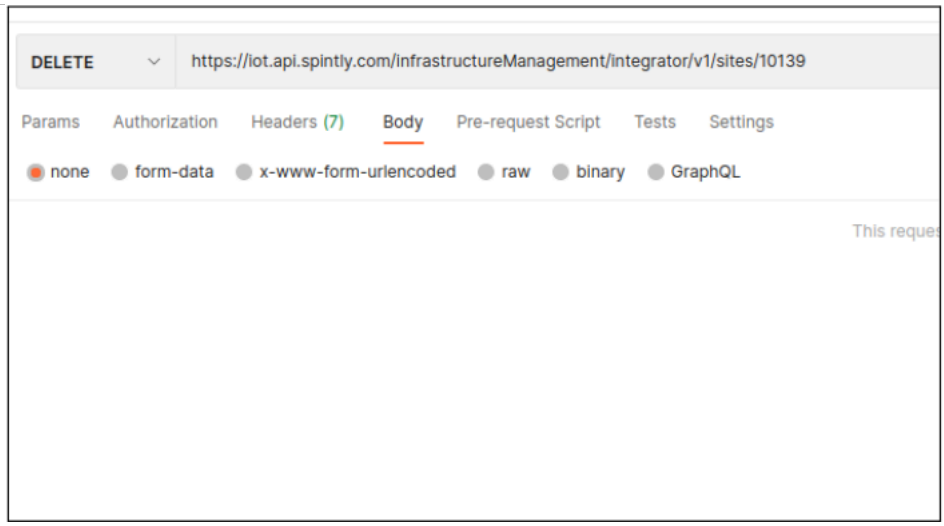
3) Create Site

This api is responsible for creating a new site under an organisations, this api needs to be called whenever client opens a new branch or center in a new location



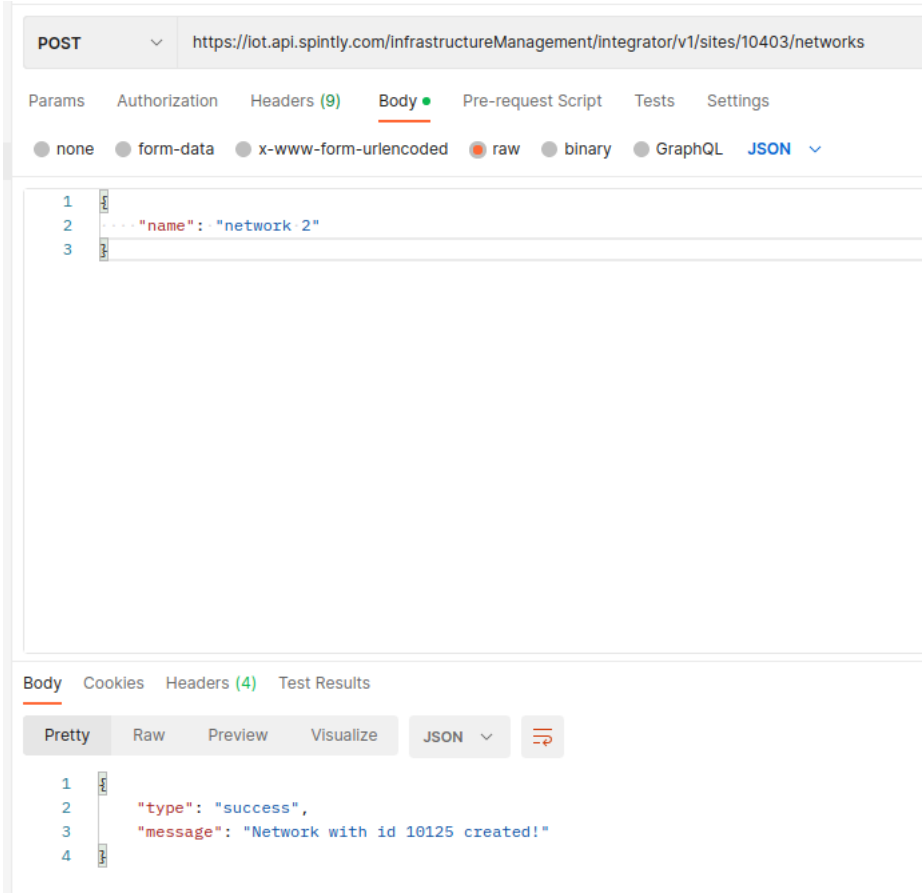
4) Delete Site

This api is responsible for creating a new site under an organisations, this api needs to be called whenever client opens a new branch or center in a new location



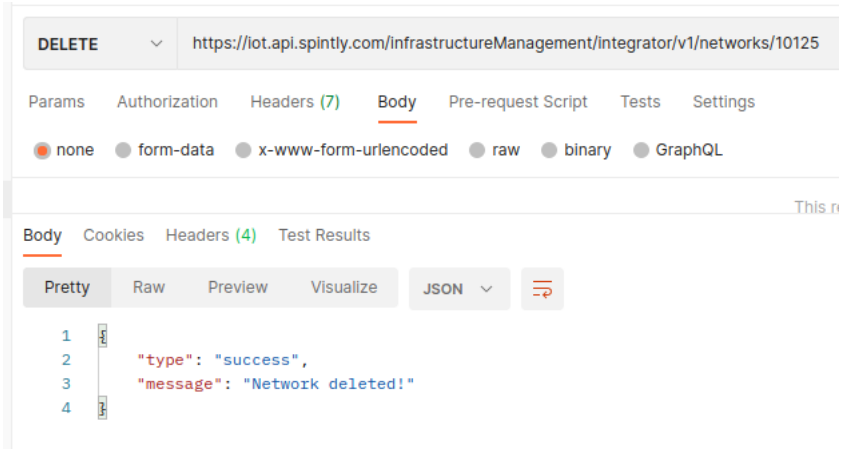
5) Create Network

This api is responsible for creating a new network under a site, say suppose there is 10 floor building, and the company has occupied two floors one on the 1st floor and one on the ninth floor, that time we create one network for 1st floor and one network we create for 9th floor



6) Delete Network

This api is responsible for deleting existing network



Add Access point

As mentioned in the “Spintly Software Resource” section, access point is nothing but a door, Spintly allows readers to be installed in different configurations:

1. Entry and Exit reader: if a door need entry as well as exit reader
2. Entry and REX: if door needs to have only an entry reader and on exit exit button(REX)
3. Clock-in device: In this configuration device is not attached to a door, it just attached to wall, this kind configuration is used for attendance purpose where employees can can check in and checkout
4. Door Lock

Type of Configuration	**Values to be put in the api**
Entry and Exit reader	1
Entry and REX	2
Clock-in device:	5
Door Lock	6

Installation method: Here there are two kinds of installation method

1. New Install: This is a brand new installation method, where it a reader needs to be attached to a new door
2. Retrofit Install: This kind of installation where in the existing reader of another 3rd party is removed and instead Spintly reader is put

Installation method	Values to be put in the api
New Install	1
Retrofit type 1	2
Retrofit type 1	3

Locking mechanism: Here Spintly has the following locking mechanism

1. Door with mag lock
2. Door with strike lock
3. Boom barrier or Sliding gate

Locking mechanism	Values to be put in the api
Door with mag lock	1
Door with strike lock	2
Boom barrier or Sliding gate	3

Device Type

1. entry
2. exit
3. sc_controller
4. mc_controller

Device Type	Values to be put in the api
Entry	1
Exit	2
SC_controller	3
MC_controller	4

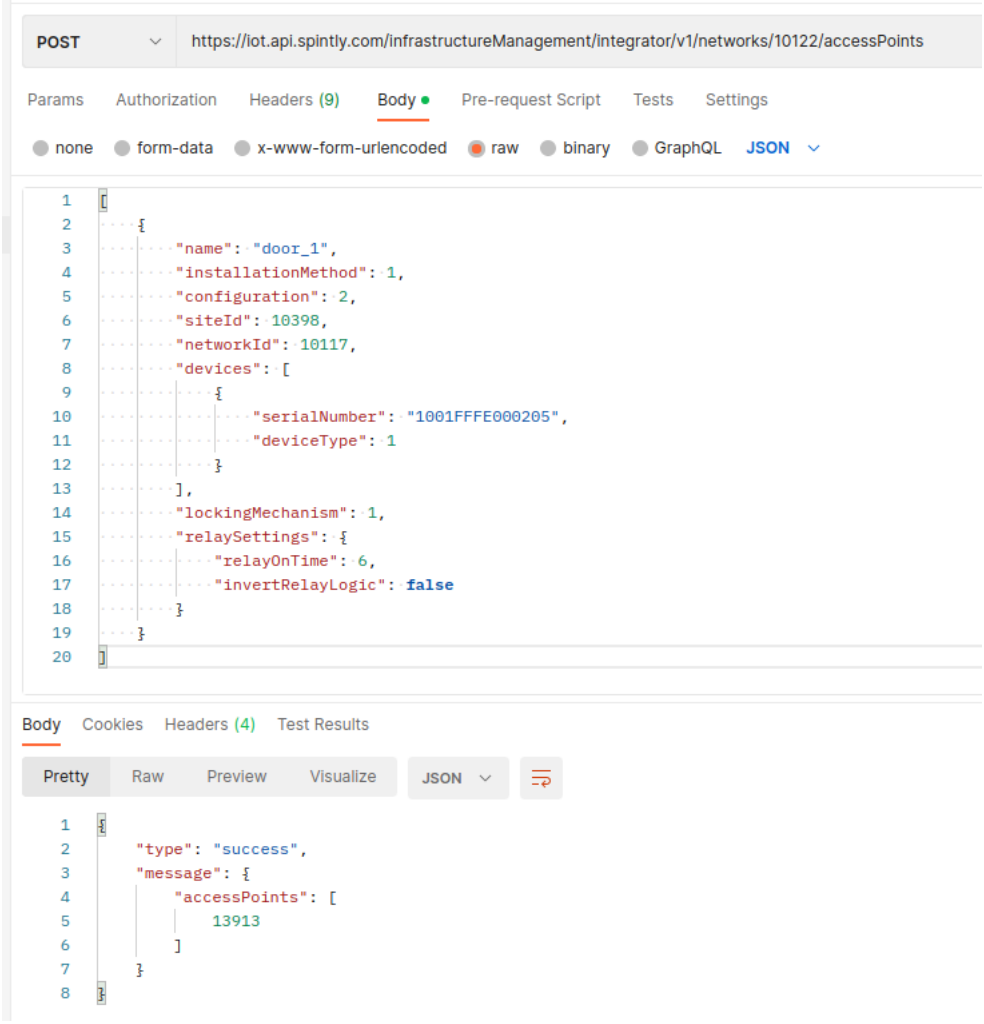
SerialNumber: All the readers have a serial number attached to it at the bottom, the user needs to enter the same when creating the access point

Please note once the access point is added, it can be in one of the following states

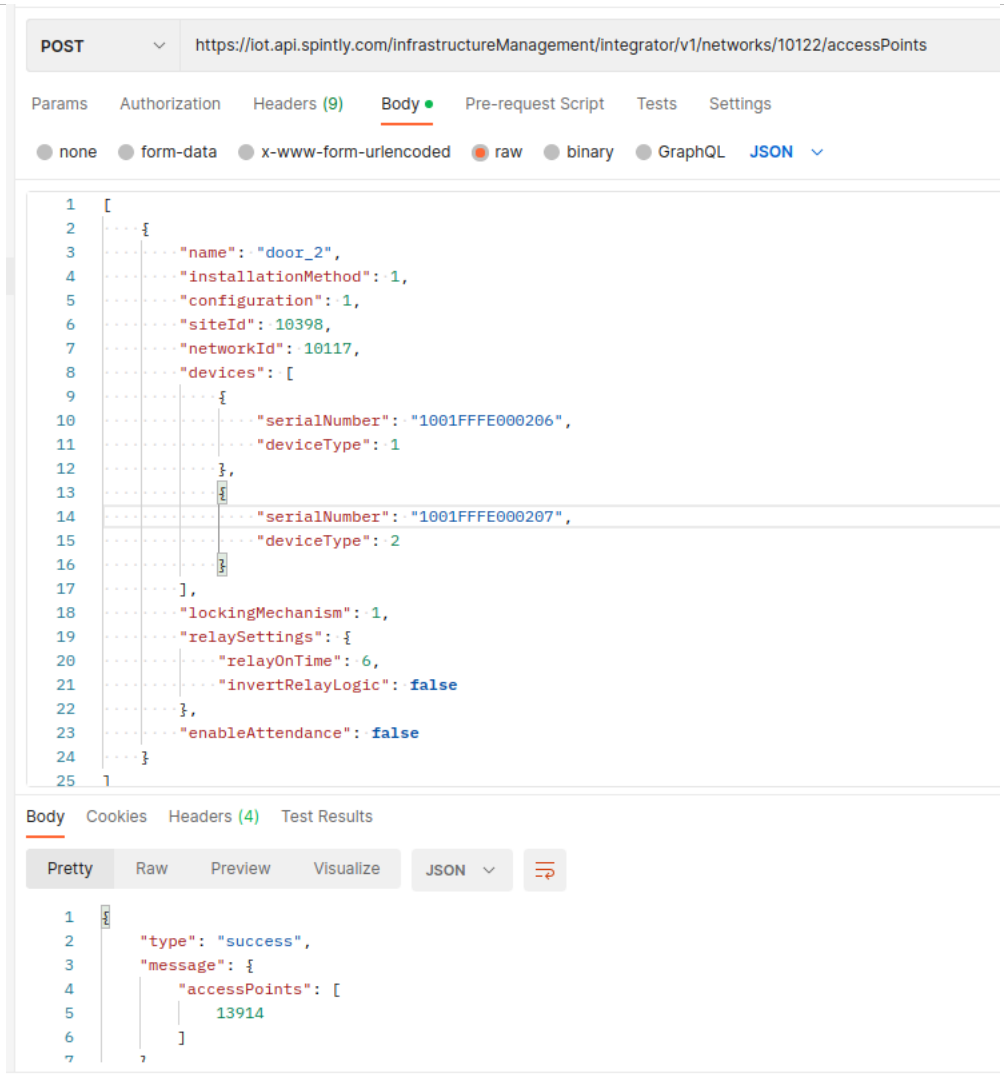
Status	Description	Can be assigned permissions to the user
create_pending	When a access point is created, the initial status will be always be “create_pending”	Cannot be used
ready_pending	This means the access point is created successfully, only that it needs to configured via spintly partner app	Cannot be used
alive_pending	Device is in process of getting configured	Cannot be used
in_sync	Access point is ready and can be used	Can be used
update_pending	Access point name is updated or some setting of access point is updated	Can be used
delete_pending	Here the Access point is in process of being deleted	Cannot be used

NOTE: Once the access points are added, the readers under the access points needs to be configured using the Spintly mobile based partner application

7a) Add Access point as entry and rex configuration



7b) Add Access point as entry and exit configuration



8) Add Controller

Adding of controller is similar to adding of access point. Spintly has 2 types of controller ie

1. Single door Controller
2. Two Door Controller

Spintly allows controller to be installed in different configurations:

1. Entry Reader + Door Controller
2. Door Controller + REX

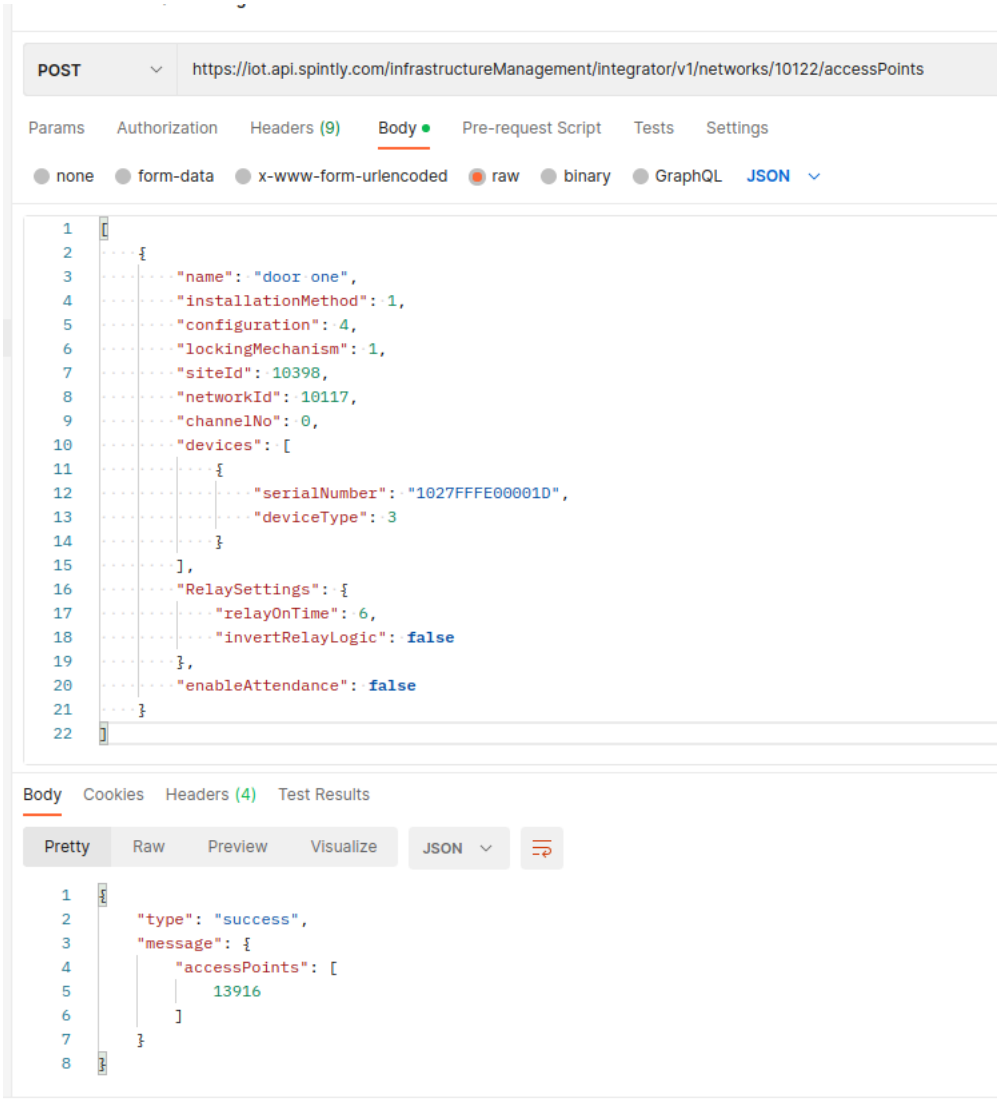
Type of Configuration	Values to be put in the api
Entry Reader + Door Controller	3
Door Controller + REX	4

Locking mechanism: Here Spintly has the following locking mechanism

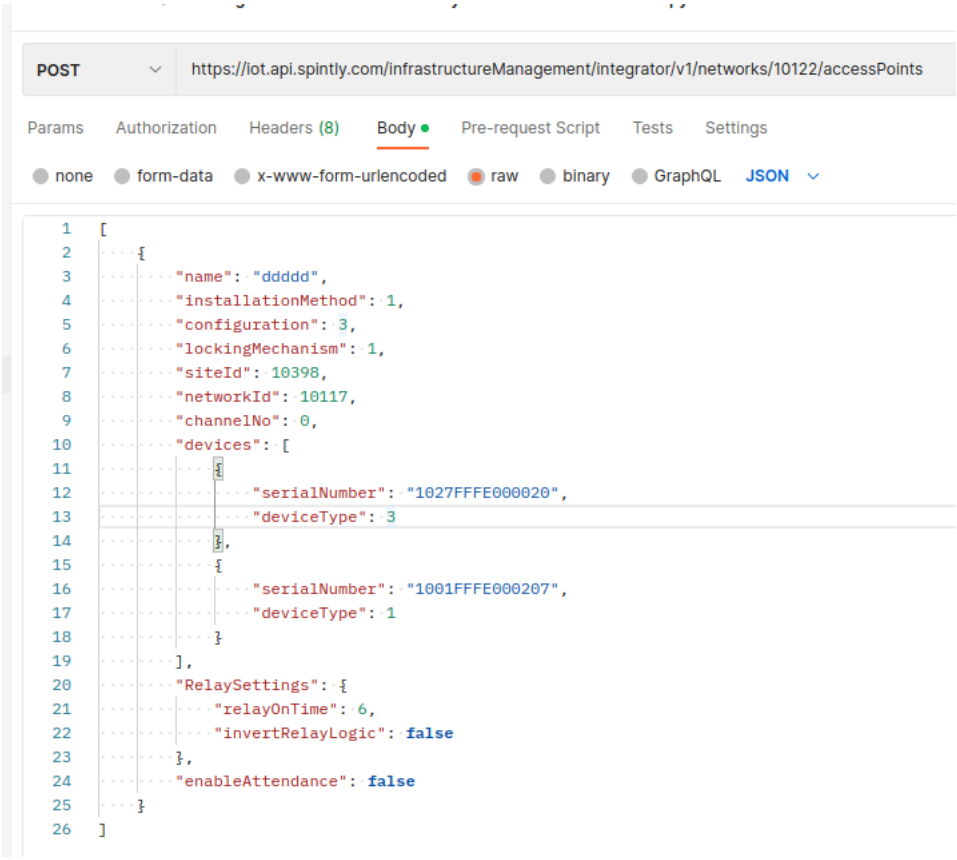
1. Door with mag lock
2. Door with strike lock

Locking mechanism	Values to be put in the api
Door with mag lock	1
Door with strike lock	2

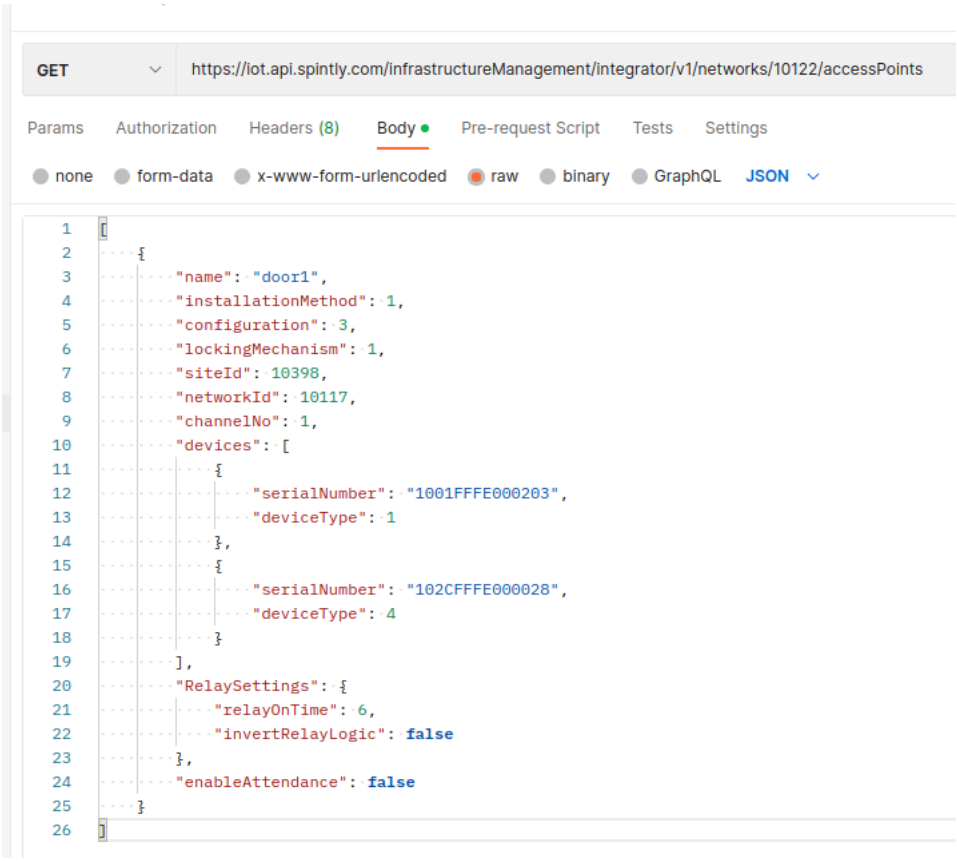
8a) Add Single Door Controller as Door Controller + REX configuration



8b) Add Single Door Controller as Entry Reader + Door Controller configuration



8c) Add Two Door Controller with one channel enabled



8d) Add Two Door Controller with two channel enabled

POST https://iot.api.spintly.com/infrastructureManagement/integrator/v1/networks/10122/accessPoints

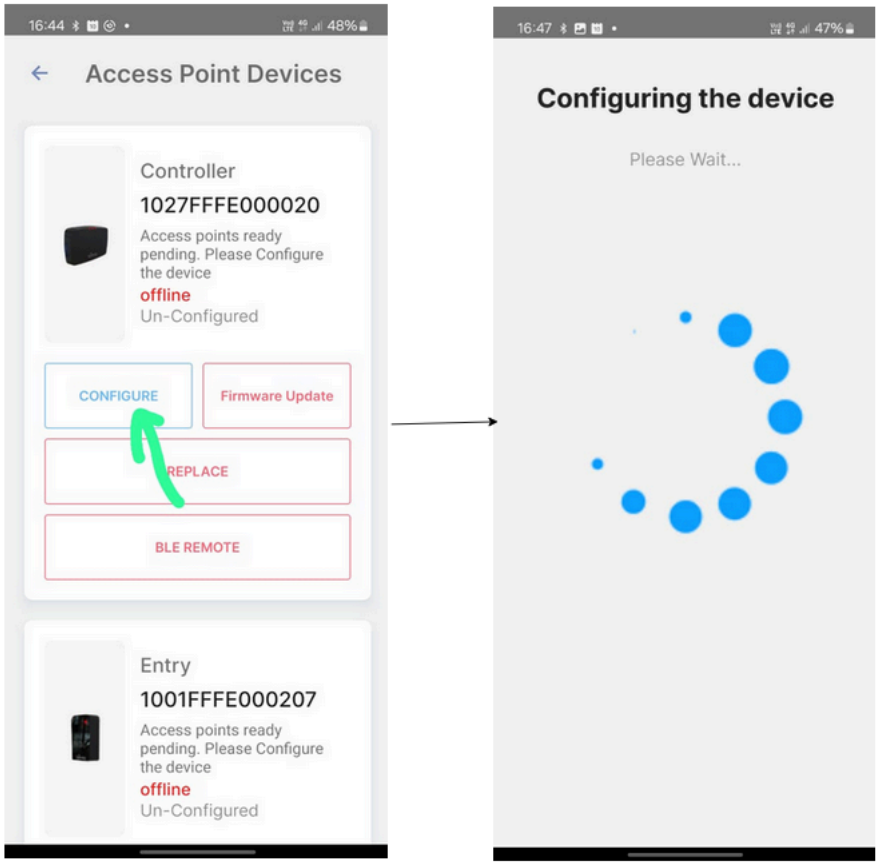
Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON


```
[
    {
        "name": "entry_reader",
        "installationMethod": 1,
        "configuration": 3,
        "lockingMechanism": 1,
        "siteId": 10398,
        "networkId": 10117,
        "channelNo": 1,
        "devices": [
            {
                "serialNumber": "1001FFFE000203",
                "deviceType": 1
            },
            {
                "serialNumber": "102CFFFE000028",
                "deviceType": 4
            }
        ],
        "RelaySettings": {
            "relayOnTime": 6,
            "invertRelayLogic": false
        },
        "enableAttendance": false
    },
    {
        "name": "exit_reader",
        "installationMethod": 1,
        "configuration": 3,
        "lockingMechanism": 1,
        "siteId": 10398,
        "networkId": 10117,
        "channelNo": 2,
        "devices": [
            {
                "serialNumber": "1001FFFE000204",
                "deviceType": 1
            },
            {
                "serialNumber": "102CFFFE000028",
                "deviceType": 4
            }
        ],
        "RelaySettings": {
            "relayOnTime": 6,
            "invertRelayLogic": false
        },
        "enableAttendance": false
    }
]
```

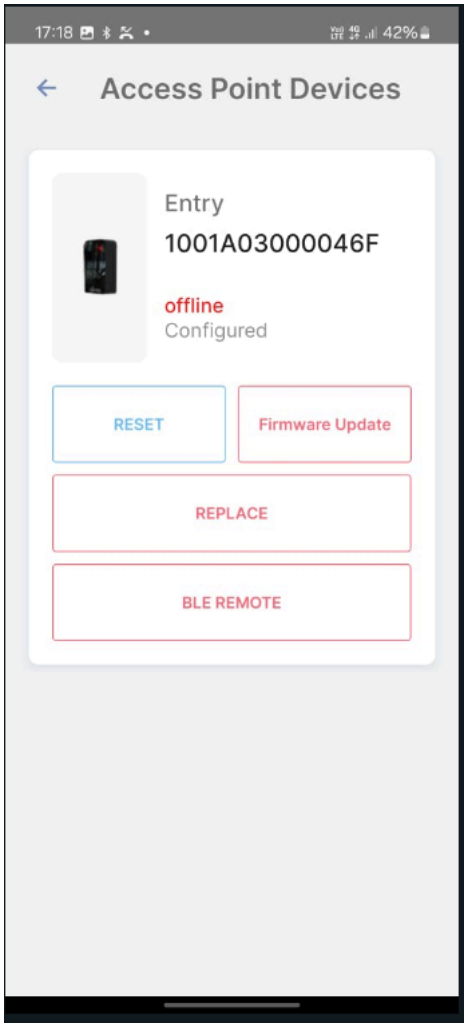

8e) Configuring the access point

Once the access point or controller is added, it needs to be configured using Spintly partner app

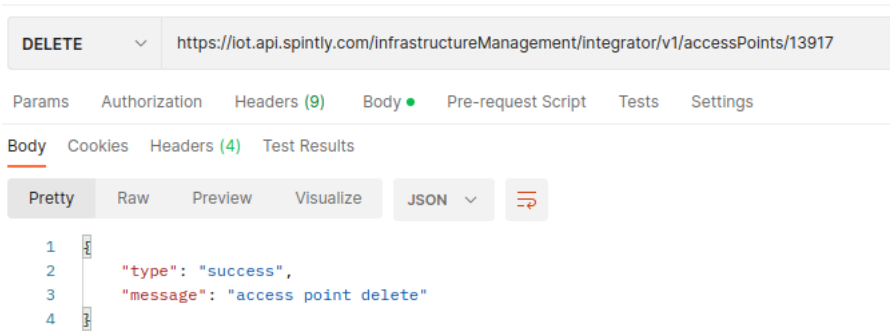


9) Delete Access point and Controller

Before deleting the access point/controller via the api we need to reset the devices under it using spintly partner app



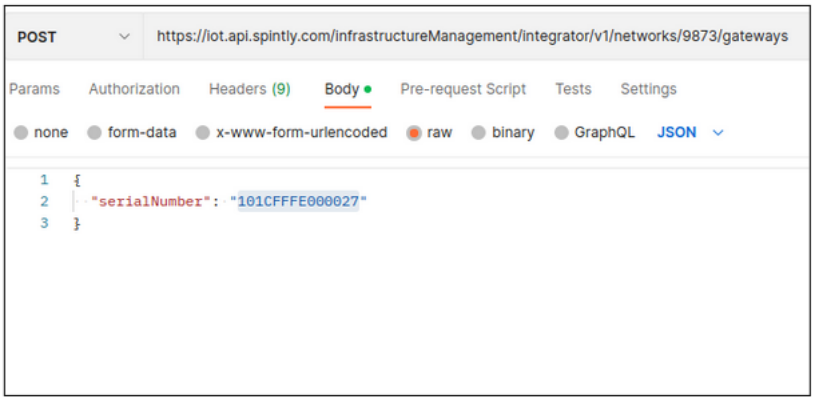
Same api will be used to delete the access points and controller, please note delete of access point wont happened if the device is not reseted using the partner app



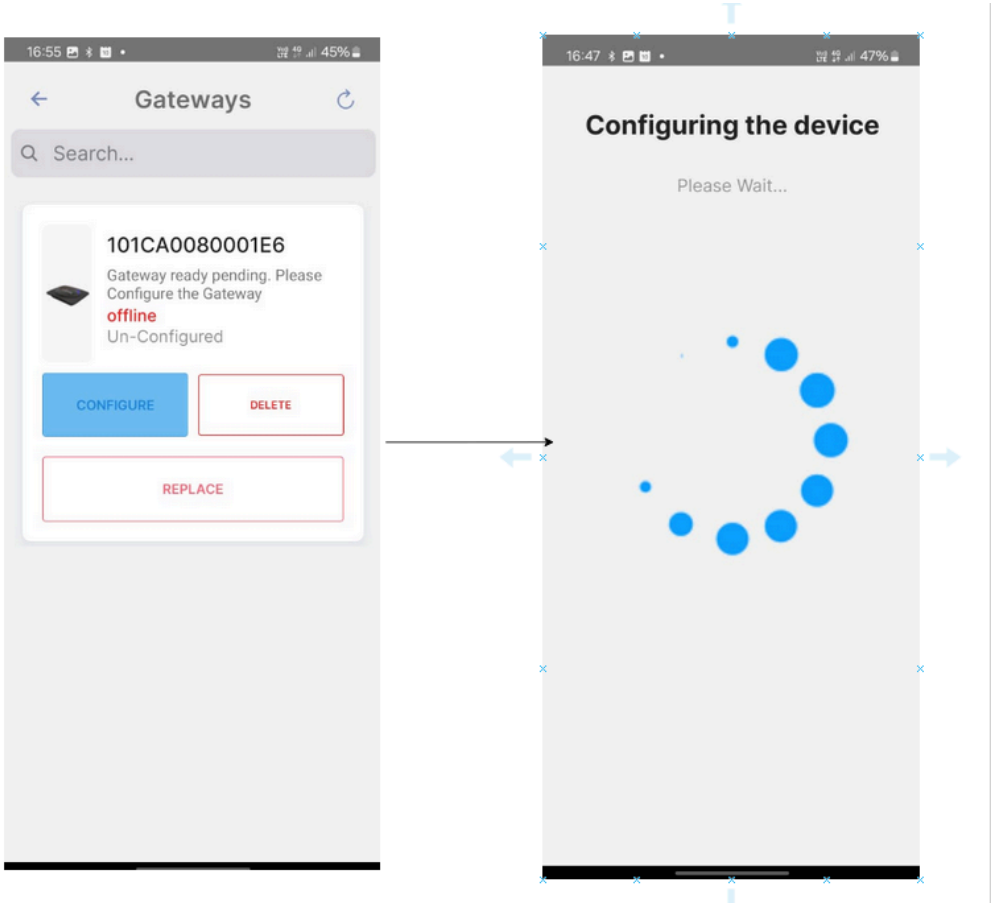
10) Add Gateway

This api is responsible for creating a new gateway, just like how a reader has a serial number will attached bottom of the gateway, just like the access point it needs to be under a network and it needs to be

configured using Spintly partner app



10e) Configure Gateway



11) Delete Gateway

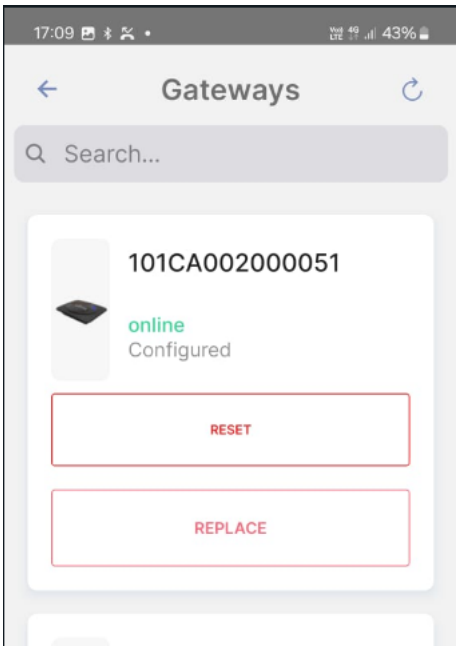
Before deleting the gateway, we need to reset the gateway from the partner app, make sure the gateway is online(ie connected to gateway)

Related Articles

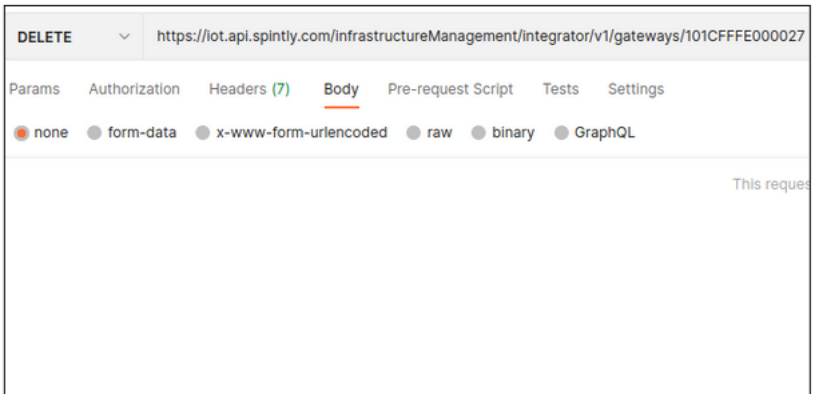
Access Points

Door Sensors

Fire Integration & Configuration



This api is responsible for deleting a gateway, just make sure the gateway is first reseted using the Spintly partner app



incorporation fusion

Was this article helpful?

Yes No

Give feedback about this article

Spintly
Amplify the Simplicity

Industries

Multifamily / Residential
Co-Working
Education

Healthcare
Hospitality
Manufacturing
Fitness & Wellness

Solutions

Access Control
Visitor Management
Tenant Management
Time & Attendance Management

About

About Us
Partners
FAQ
Privacy Policy
Terms & Conditions

Resources

Case Studies
Blogs
Whitepapers
Webinars
Press
Data Sheets

Contact Us

Sales: sales@spintly.com
Tech Support: support@spintly.com
Get a Quote

How can we help?

Categories

- Overview
- FAQs
- Software Guides
- Hardware Guide
- Integration
  - Spintly Platform Integrations
    - Mobile Docs
    - Spintly Integration Overview
    - Prerequisite for backend api Integration
    - Prerequisite for mobile sdk Integration
    - Steps to get started
    - Type 1 Integration
    - Steps to get up and running (Type 1 integration)
    - Examples of how to use type 1 with your application
    - Type 2 integrations
    - Steps to get up and running (Type 2 integration)
    - Examples of how to use type 2 with your application
    - Spintly Visitor Qr Apis integration
    - Type 3

Home / Integration / Spintly Platform Integrations

Type 3 integration

Categories

Type 3 integration

Table of Contents

Note

Make sure you have gone through the Spintly Integration Overview section, as that section will help you understanding the spintly platform and the types of integration, This integration can be combined with type 5 integration

Use Case for Integration

This integration is normally used when a client just want to use Spintly access management service, how this is different from type 2 integration is in type 2 spintly apart from access management, also gives other modules like user management, visitor management, attendance management, and all its ui capabilities, In type 3 integration, the client gets apis only of the access management section.

Live example:

This solution is already live and implemented by lot of our Indian customers namely Godrej, So here the client had thier own website and app, with this integration the client included the spintly sdk in thier app, so with this app the client could open the doors, they also integrated spintly backend with thier own software, whenever a user is added or permissions were changed, spintly apis were called to create accessor and update the permissions and the same permission would reflect in the app via the sdk

The features to be offered in this integration

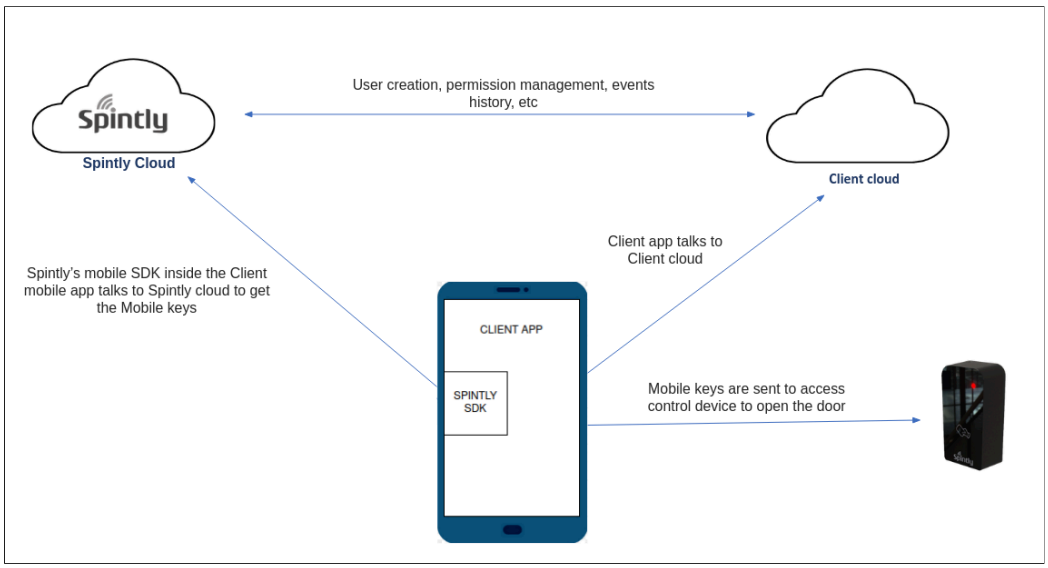
Features	Available
SPINTLY SAAMS API	YES
SPINTLY ACCESS SDK	YES
SPINTLY SAAMS APP	NO

SPINTLY SAAMS WEBSITE	NO
SPINTLY PARTNER WEBSITE/APP	YES
SPINTLY PARTNER SDK/API	YES(if type 5 integration is used)

Entities involved in this integration

As you can see in the below diagram, for type 3 integration to work the following entities will be needed, below d

1. Spintly Cloud: here different apis will be exposed which the client can use
2. Client Cloud: here the backend of the client will be hosted this backend will be reponsible for talking to the
3. Spintly SDK: Here the spintly access sdk will be given to the client, client must include the sdk in its app, th functions which the client app can use to access the doors
4. Client app: This will be front facing app for the customer, apart from access client will be exposing differer
5. Spintly Reader: Once the user is logged in the client app, he can tap on the reader and open the door



Steps to get up and running

Please refer the following link to get up and running along with type 6 integration

please refer the following link to get up and running without any other integration

Examplexample of how you can integrate type3 with your application

Mobile SDK

Please refer the following link to know in depth how to integrate the spintly access sdk

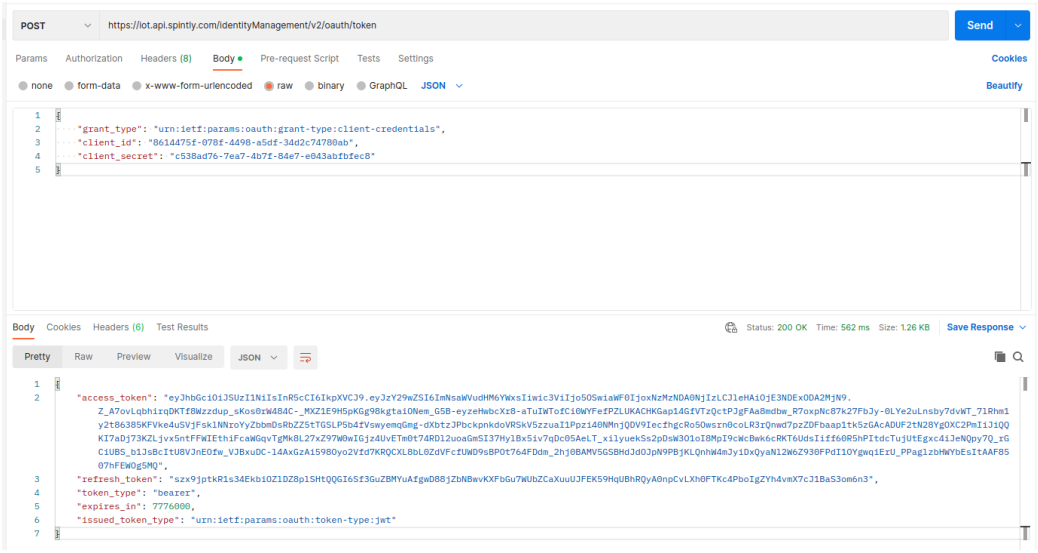
Following are the apis

1. Oauth apis
2. Get all sites
3. Get access points of a site
4. Create accessor
5. Get accessor permissions
6. Update accessor permissions
7. Get access point permissions
8. Update access point permissions

9. Assign Credential
10. Unassign Credential
11. Delete accessor

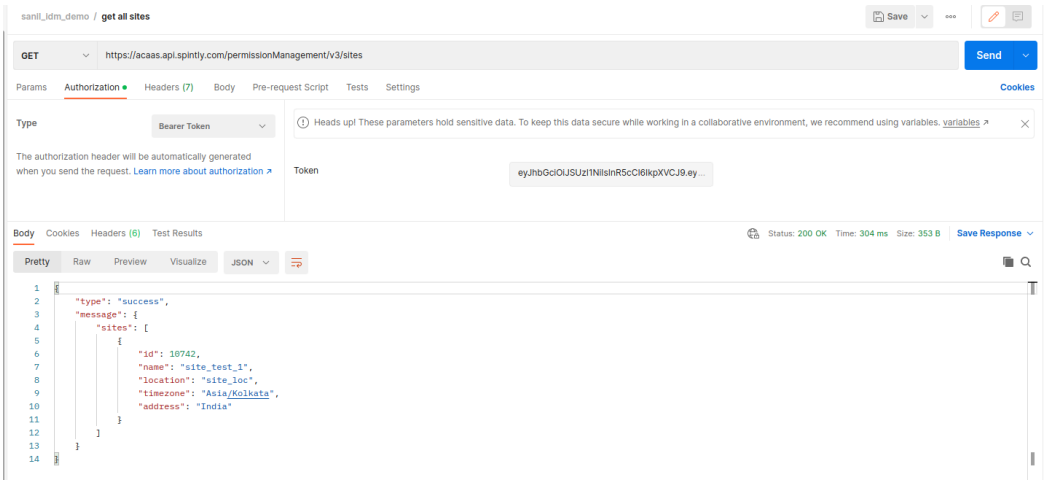
1) Ouath Api

Spintly uses oauth 2.0 client credential flow for authorization purpose, here in the api the client just has to put a clientid and clientSecret to get an access token, this access token can then be used to access other apis of Spintly.



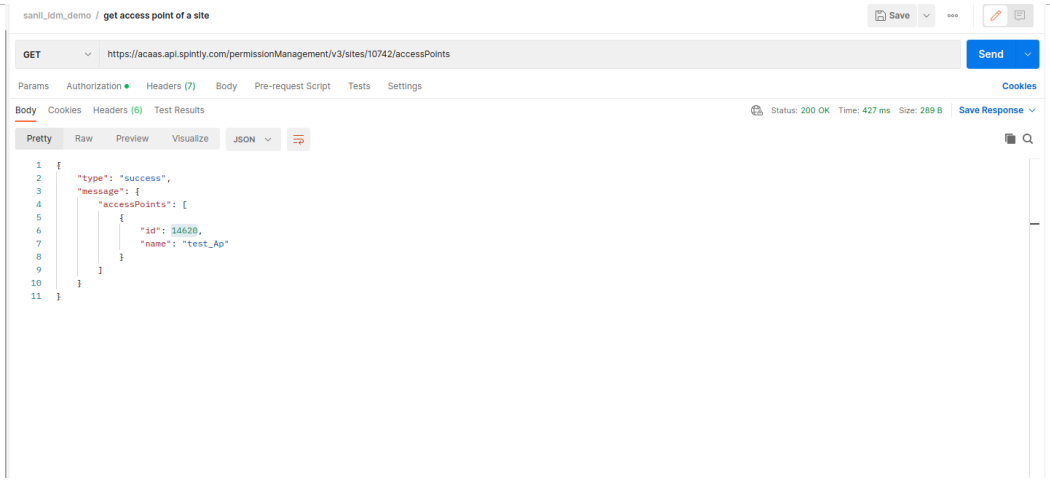
2) Get Sites

This api is responsible for getting all the sites belonging to the Integrator, As you can see in the diagram



3) Get Access points

This api is responsible for getting all the access points(doors), that belong to a particular site



4) Create Accessor

This api is responsible for creating an accessor(user), while creating the accessor, you will need to provide the organisation, you can also specify the permissions while creating the user(not mandatory, you can give permissions after creating the accessor also, you can check below few apis for the same), while giving permissions you can mention what kind of modes of permission the accessor can get ie mobile, card,fingerprint or face, in mobile there are more modes namely tap to access, click to access remote access etc, if Mobile access is given then the identity info needs to be present ie providerid(this will be provided by spintly) and sub(this is unique identifier which will be present in the jwt token which will be passed to the mobile access sdk). When the accessor is created it will give you an accessorid in the response which needs to be mapped to the userid in you software/backend

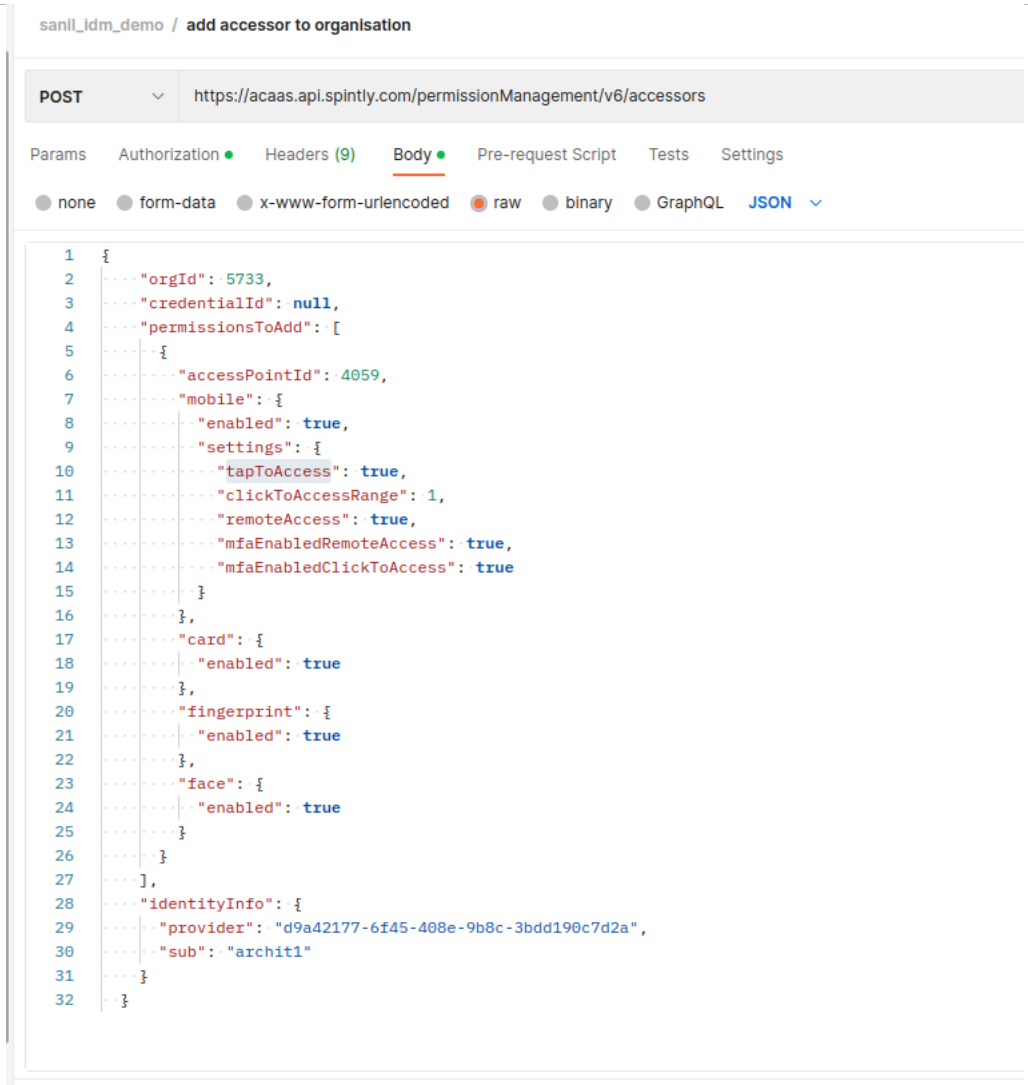
Modes Of access

Mobile
Card
Finger Print
Face

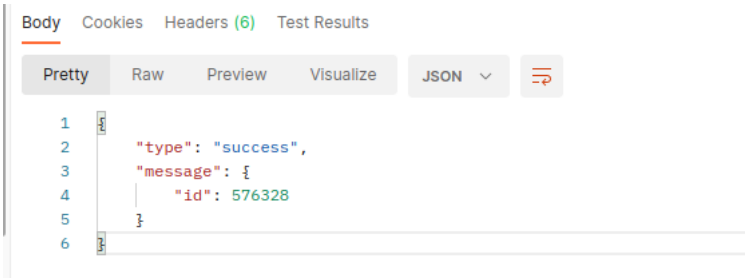
Modes of Mobile Access

Tap To Access
Click To Access
Remote Access

Request Body

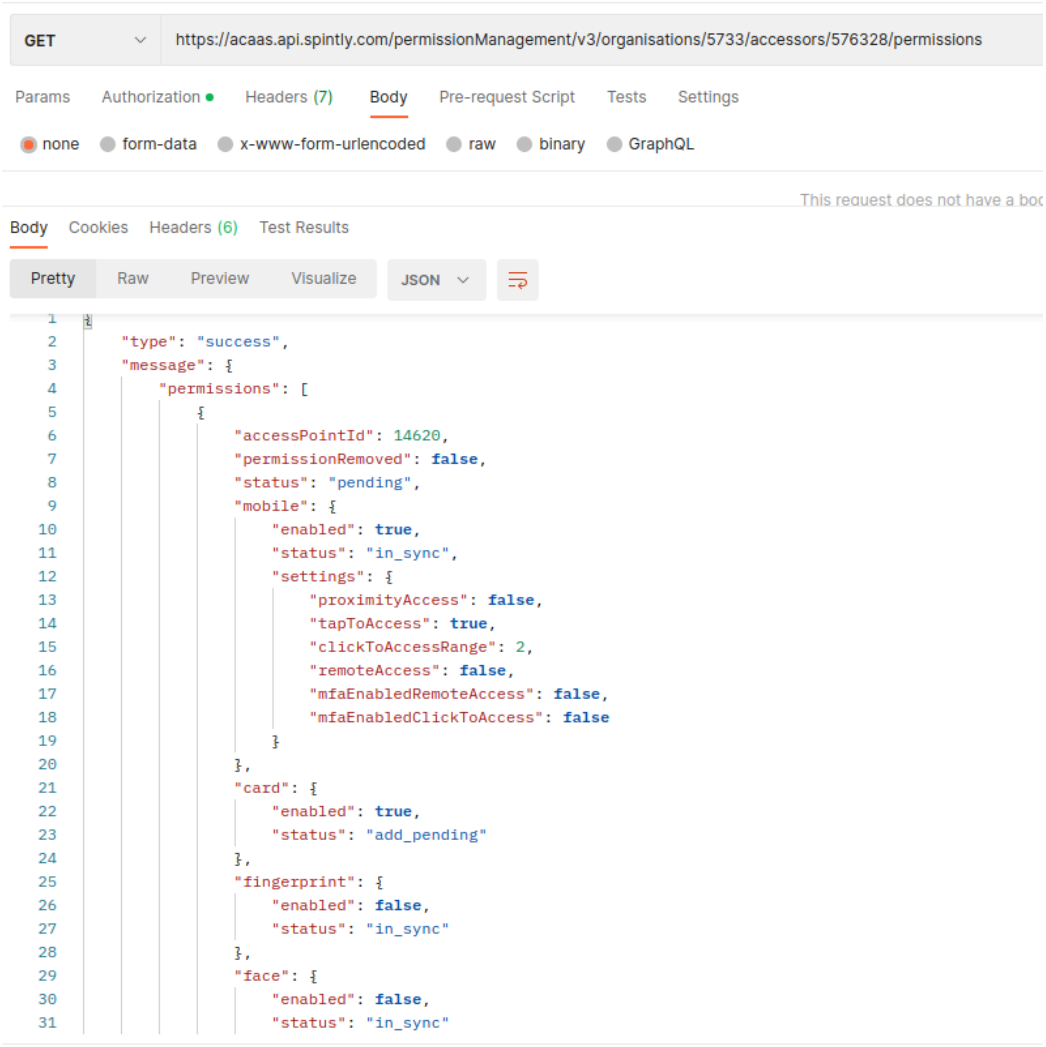


Response



5) Get accessor permissions

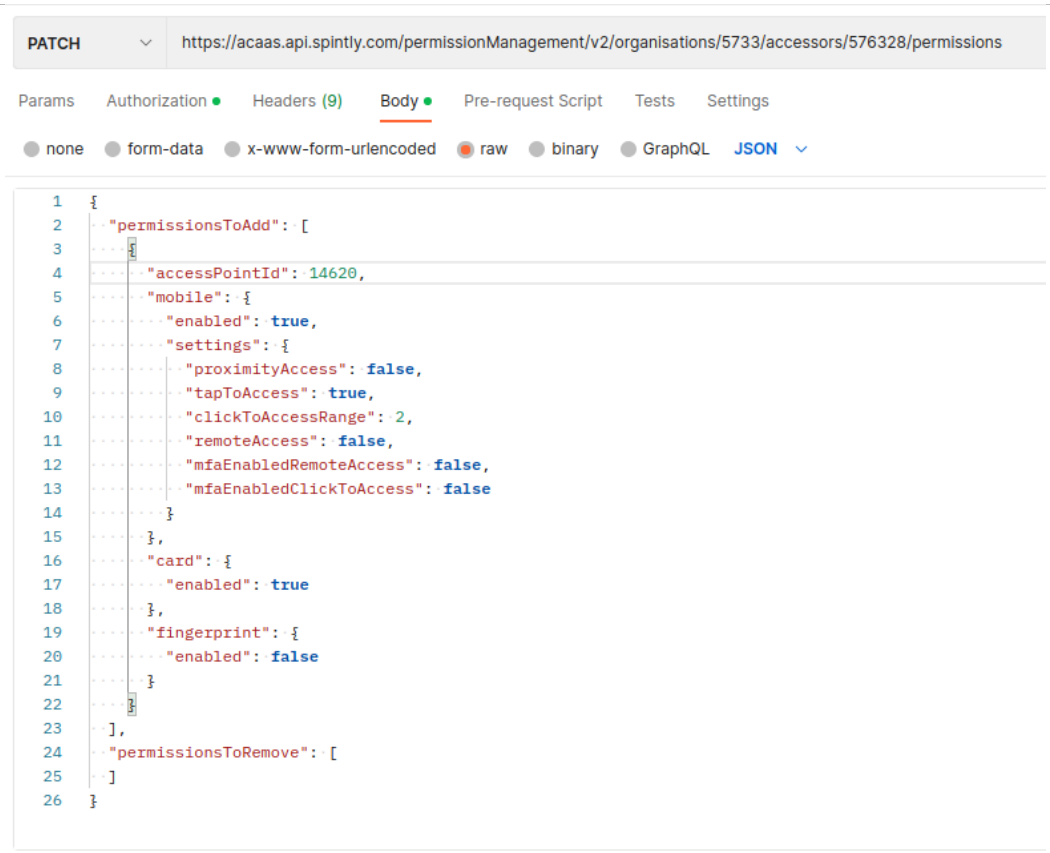
This api is responsible for for getting all the access point permissions of the accessor



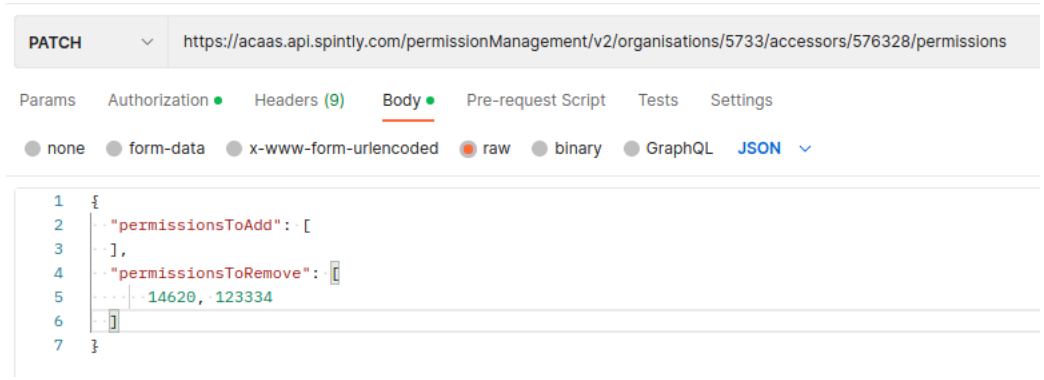
6) Update accessor permissions

This api is responsible for updating the accessor permissions, in the permissionsToAdd you can specify what access point permission to give. Here the request body is similar to create accessor. In the permissionstoRemove you can specify the permissions to be removed. Please note you cannot add and remove the same access point permissions at the same time. The below combinations are allowed.

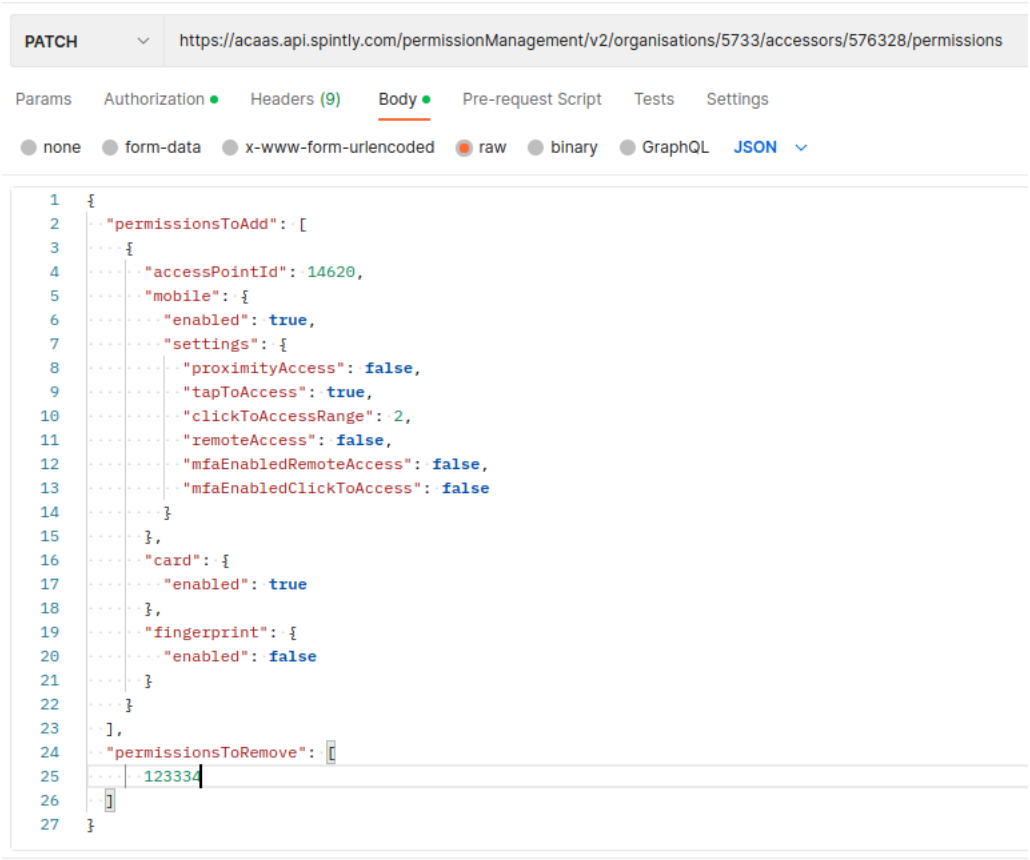
1. Updating only permissionsToAdd



2) Updating only permissionsToRemove

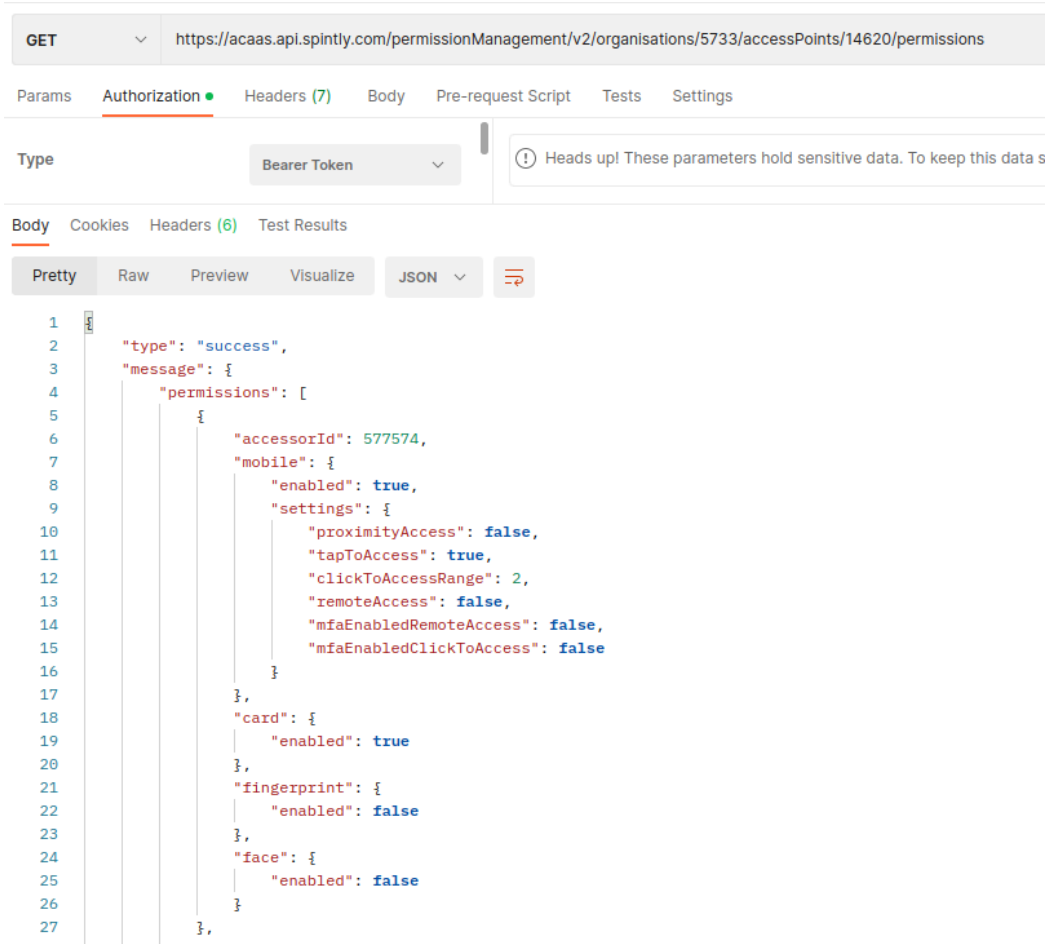


3) Updating both permissionsToAdd and permissionsToRemove



7) Get access point permissions

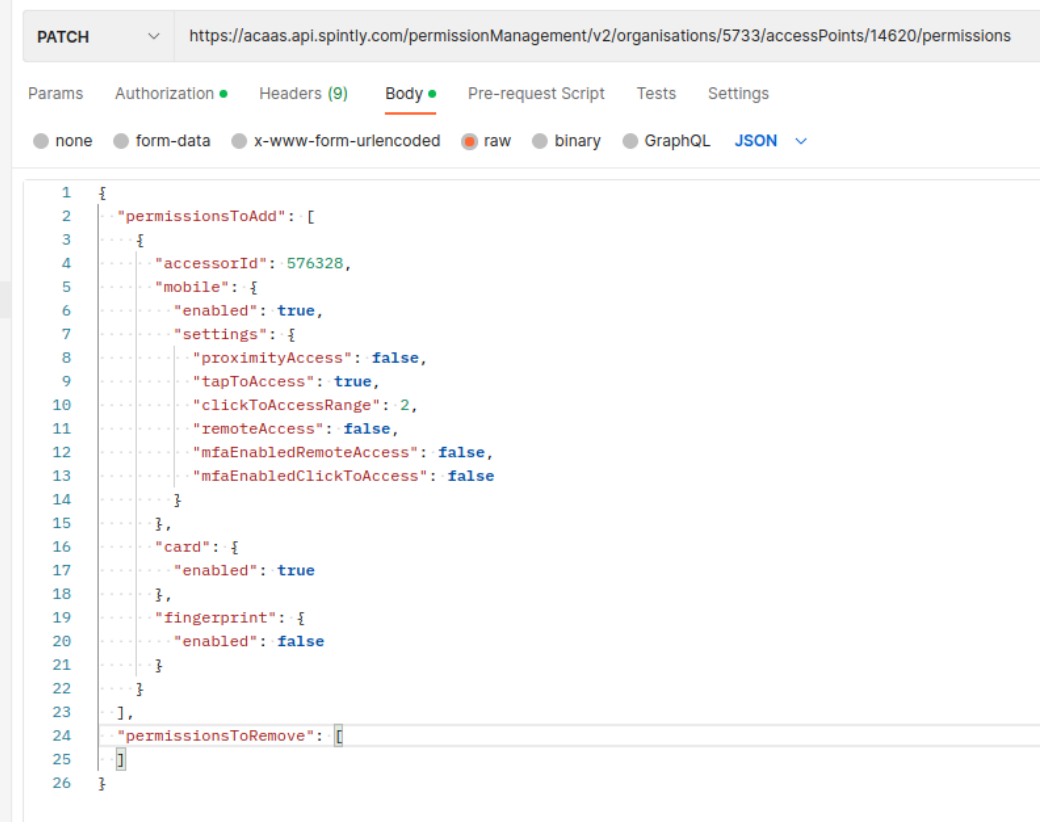
This api is responsible for for getting all the accessor which has permissions to a partiicular access point



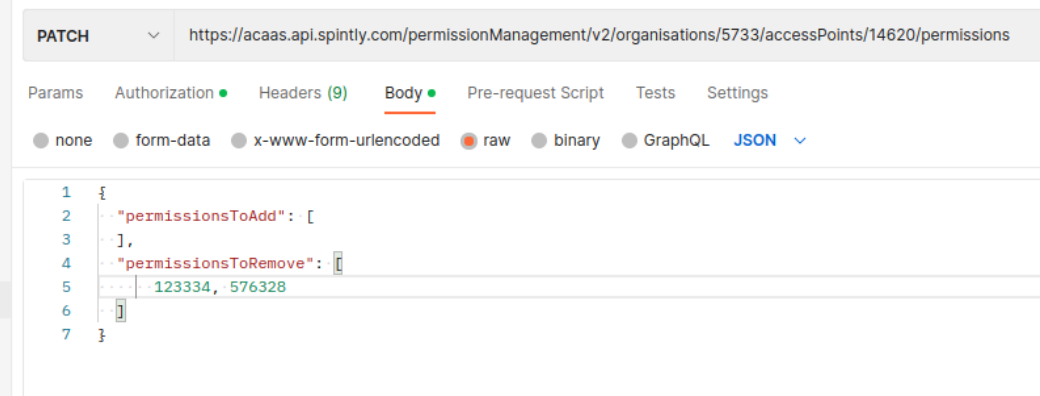
8) Update access point permissions

This api is responsible for updating all the accessors on a particular access point. Please note this api may look similar to update accessor permissions api, only difference is there for particular access you can give different access point permissions, whereas in this api for a particular access point you can give permissions to different accessors. Please note you cannot add and remove the same access point permissions at the same time. Like update accessor permissions api, here also different combintation are allowed:

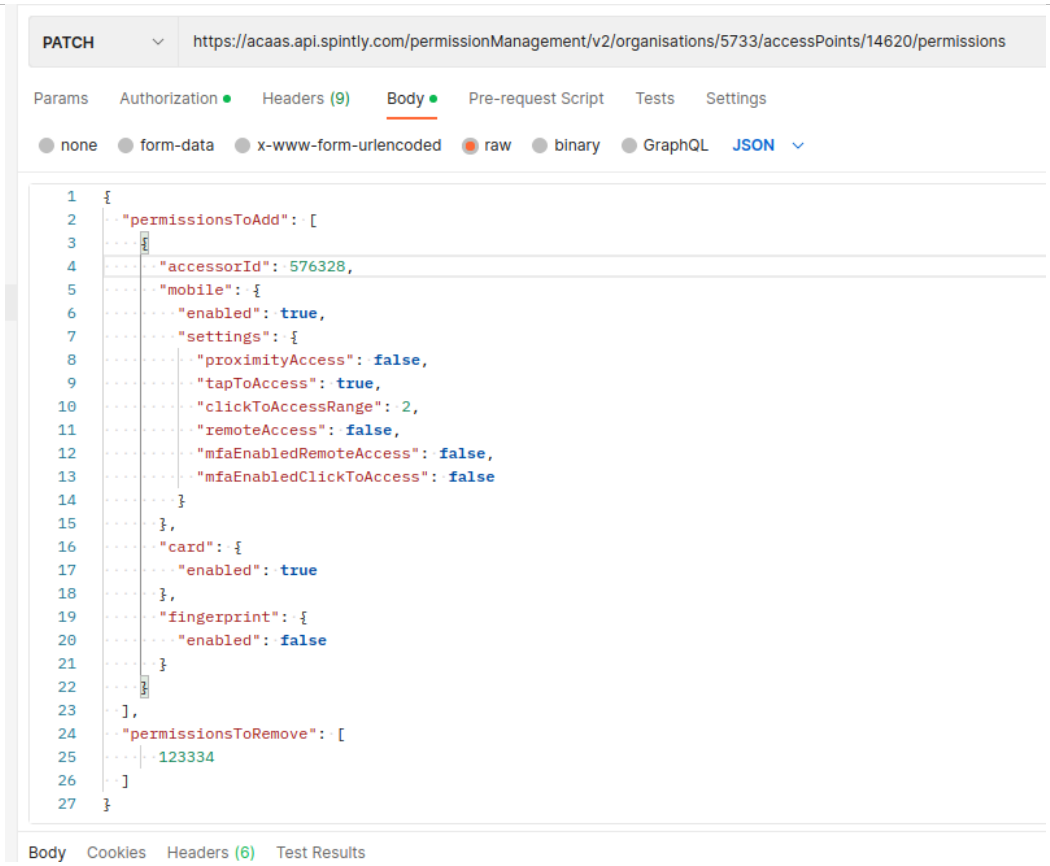
1.



2) Updating only permissionToRemove

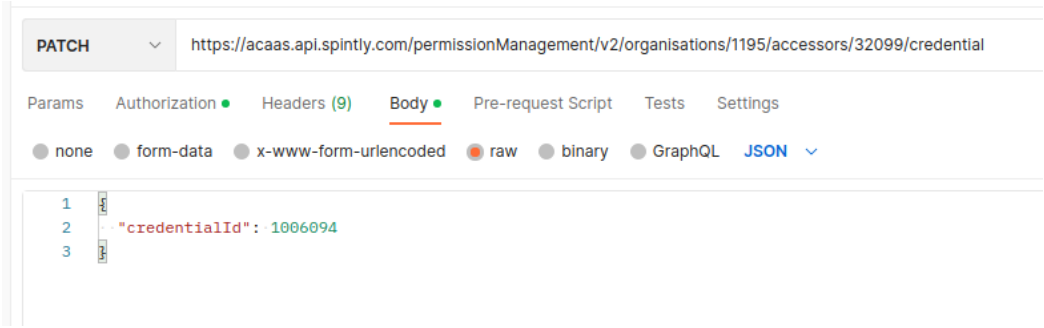


3) Updating both permissionsToAdd and permissionsToRemove



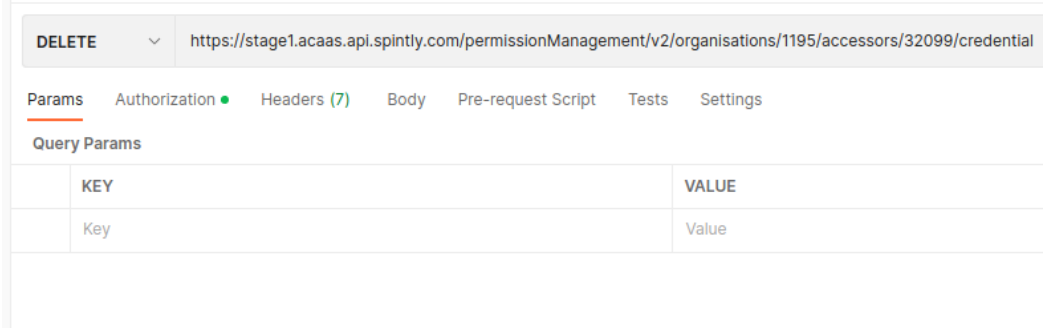
9) Assign Card Api

This api is responsible for assigning a card to an accessor



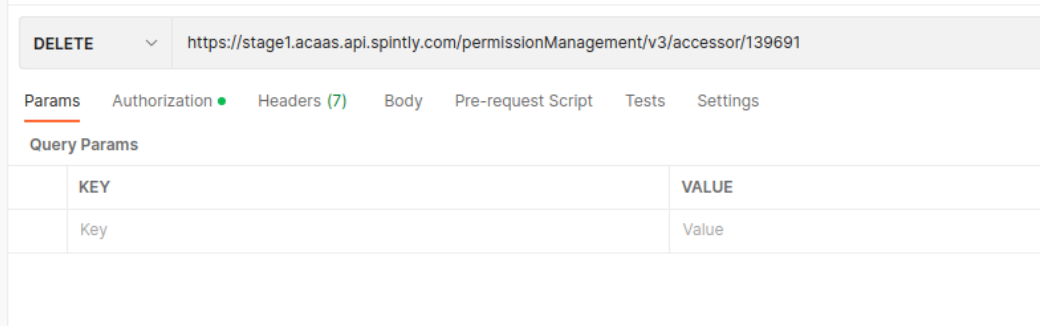
10) Unassign Card Api

This api is responsible for unassigning a card of an accessor



11) Delete Accesor Api

This api is responsible for deleting an acccesor



12 User Schedule

12a) Create Schedule

This api is responsible for creating user schedule, Steps how to get user schedule working are as followed

1. You need to first create a schedule
2. Create an accessor and assign the accessor permissions
3. Assign the user schedule

The request body of create schedule has the following

name: When creating a schedule, please make sure a meaningful name is given

sections: Here in this section you can specify what will be settings of the schedule for each day of the week

repeatType: In sections, there will be field called as repeatType, it has two value ie once or weekly, once means if the schedule needs to be triggered once only, if it once you need specify the date when it will get triggered. weekly means the schedule will get always triggered every week

daysOfWeek: In section, there will be field called as daysOfWeek. Here you need to specify the days in integer format, below are the values

Day	Value
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6
Sunday	7

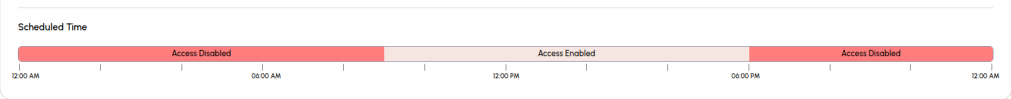
Triggers: In section, there will be a field called as triggers, triggers basically tell which time section the door should deny access to the user and when the door should grant access to the user, it has two fields ie state and time, The time field you can either specify in minutes or in hours and below are the values for the same

State
deny
allow

Once a schedule is created it will give a scheduleId and different sectionId. When creating a schedule it is mandatory to put the holiday section also(ie christmas 25th December, new years 1st January), the holiday section will always overwrite the existing schedule

Please check the below section how you can use the above in an api

a) Creating a user schedule from 9am to 6pm for the all the days of the week (Case 1) and also specifying the holidays also (25th dec)



Below is how the requestBody will look like, here the triggeres section states that from 00:00 to 9:00 the user will be getting access denied, from 9:00 to 17:00 the access will be enabled for the user and from 17:00 to 23:59 the user will be getting access denied

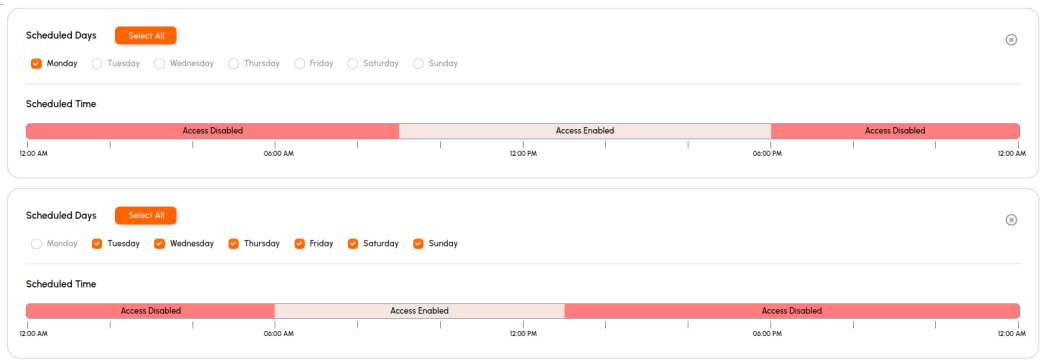
POST https://us.acaas.api.spintly.com/permissionManagement/v1/organisations/654/schedules/user

Params Authorization Headers (9) Body Pre-request Script Tests Settings

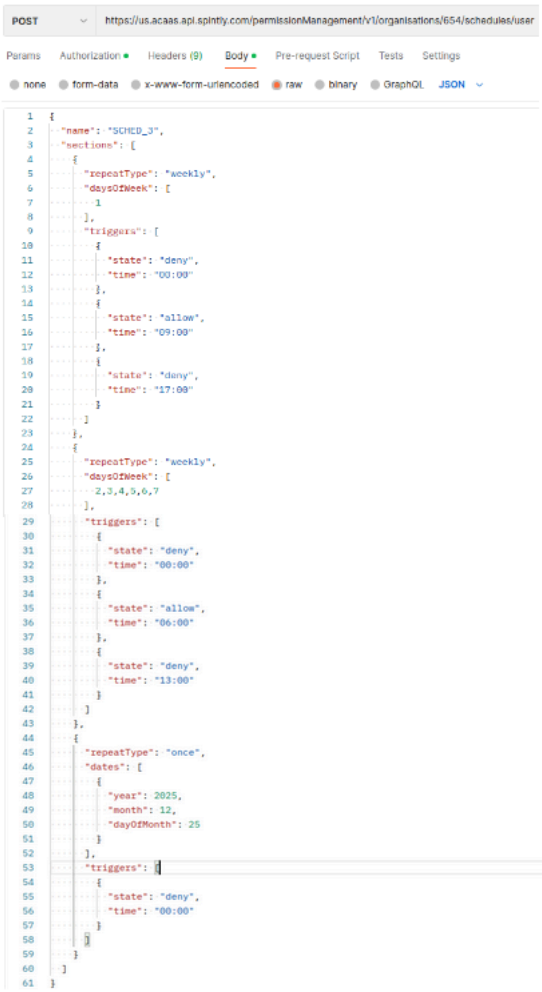
none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
  "name": "SCHED_3",
  "sections": [
    {
      "repeatType": "weekly",
      "daysOfWeek": [
        1,2,3,4,5,6,7
      ],
      "triggers": [
        {
          "state": "deny",
          "time": "00:00"
        },
        {
          "state": "allow",
          "time": "06:00"
        },
        {
          "state": "deny",
          "time": "13:00"
        }
      ]
    },
    {
      "repeatType": "once",
      "dates": [
        {
          "year": 2025,
          "month": 12,
          "dayOfMonth": 25
        }
      ],
      "triggers": [
        {
          "state": "deny",
          "time": "00:00"
        }
      ]
    }
  ]
}
```

a) Creating a user schedule from 9am to 5pm for Monday, and 6am to 1pm for the other days (Case 2)



Below is how the requestBody will look like, as you can see there are 2 entries in the sections, on entry has the setting only for monday schedule, where as the second section has settings for Tuesday,Wednesday, Thusrday, Friday and Saturday. Here mondays trigger section states that from 00:00 to 9:00 the user will be getting access denied, from 9:00 to 17:00 the access will be enabled for the user and from 17:00 to 23:59 the user will be getting access denied. Whereas for the rest of the week the second section states that from 00:00 to 06:00 the user will be getting access denied, from 06:00 to 13:00 the access will be enabled for the user and from 13:00 to 23:59 the user will be getting access denied.



As you can see above each week of a day can have its own schedule, it all depends on a companies use, some company may want to have the same schedule for all the days of the week, some companies may want to have different schedule for all different days of the week. Basically there can be different permutation and

combination of the schedules. but whenever a schedule is created all the days of the week need to be specified, otherwise schedules cannot be created

When a schedule is created, the api will always give a schdeuleid and sectionids(depending on how many schedules are created), below are the response for the above two cases

Case1 response

Basically it will return you the same thing what you had mentioned in the requetbody, only for few things, it will return a scheduleId(highlighted in blue) and sectionId(highlighted in red), this values are important, as using the ids you can update or delete the schedule accordingly

Case2 response

It will return a response identitcal to case1, but since in case 2 there were different sections, it will return you two section ids, for explanation sake , the response has been minimized, here the red arrow indicates section 1 id (ie of mondays) and orange arrow indicates section 2 id (ie of Tuesdays, Wednesdays, thursdays, Fridays and aturdays)

12b) Getting all the user schedules of an organisations

This api is responsible for getting all the schedule for the organisaitons

12c) Getting details of a particular schedule in depth

This api is responsible for getting details of a particular schedule

12d) Updating an existing schedule

This api is responsible for getting updating existing schedules, here lets try to update the schedule already created in the create schedule section, pleas refer case1 and case 2 of create schedule for more context

case 1 updating

Say you wan to update the end tie for case 1 from 17:00 to 19:00, then the reuqest body will look like below, as you can see below the scheduleId(marked as organe) in the url is the same id which we got while creating the first schedule. In the request body there is a field called as update, the update is nothing but the sectionid which you need to update,

case 2 updating

Case 2 updating is the same as case 1 updating, only different is, in case 2 there were 2 sections created, you can either update both the sections o update only one section

updating 2 sections on an existing schedules, as you can see you need to specify both the section ids(red and orange) if you want to update both

12e) Deleting an existing schedule

This api is responsible for deleting an existing schedules, as you can see you just need to specify the orgid and scheduleid

Related Articles

Door Schedules

Organization Settings

Can I unassign and assign the same card to another user?

12f) Assigning schedule to an accessor

This api is responsible for assigning a schedule to an accessor, here you can specify which alll accessors needs to given the schedule, also it needs to tell which access point the schedule needs to be applied. Multiple schedules cannot be assigned on the same door as of now

12g) Unassigning schedule to an accessor

This api is responsible for unassigning a schedule to an accessor, here you can specify which alll schedules needs to be removed from an accessor

12i) Get the schedules assigned to a user (accessor filter)

This api is responsible for getting all permissions of an accessor and showing which all access points has schedule assigned and which all access point has schedule unassigned

12i) Get the schedules assigned to a user (access point filter)

This api is responsible for getting all permissions of an access point and showing which all access points has schedule assigned and which all access point has schedule unassigned

incorporation complete integration

Was this article helpful?

Yes No

Give feedback about this article

Resources

Case Studies
Blogs
Whitepapers
Webinars
Press
Data Sheets

Contact Us

Sales: sales@spintly.com
Tech Support: support@spintly.com
Get a Quote

How can we help?

Categories

- Overview
- FAQs
- Software Guides
- Hardware Guide
- Integration
  - Spintly Platform Integrations
    - Mobile Docs
    - Spintly Integration Overview
    - Prerequisite for backend api Integration
    - Prerequisite for mobile sdk Integration
    - Steps to get started
    - Type 1 Integration
    - Steps to get up and running (Type 1 integration)
    - Examples of how to use type 1 with your application
    - Type 2 integrations
    - Steps to get up and running (Type 2 integration)
    - Examples of how to use type 2 with your application
    - Spintly Visitor Qr Apis integration
    - Type 3

Home / Integration / Spintly Platform Integrations

Type 1 Integration

Categories

Type 1 Integration

Table of Contents

Use Case for Integration

This integration is normally used when a client already have its own website, and wants to automate granting of accesses

Example: Say you own company called as acme, acme has its own website. When a user joins acme company, the hr has to go to the acme website create user and then go to Spintly website and add user, imagine if there were 1000 users, this can be cumbersome for the Hr. To solve this problem, ACME can integrate using Spintly apis, where in whenever a new user is created, ACME software can call Spintly apis to create user. Also the admin can mark certain doors as default doors which whenever a user is added, he will get the door permission by default without the admin manual intervention. Same thing can be done with update and delete. So during month end , the admin can go to the Spintly dashboard and download all kinds of user reports which can help the efficiency of the organisation

Live example:

This solution is already live and implemented by a client called as ff21. More about FF21 below

FF21 offers best-in-class **co-living spaces** with a strong focus on **experience design**, **community** and **technology.** We currently operate over a 1000+ beds in Bangalore. At the core of FF21, is to create safe and fun spaces for the young millenials who are new to the city and provide the right environment *Where Friends Become Family*. The FF21 brand, is wholly owned and operated by Sherwood Longstay Pvt Ltd.

So with this solution ff21 has automated granting of access and has made life easier for the ff21 staff.

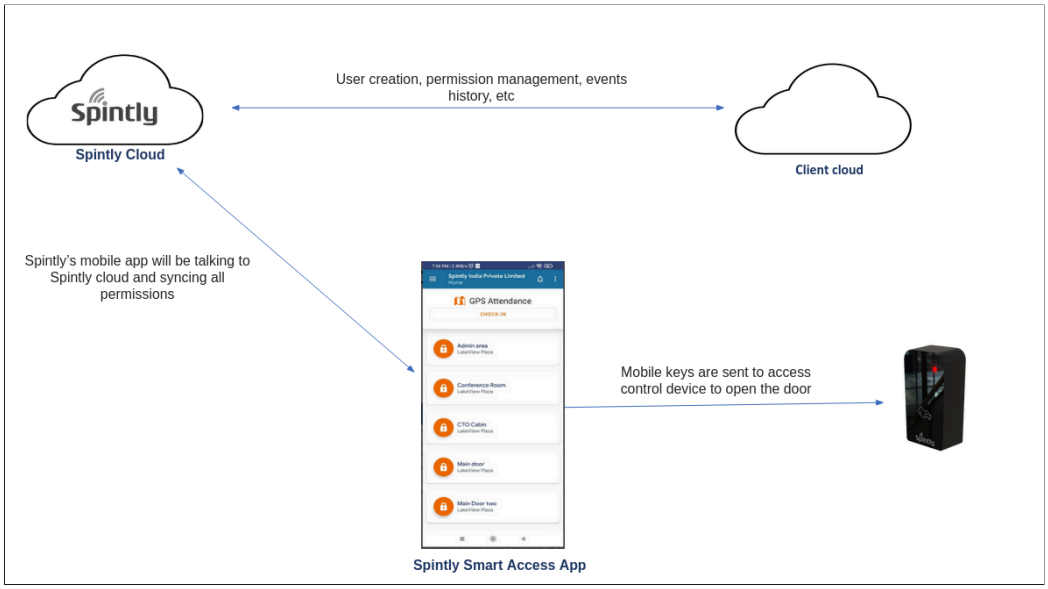
The features to be offered in this integration

Features	Available
SPINTLY SAAMS API	YES
SPINTLY ACCESS SDK	NO
SPINTLY SAAMS APP	YES
SPINTLY SAAMS WEBSITE	YES
SPINTLY PARTNER WEBSITE/APP	YES
SPINTLY PARTNER SDK/API	NO

Entities involved in this integration

As you can see in the below diagram, for type 3 integration to work the following entities will be needed, below c

1. Spintly Cloud: here different apis will be exposed which the client can use
2. Client Cloud: here the backend of the client will be hosted this backend will be reponsible for talking to the
3. Spintly Smart access app: with this app the user will be able to access the doors
4. Spintly Reader: Once the user is logged in the client app, he can tap on the reader and open the door



Steps to get up and running

Please refer the following link to get up and running

Simple example ono how to do changes in your backend/cloud is shown here

Spintly Apis

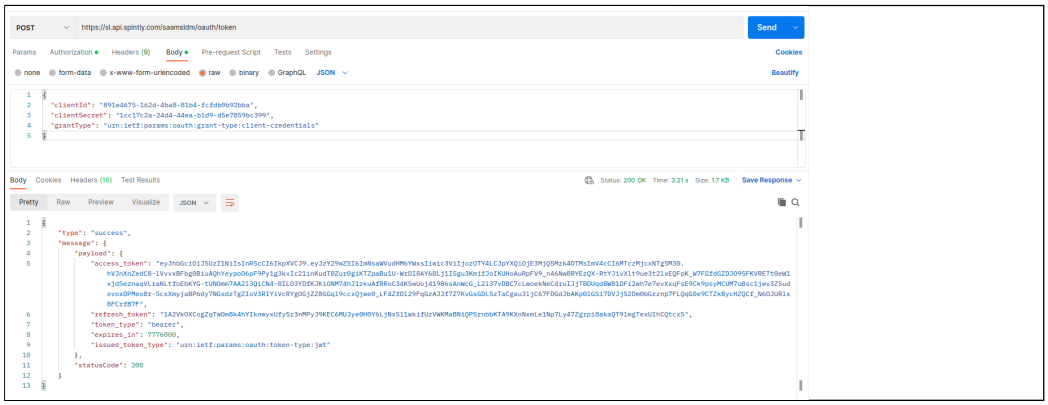
Spintly provides different apis so that another software can seamlessly integrate with spintly
Following are the apis spintly provides

1. Ouath token api
2. Get sites
3. Get Access points
4. Get organisation roles
5. Create a new user
6. Update a user
7. Deactivate a user
8. Activate a user
9. Delete a user
10. Get organisation user list
11. Get user permissions

12. Get all users who have permission assigned to a door
13. Modify permissions of a user

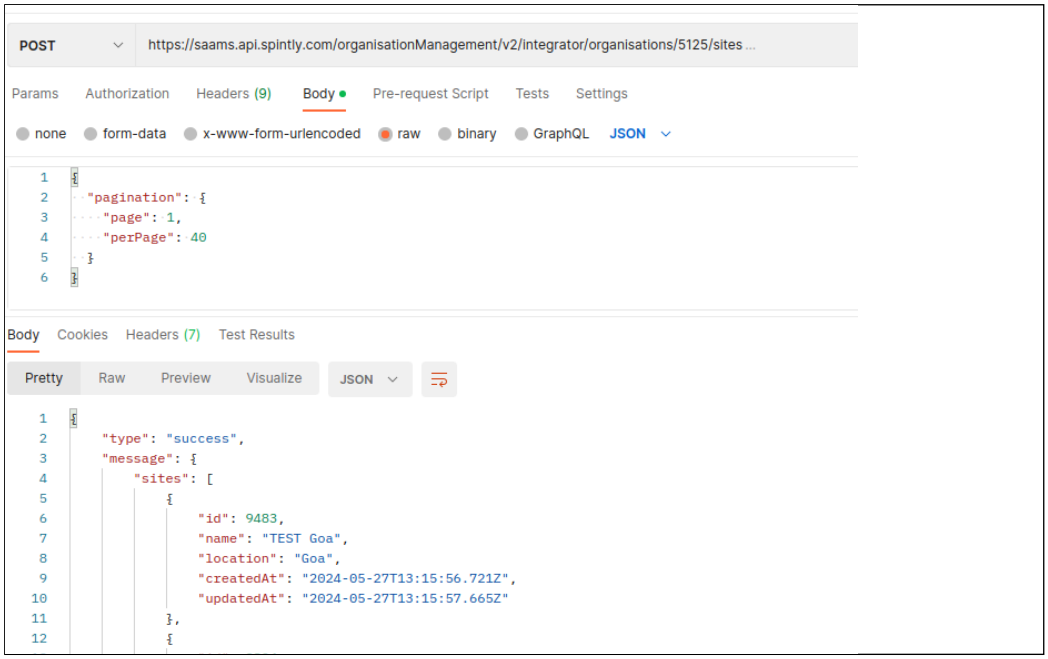
1. Ouath Api

Spintly uses oauth 2.0 client credential flow for authorization purpose, here in the api the client just has to put a clientid and clientSecret to get an access token, this access token can then be used to access other apis of Spintly.



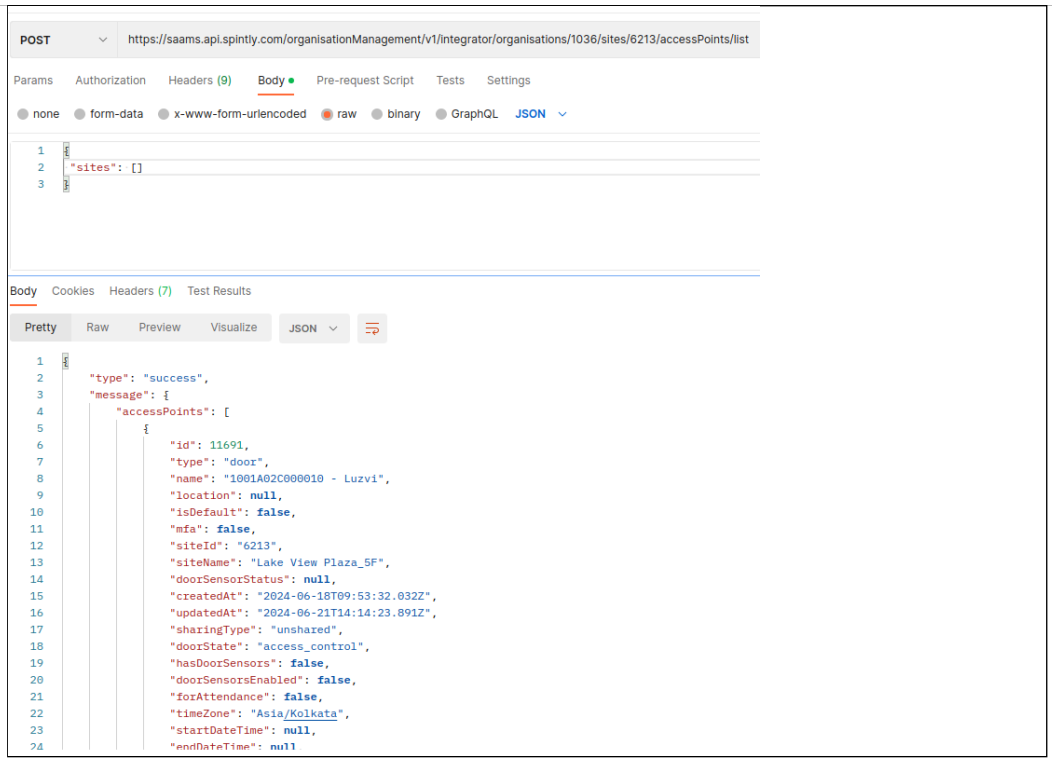
2) Get Sites

This api is responsible for getting all the sites belonging to the organisation, As you can see in the diagram,If there multiple site, spintly provides pagination support also



2) Get Access points

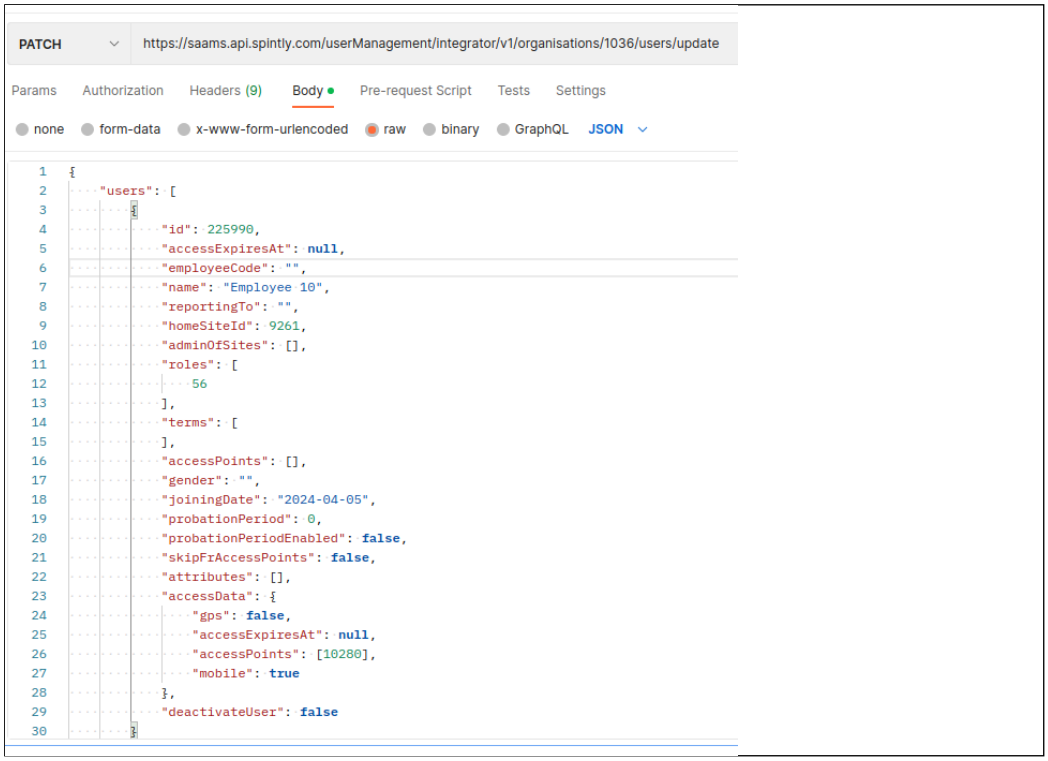
This api is responsible for getting all the access points(doors), that belong to a particular site, like the previous site apis, Spintly provides pagination support if there multiple sites, also Spintly will provide filter support where user can search by access point names



2) Get Organisation roles

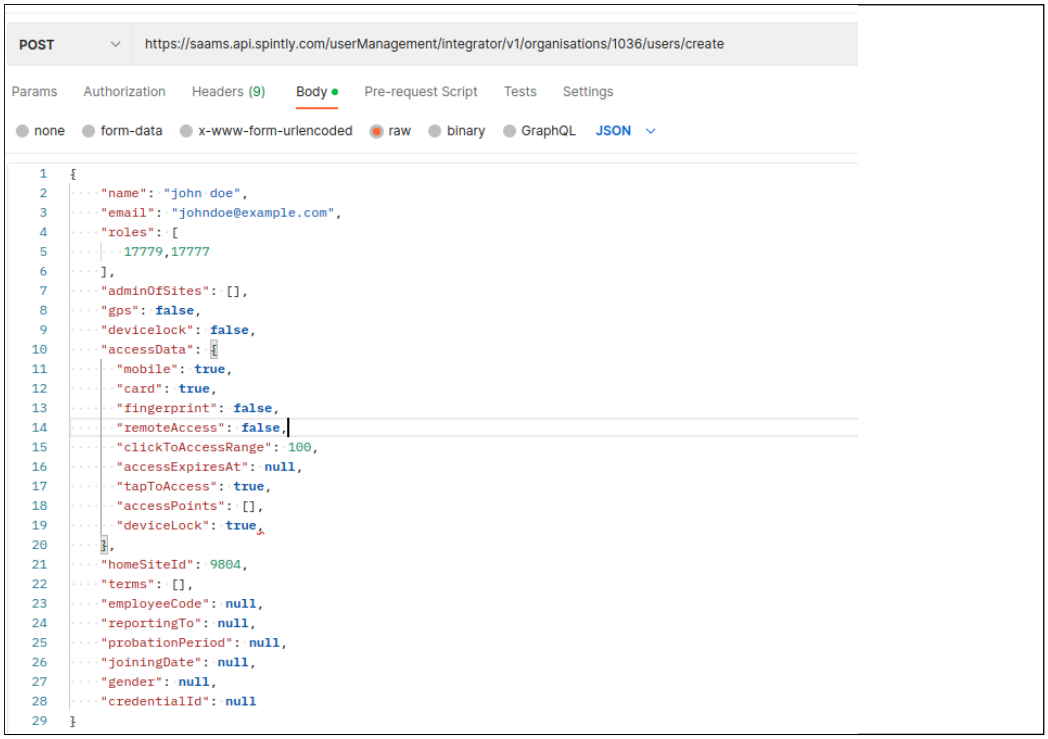
Before creating a user, you need to all the roles belonging to the organisations, Spintly has different roles

1. Spintly user
2. Site admin
3. Super admin
4. Front Desk role(if visitor management module is enabled)
5. Manager role(if attendance module is enabled)



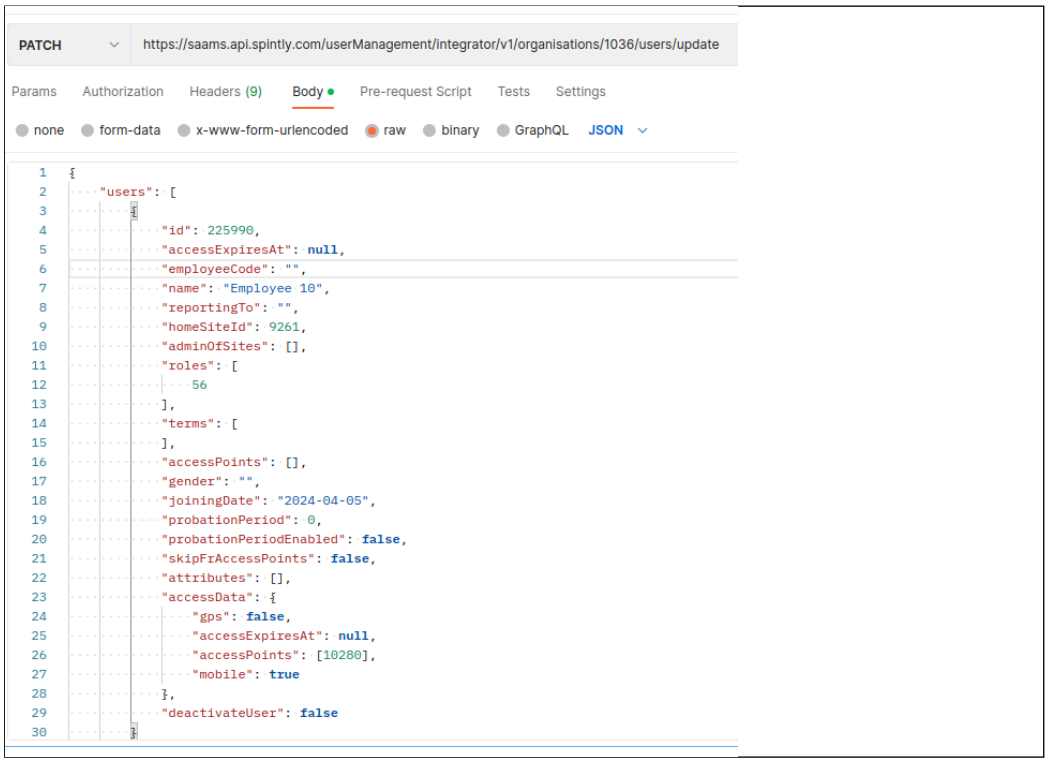
2) Create User

This api is responsible for creating a user in Spintly, when creating the user, you can specify what kind of roles the user gets, what kind of door permissions the user gets, what kind of mode of access the user can have etc. Once the user is created Spintly returns the id of the user



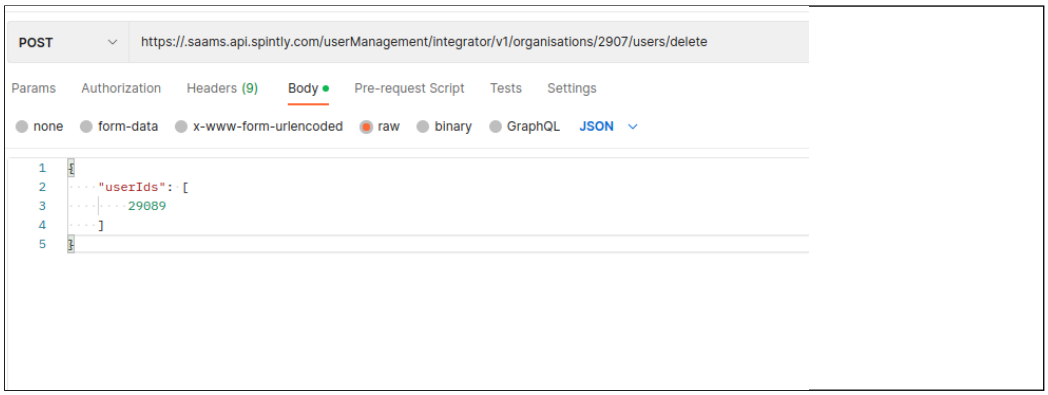
2) Update User

This api is responsible for creating a user in Spintly, when creating the user, you can specify what kind of roles the user gets, what kind of door permissions the user gets, what kind of mode of access the user can have etc. Once the user is created Spintly returns the id of the user



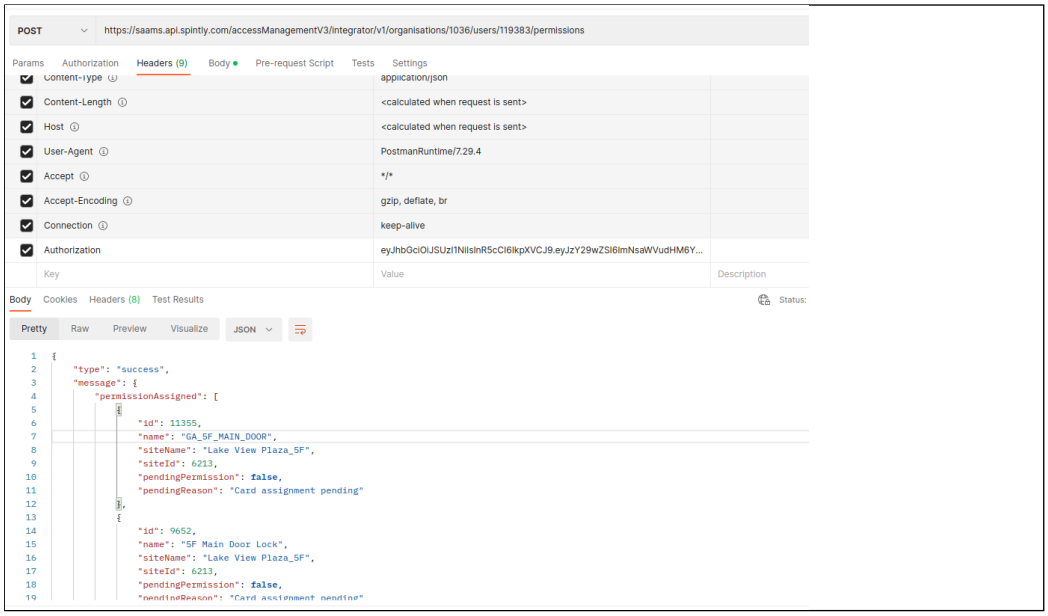
2) Delete User

This api is responsible for deleting the user in Spintly, here the id which was returned when the user was created needs to be specified if u want to delete a particular user



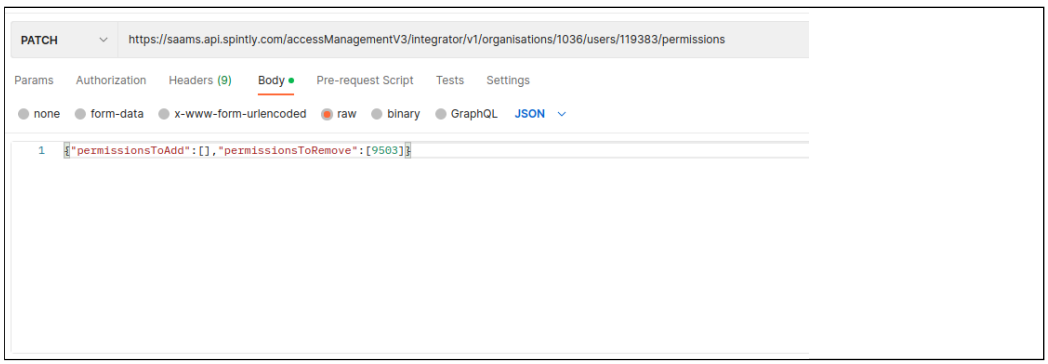
2) Get User Permissions

This api is responsible getting the permissions of a user, it returns 3 things
1) permissionAssigned: this are permissions currently assigned to the user
2) permissionNotAssigned: this are permissions not assigned to the user
3) pendingPermissions: this are permissions which users have but not reached to the device yet



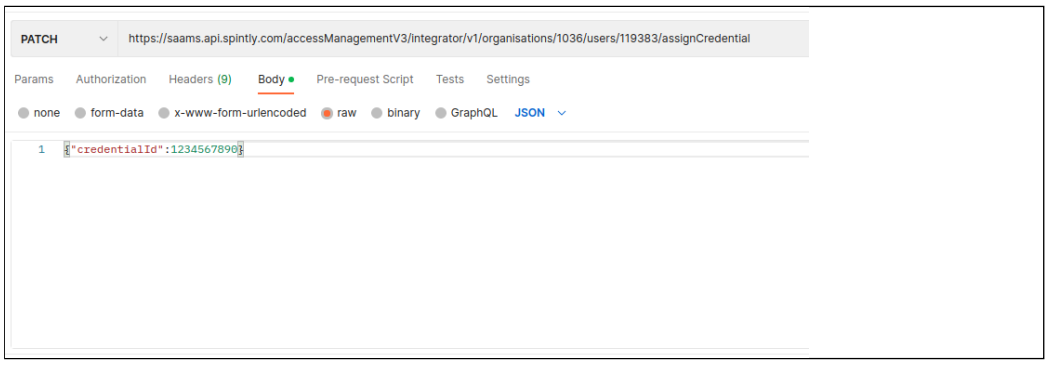
2) Update user permissions

This api is responsible for assigning and unassigning permissions of a user



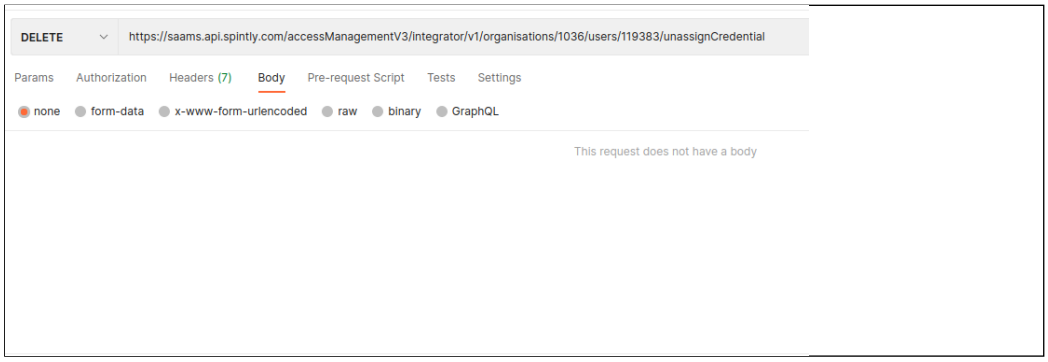
2) Card Assignment

This api is responsible for assigning a card to a user



2) Card Unassignment

This api is responsible for unassigning a card to a user



merge unified

Was this article helpful?

Yes No

Give feedback about this article

Related Articles

Device Management

Fire Integration & Configuration

Door Sensors

Spintly
Amplify the Simplicity

Industries

Multifamily / Residential
Co-Working
Education
Healthcare
Hospitality
Manufacturing
Fitness & Wellness

Solutions

Access Control
Visitor Management
Tenant Management
Time & Attendance Management

About

About Us
Partners
FAQ
Privacy Policy
Terms & Conditions

Resources

Case Studies
Blogs
Whitepapers
Webinars
Press
Data Sheets

Contact Us

Sales: sales@spintly.com
Tech Support: support@spintly.com
Get a Quote

How can we help?

Categories

- Overview
- FAQs
- Software Guides
- Hardware Guide
- Integration
  - Spintly Platform Integrations
    - Mobile Docs
    - Spintly Integration Overview
    - Prerequisite for backend api Integration
    - Prerequisite for mobile sdk Integration
    - Steps to get started
    - Type 1 Integration
    - Steps to get up and running (Type 1 integration)
    - Examples of how to use type 1 with your application
    - Type 2 integrations
    - Steps to get up and running (Type 2 integration)
    - Examples of how to use type 2 with your application
    - Spintly Visitor Qr Apis integration
    - Type 3

Home / Integration / Spintly Platform Integrations

Type 2 integrations

Categories

Type 2 integrations

Table of Contents

Note

Make sure you have gone through the Spintly Integration Overview section, as that section will help you understanding the spintly platform and the types of integration

Use Case for Integration

This integration is normally used when a client already has it own app and the client wants the Spintly SDK to be in its own app rather then using spintly app to give access, with this approach the end customer thinks the client has implemented the access control functionality, it also prevents the end customer from using two apps ie client app for client facilities and spintly app for access, thus enabling seamless experience to the end customer

Live example:

This solution is already live and implemented by lot of our indian customers namely, smartworks, cowrks, anarock, nucleus etc

The features to be offered in this integration

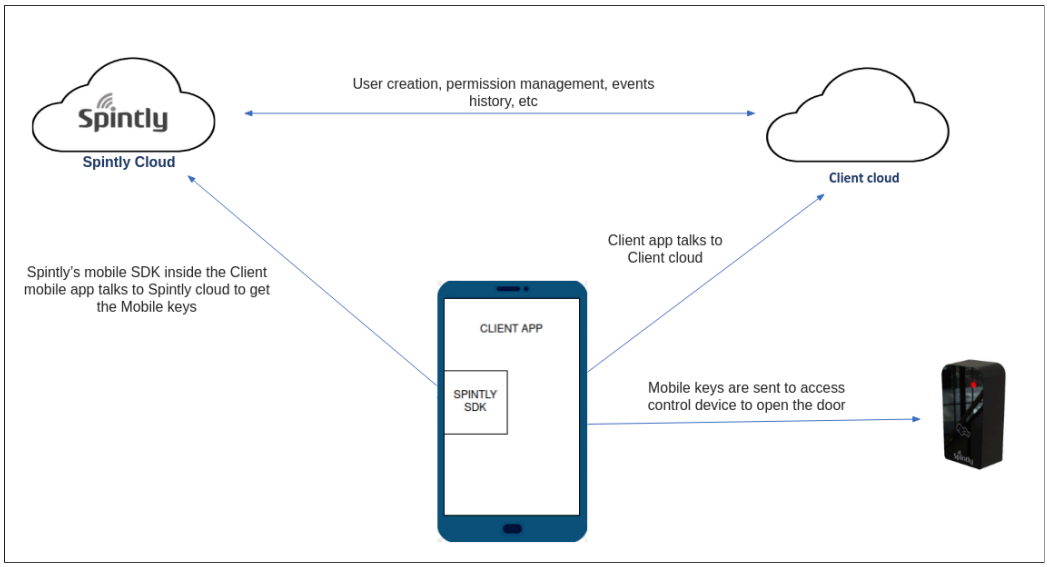
Features	Available
SPINTLY SAAMS API	YES
SPINTLY ACCESS SDK	YES
SPINTLY SAAMS APP	NO
SPINTLY SAAMS WEBSITE	YES
SPINTLY PARTNER WEBSITE/APP	NO

SPINTLY PARTNER SDK/API	NO

Entities involved in this integration

As you can see in the below diagram, for type 3 integration to work the following entities will be needed, below

1. Spintly Cloud: here different apis will be exposed which the client can use
2. Client Cloud: here the backend of the client will be hosted this backend will be reponsible for talking to the
3. Spintly SDK: Here the spintly access sdk will be given to the client, client must include the sdk in its app, th functions which the client app can use to access the doors
4. Client app: This will be front facing app for the customer, apart from access client will be exposing differer
5. Spintly Reader: Once the user is logged in the client app, he can tap on the reader and open the door



Steps to get up and running

Please refer the following link to get up and running along with type 6 integration

Simple example ono how to do changes in your backend/cloud is shown here

Mobile SDK

Please refer the following link to know in depth how to integrate the spintly access sdk

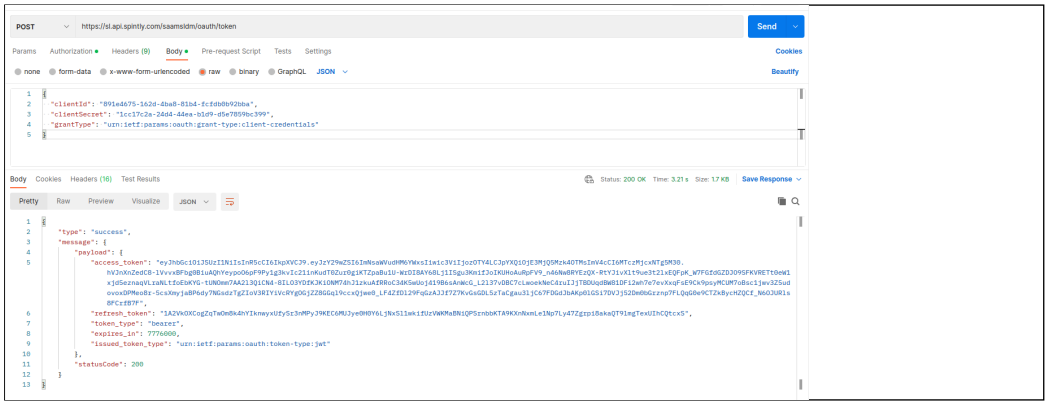
Spintly Apis

Spintly provides different apis so that another software can seamlessly integrate with spintly
Following are the apis spintly provides

1. Ouath token api
2. Get sites
3. Get Access points
4. Get organisation roles
5. Create a new user
6. Update a user
7. Deactivate a user
8. Activate a user
9. Delete a user
10. Get organisation user list
11. Get user permissions
12. Get all users who have permission assigned to a door
13. Modify permissions of a user

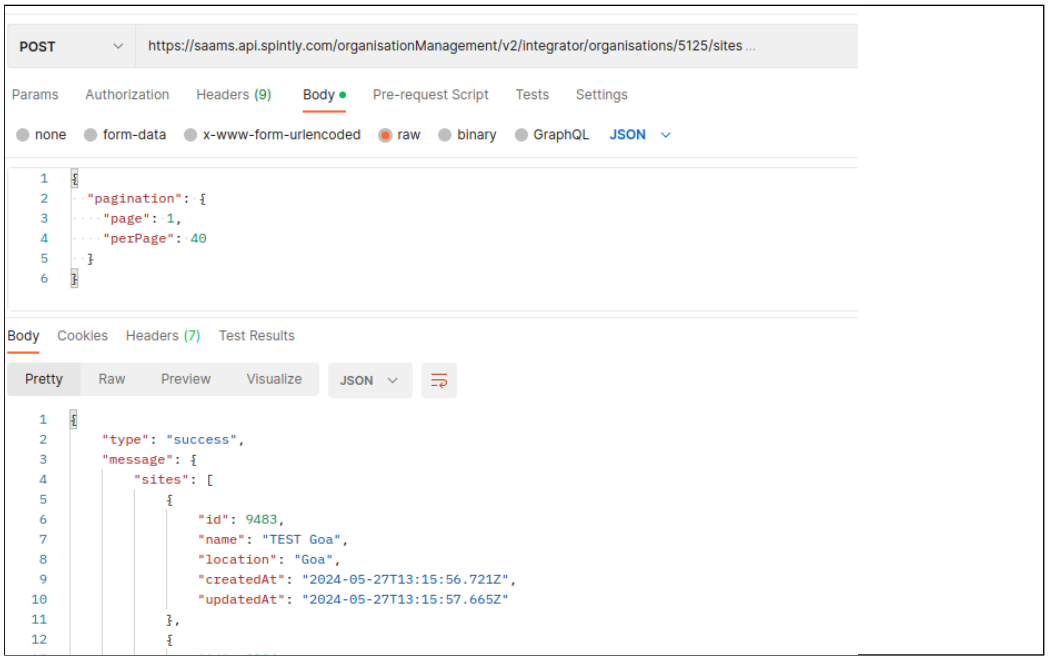
1. ### Ouath Api

Spintly uses oauth 2.0 client credential flow for authorization purpose, here in the api the client just has to put a clientid and clientSecret to get an access token, this access token can then be used to access other apis of Spintly.



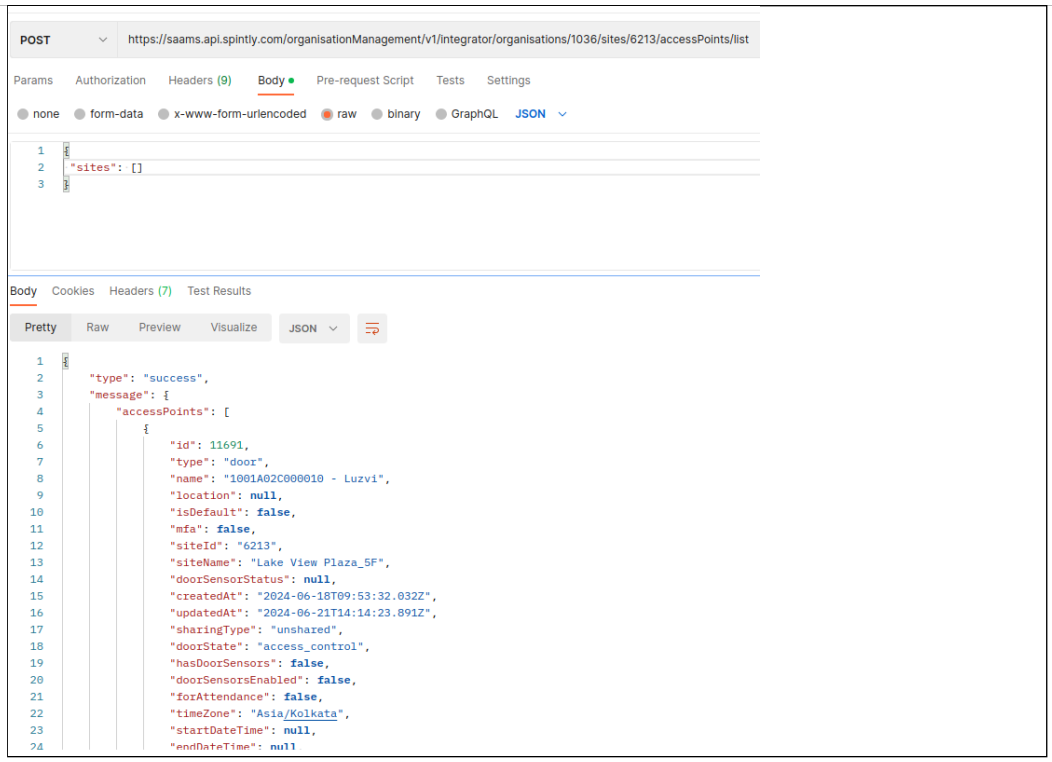
2) Get Sites

This api is responsible for getting all the sites belonging to the organisation, As you can see in the diagram,If there multiple site, spintly provides pagination support also



2) Get Access points

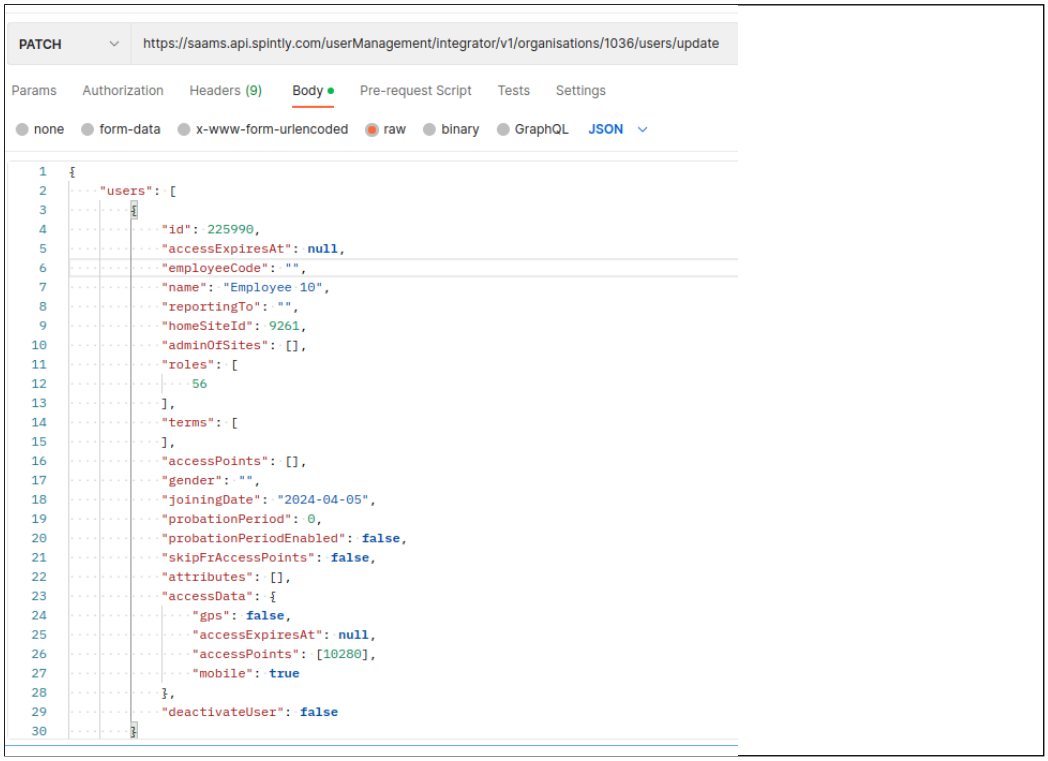
This api is responsible for getting all the access points(doors), that belong to a particular site, like the previous site apis, Spintly provides pagination support if there multiple sites, also Spintly will provide filter support where user can search by access point names



2) Get Organisation roles

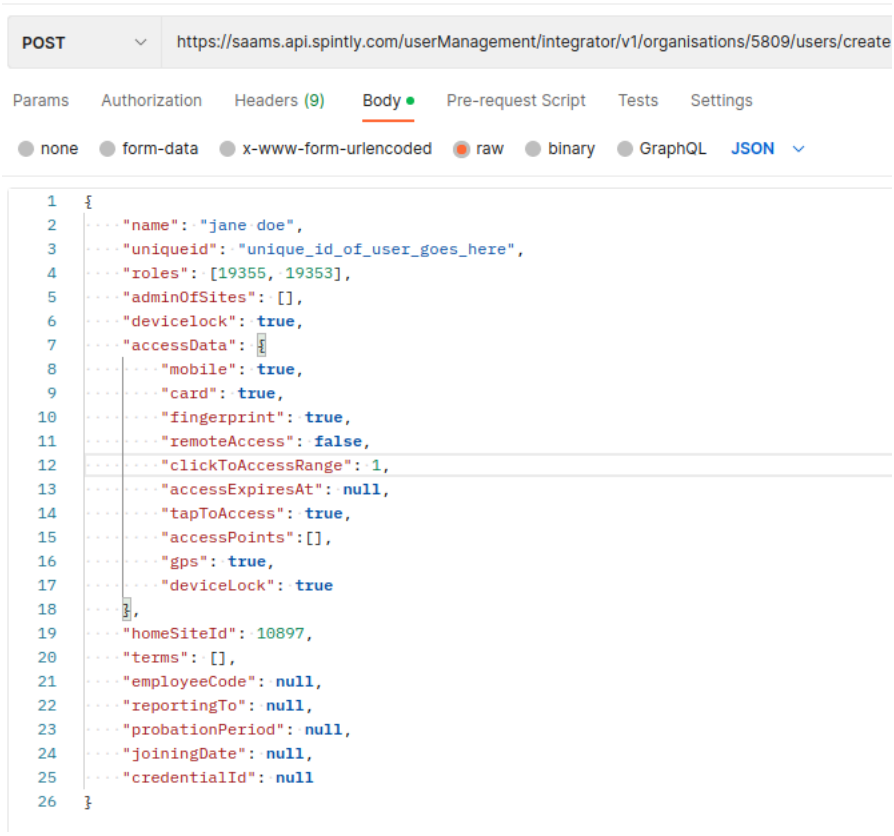
Before creating a user, you need to all the roles belonging to the organisations, Spintly has different roles

1. Spintly user
2. Site admin
3. Super admin
4. Front Desk role(if visitor management module is enabled)
5. Manager role(if attendance module is enabled)



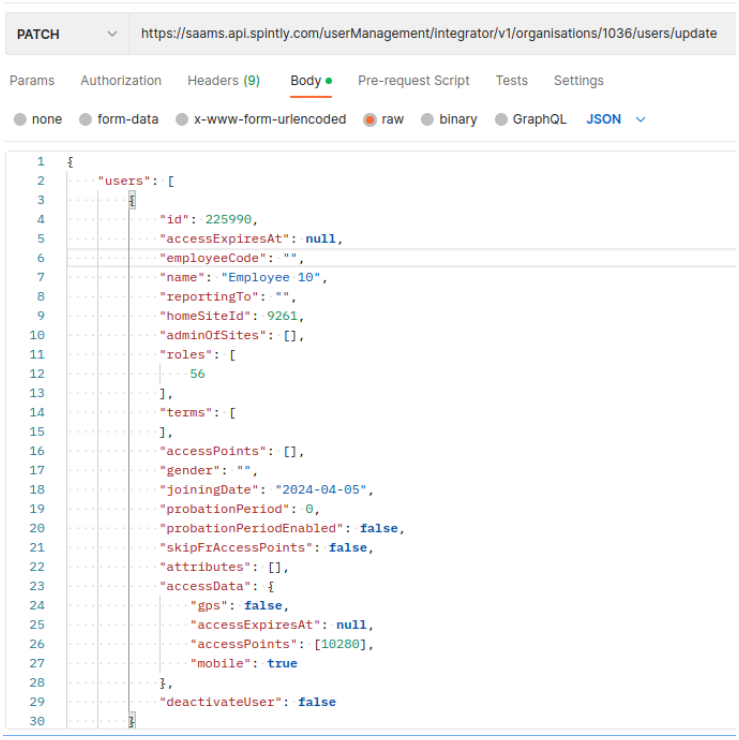
2) Create User

This api is responsible for creating a user in Spintly, when creating the user, you can specify what kind of roles the user gets, what kind of door permissions the user gets, what kind of mode of access the user can have etc. Once the user is created Spintly returns the id of the user



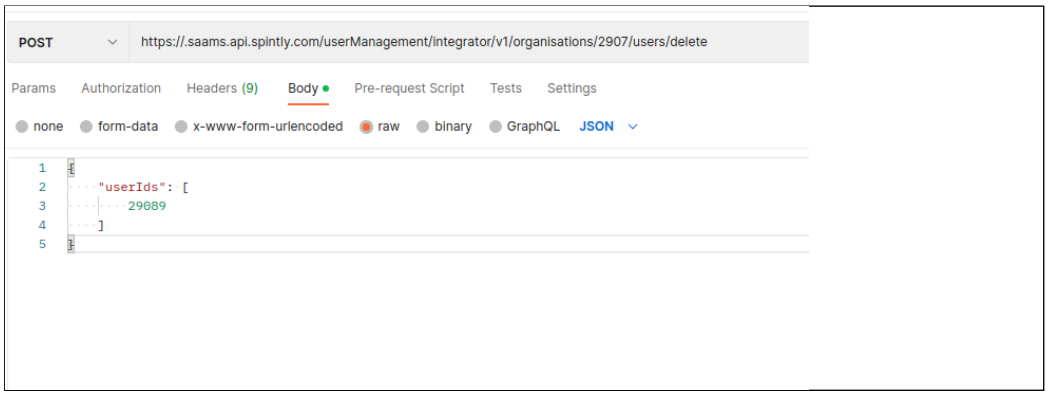
2) Update User

This api is responsible for creating a user in Spintly, when creating the user, you can specify what kind of roles the user gets, what kind of door permissions the user gets, what kind of mode of access the user can have etc. Once the user is created Spintly returns the id of the user



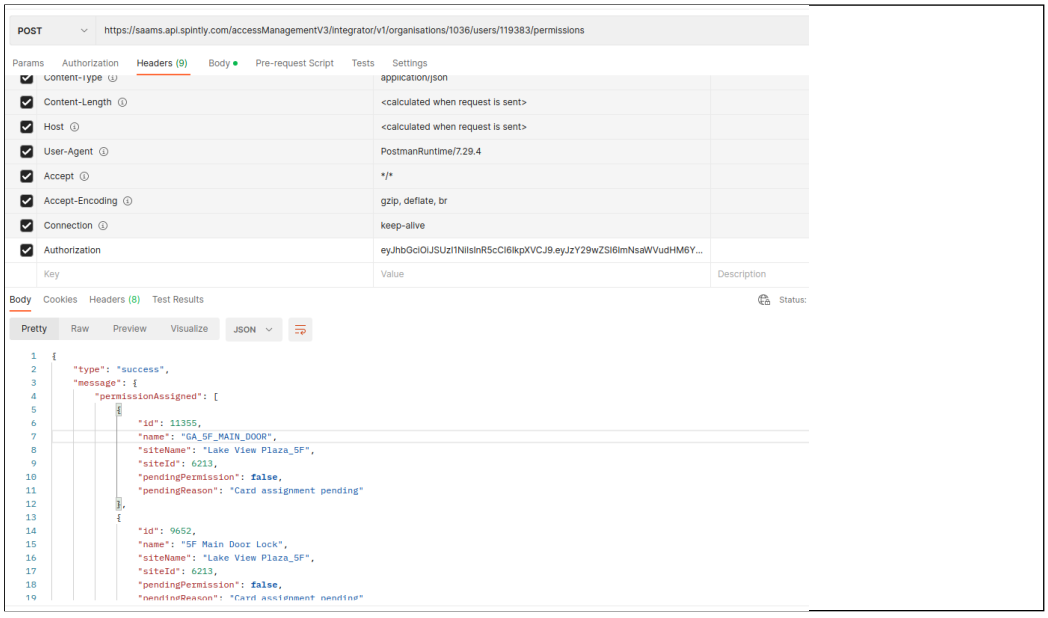
2) Delete User

This api is responsible for deleting the user in Spintly, here the id which was returned when the user was created needs to be specified if u want to delete a particular user



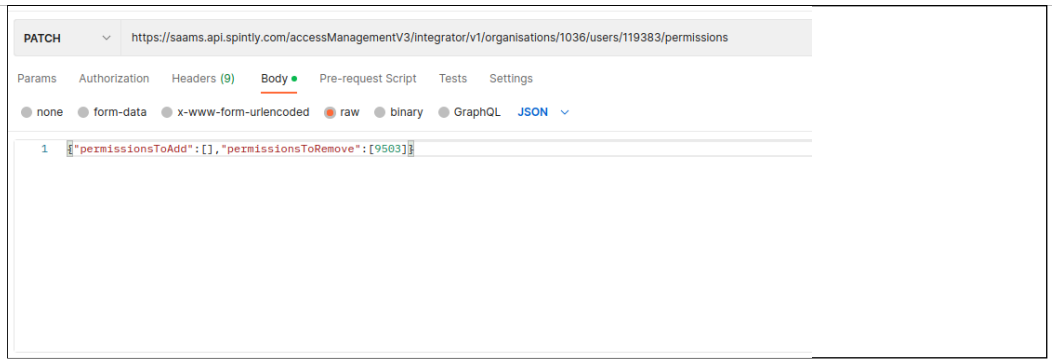
2) Get User Permissions

This api is responsible getting the permissions of a user, it returns 3 things
1) permissionAssigned: this are permissions currently assigned to the user
2) permissionNotAssigned: this are permissions not assigned to the user
3) pendingPermissions: this are permissions which users have but not reached to the device yet



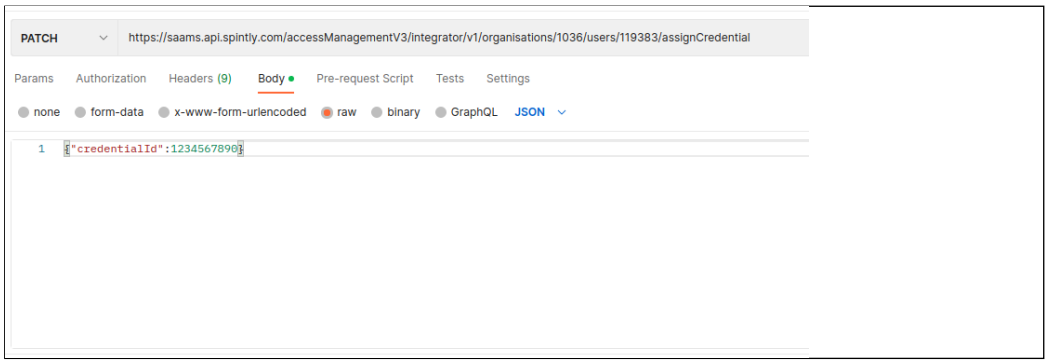
2) Update user permissions

This api is responsible for assigning and unassigning permissions of a user



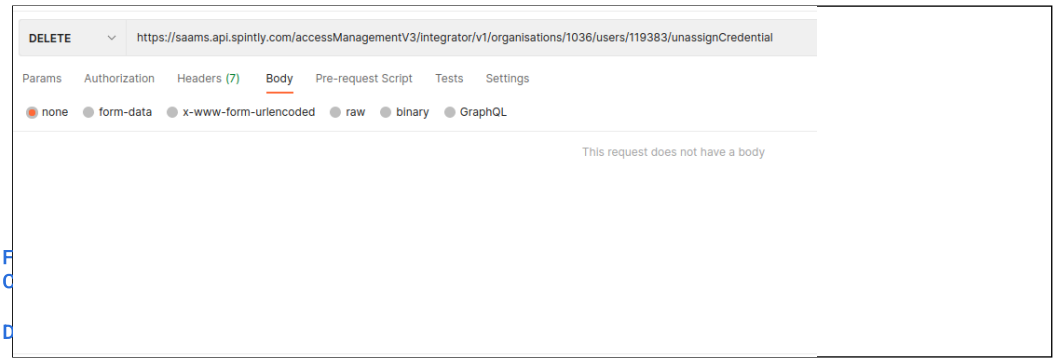
2) Card Assignment

This api is responsible for assigning a card to a user



2) Card Unassignment

This api is responsible for unassigning a card to a user



Related Articles

Device Management

dual associations secondary connections

Was this article helpful?

Yes No

Give feedback about this article

Spintly
Amplify the Simplicity

Industries

Multifamily / Residential
Co-Working
Education
Healthcare
Hospitality
Manufacturing
Fitness & Wellness

Solutions

Access Control
Visitor Management
Tenant Management
Time & Attendance Management

About

About Us
Partners
FAQ
Privacy Policy
Terms & Conditions

Resources

Case Studies
Blogs
Whitepapers
Webinars
Press
Data Sheets

Contact Us

Sales: sales@spintly.com
Tech Support: support@spintly.com
Get a Quote