

API Documentation

July 2017



Table of Contents

Overview 3

Authentication 3

Other REST API 5

# Overview

Coalesce.Info (“Coalesce”) has developed API to access the data through RESTful Services.

1. API to add company and its attributes.
2. API to get the company profile.
3. API to get the Alert information for specific company.
4. API to get a list of all the Alert definitions.
5. API to add training data to an Alert.
6. API to publish the Alert.

HTTP endpoints that can provide results in JSON media formats.

Endpoint : <https://v2-0.coalesce.info>

## Authentication

To access the API, a valid sessionId is necessary. To generate the sessionId one should have the valid username and password credentials which will be generated and shared to the user through the registered email by the Coalesce Admin.

Follow the steps to create the sessionId:

**URI :** /login.json

**Method :** GET

1. Concatenate the username and password with the separator ‘:’ and without space.
2. Then use a function to encode the string to base64 ASCII String.
3. Then Append ‘Basic <base64 String>‘ to the encoded string.
4. Then add a request header with the key ‘Authorization’ and the generated string in the previous step as the value.
5. Once it is successfully authenticated, the response body has the sessionId.
6. To access the rest of the API endpoints, set a request header as Cookie and then prepend the string ‘JSESSIONID=’ to the sessionId and set it as value.

**Response Entity:**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Description** |
| status | Boolean | Status of the login validation |
| message | String | Login validation message |
| sessionId | String | Session id(Authentication key) |
| **User Entity:** |  |  |
| id | Integer | User id |
| username | String | name of the currently logged in user. |
| displayName | String | Display name of the user |
| role | String | Role of the user(admin,manager) |
| isActive | Boolean | Flag mention the active user or not |
| lastLogin | Date | Last login date |
| orgCode | String | Organization code |
| organizationId | Integer | Organization id |
| groupIds | String | Group ids of the user |
| alertIds | List of Integers | List of alert ids which are associated (organization, group, user level) to the user. |

**Response Code : 200**

{

"responseDTO":

{

"status": true,

"message": "SESSION CREATED SUCCESSFULLY",

"sessionId": "kq15nrv2dpvigeh4khfge073",

"companyId": 0,

"user": {

"id": 1,

"username": "demo",

"displayName": "Demo",

"role": "Administrator",

"isActive": true,

"lastLogin": "2017-06-08",

"orgCode": "COALESCE-INC",

"organizationId": "1",

"groupIds": "1",

"modelIds": [ 15, 27 ]

}

}

**Response Code : 401**

{

"responseCode" : 401,

"responseMessage" : "Username/password is invalid.",

"dateTime" : "2017-06-08T13:26:39.723",

}

## Other REST API

For the remaining APIs we have developed the Swagger UI. You can use the URL : <https://v2-0.coalesce.info/swagger-ui.html>

For testing the UI, you need to provide your username / password credentials by clicking the Authorize button in the top right corner.

This will do Basic authentication for every REST API call. For the testing purpose it is absolutely fine to go with it.

But when you try using programmatically follow the authentication step as we mentioned above to avoid unnecessary basic authentication.