



CumuloLogic Core REST API Guide

June 2014

Table of Contents

| | |
|---|----|
| Core REST API | 1 |
| 1. Login and Logout | 2 |
| 1.1 Login User | 2 |
| 1.2 Logout User | 3 |
| 2. User | 4 |
| 2.1 Get user(s) by ID | 4 |
| 2.2 Create User | 5 |
| 2.3 Edit User | 6 |
| 2.4 Activate and deactivate user | 8 |
| 2.5 Get users by role | 9 |
| 2.6 Check username availability | 10 |
| 2.7 Update password | 12 |
| 2.8 Check assigned/unassigned users | 13 |
| 2.9 Sign up | 14 |
| 2.10 Get all users | 15 |
| 2.11 Update Myaccount | 16 |
| 3. Email template | 18 |
| 3.1 Get email template | 18 |
| 3.2 Get All Email Template(s) | 19 |
| 3.3 Edit email template | 20 |
| 4. Service type | 21 |
| 4.1 Get all service types | 21 |
| 5. Support settings | 23 |
| 5.1 Get all support settings | 23 |
| 5.2 Update Support Settings | 24 |
| 6. Target cloud | 25 |

| | | |
|------|---|----|
| 6.1 | Get target cloud(s) by user ID | 25 |
| 6.2 | Get target cloud details by target cloud ID | 26 |
| 6.3 | Get default target cloud..... | 27 |
| 6.4 | Add target cloud | 28 |
| 6.5 | Update target cloud | 31 |
| 6.6 | Delete Target Cloud | 33 |
| 6.7 | Update Target Cloud As Default..... | 34 |
| 6.8 | Assign To User | 35 |
| 7. | Virtual machine pool..... | 36 |
| 7.1 | Get Virtual Machine Pool(s)..... | 36 |
| 7.2 | Get virtual machine by ID | 38 |
| 7.3 | Delete virtual machine pool | 40 |
| 7.4 | Add virtual machine pool | 41 |
| 7.5 | Add virtual machine | 42 |
| 7.6 | Remove virtual machine..... | 44 |
| 8. | IaaS Provider..... | 45 |
| 8.1 | Get all available IaaS provider | 45 |
| 8.2 | Get IaaS provider by user | 46 |
| 9. | Global Settings..... | 47 |
| 9.1 | Get all global settings..... | 47 |
| 9.2 | Update global settings | 48 |
| 10. | Images | 49 |
| 10.1 | Get all image(s) | 49 |
| 10.2 | Get image by ID | 50 |
| 10.3 | Get image by IaaS provider ID | 51 |
| 10.4 | Get image by endpoint URI | 52 |
| 10.5 | Import image | 53 |
| 10.6 | Import and obsolete old image | 56 |
| 10.7 | Delete image | 58 |

| | |
|---|----|
| 10.8 Get all images by target cloud | 59 |
| 11. Instance Types..... | 60 |
| 11.1 Get all instance types..... | 60 |
| 11.2 Get all instance types by AZ and image ID | 61 |
| 11.3 Get all instance types by ID and image ID..... | 62 |
| 12. Newrelic | 64 |
| 12.1 Get Newrelic account..... | 64 |
| 12.2 Create Newrelic account | 65 |
| 12.3 Delete Newrelic account..... | 66 |
| 13. Service Access Group | 67 |
| 13.1 Get service access group..... | 67 |
| 13.2 Get all service access group by owner type & created by..... | 67 |
| 13.3 Get service access group by engine..... | 69 |
| 13.4 Create service access group | 70 |
| 13.5 Delete service access group..... | 72 |
| 13.6 Apply service access group | 73 |
| 13.7 Update service access group | 74 |
| 14. Availability zone | 76 |
| 14.1 Get availability zone by target cloud ID | 76 |
| 15. Appdynamic..... | 77 |
| 15.1 Get appdynamic account | 77 |
| 15.2 Create Appdynamic account..... | 78 |
| 15.3 Delete appdynamic account | 79 |
| 16. Access Group Template | 80 |
| 16.1 Get access group template by host type..... | 80 |

Core REST API

The API uses conventional REST request types for each command:

- List or view requests:** GET request type
- Add object request:** POST request type
- Edit object request:** POST or PUT request type
- Delete object request:** DELETE request type

NOTE: For every request need to add a CumuLogic auto generated authentication token in header ex. cl-auth-token:7e1cc53e-dc09-4712-bc5d-ea69bd63abc9. Except login and signup request.

There are two types of User Types/Categories (i.e. Premium and Admin) available when using CumuLogic Services. The users are requested to use their tokens as per their user type.

1. Login and Logout

1.1 Login User

| Http Method | URL | Description |
|-------------|---|--------------------------|
| POST | /core/api/v1/auth | Login to the application |

Request Parameters:

| Attribute | Type | Description | Required |
|-----------|--------|---------------------------------|----------|
| <action> | String | Action name "login" | Yes |
| Username | String | Login with username of the user | Yes |
| Password | String | Login with password of the user | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/auth" -X POST -d '{"login":{"username":"John@cumulogic.com","password":"cumulogic"}}'
```

Example Response:

```
{ "success": true, "errors": {}, "response": { "access": { "lastName": "Martin", "username": "John@cumulogic.com", "token": "46d0a966-2847-47aa-bfc8-e505baa44c53", "userId": "2", "accountType": "Premium", "firstName": "John" } } }
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

1.2 Logout User

| Http Method | URL | Description |
|-------------|-------------------|--------------------------------|
| POST | /core/api/v1/auth | Logs out user from application |

Request Parameters:

| Attribute | Type | Description | Required |
|-----------|--------|----------------------|----------|
| Token | String | Unique token of user | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/auth" -X POST -d '{"logout":{"token":"46d0a966-2847-47aa-bfc8-e505baa44c53"}}'
```

Example Response:

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500, ...), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

2. User

2.1 Get user(s) by ID

| Http Method | URL | Description |
|-------------|-------------------|---|
| GET | core/api/v1/users | To get specific user and users with different filters |

Query Parameters:

| Filter Name | Type | Description | Required |
|-------------|---------|---|----------|
| ID | Integer | ID of the existing user Put user ID as ID in URL | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/users/2" -X GET -H "CL-AUTH-TOKEN:4dab5e90-79c5-4a99-bacd-0bba05dc1529"
```

Example Response:

```
{"success":true,"errors":{},"response":{"users":[{"lastName":"Johnson","phone":"123456789","state":"California","companyName":"Cumulogic","city":"Albany","country":"United State","isActive":true,"id":2,"email":"John@rediffmail.com","zipCode":null,"accountType":"Premium","websiteUrl":"www.cumulogic.com","firstName":"John","createdDate":"2014-04-11 03:39:19","addrLineOne":"Albany","loginName":" John@rediffmail.com "}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

2.2 Create User

| Http Method | URL | Description |
|-------------|---|--------------------|
| POST | /core/api/v1/users | Creates a new user |

Request Parameters:

| Attribute | Type | Description | Required |
|-----------|--------|---------------|----------|
| Email | String | Email of user | Yes |

| | | | |
|------------------|---------|--|-----|
| First name | String | First name of the user | Yes |
| Last name | String | Last name of user | Yes |
| Phone | Integer | Contact number | Yes |
| City | String | City of user | No |
| Address line one | String | Address of user | No |
| Company name | String | Company name of user | No |
| Country | String | Country the user belongs to | No |
| State | String | State the user belongs to | No |
| Website URL | String | Website of user | No |
| Password | String | Password for the newly created account | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/users" -X POST -H "CL-AUTH-TOKEN:96493aca-9f87-4739-89aa-d722a5adea5b" -d '{"add":{"email": "david@hotmail.com","firstName": "David","lastName": "Thomas","phone": "0400343461253","city": "Los Angeles","addrLineOne": "Los Angeles","companyName": "Cumulogic","country": "United State","state": "California","websiteUrl": "www.cupertino.com","password": "cumulogic"}}'
```

Example Response:

```
{"success":true,"errors":{},"response":{"users":[{"id":14,"loginName":" david@hotmail.com"}]}}
```

Response Code(s):

| | |
|--------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413) itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |

| | |
|--------------------------------|--|
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |
|--------------------------------|--|

2.3 Edit User

| Http Method | URL | Description |
|-------------|-------------------------|--|
| PUT | /core/api/v1/users/<id> | Updates user information. Puts user ID as ID in URL |

Request Parameters:

| Attribute | Type | Description | Required |
|------------------|---------|--|----------|
| ID | Integer | Unique ID of user | Yes |
| First name | String | First name of the user | Yes |
| Last name | String | email ID of the user | Yes |
| Phone | String | Last name of user | Yes |
| City | String | Contact number | Yes |
| Address line one | String | City of user | Yes |
| Company name | String | Company name of user | Yes |
| Country | String | Country the user belongs to | Yes |
| State | String | State the user belongs to | Yes |
| Website URL | String | Website of user | No |
| Password | String | Password for the newly created account | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/users/14" -X PUT -H "CL-AUTH-TOKEN:4dab5e90-79c5-4a99-bacd-0bba05dc1529" -d{"firstName": "David", "lastName": "Thomas", "phone":
```

```
"04017361253", "city": "Los Angeles", "addrLineOne": "Los Angeles", "companyName":  
"Cumulogic", "country": "United State", "state": "California", "websiteUrl":  
"www.cumulogic.com", "password": "12345" }
```

Example Response: JSON

```
{"success":true,"errors":{},"response":{}}
```

Response Codes:

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityDoesNotExist (1012), invalidParameterValue(1013), invalidRequest(1000) |

2.4 Activate and deactivate user

| Http Method | URL | Description |
|-------------|---------------------|-------------------------------|
| POST | /core/api/v1/users/ | To activate the existing user |

Request Parameters:

| Attribute | Type | Description | Required |
|-----------|------|-------------|----------|
|-----------|------|-------------|----------|

| | | | |
|----------|---------|--|-----|
| <action> | String | Put action name as ' activate ' for activating and put action as ' deActive ' for de-activating user | Yes |
| ID | Integer | Unique ID of user | Yes |

Example Request:

For activating user:

```
$ curl -i "http://localhost:8080/core/api/v1/users" -X POST -H "CL-AUTH-TOKEN:6af416ab-33f4-4450-be14-779448ceb21e" -d '{"activate":{"id":"2"} }'
```

Example Response:

```
{"success":true,"errors":{},"response":{"users":[]}}
```

For de-activating user:

```
$ curl -i "http://localhost:8080/core/api/v1/users" -X POST -H "CL-AUTH-TOKEN:6af416ab-33f4-4450-be14-779448ceb21e" -d '{"deActivate":{"id":"2"} }'
```

Example Response:

```
{"success":true,"errors":{},"response":{"users":[]}}
```

Response Code(s):

| | |
|--------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |

| | |
|--------------------------------|---|
| Custom Error Response Code(s): | queryFailed(1009), invalidAction(1001), entityDoesNotExist(1012), invalidParameterValue(1013), invalidRequest(1000) |
|--------------------------------|---|

2.5 Get users by role

| Http Method | URL | Description |
|-------------|--------------------|------------------------------|
| GET | /core/api/v1/users | To get user of specific role |

Query Parameters:

| Filter Name | Type | Description |
|-------------|---------|---|
| Role ID | Integer | Put role ID as roleID in URL Possible roles of users ADMIN = 1 PREMIUM = 2 STANDARD = 3 PARTNER = 4 |

Example Request :

```
$ curl -i "http://localhost:8080/core/api/v1/users?roleId=2" -X GET -H "CL-AUTH-TOKEN:2f023e50-1712-4284-a0f7-40026748cd8a"
```

Example Response:

```
{"success":true,"errors":{},"response":{"users":[{"lastName":"Thomas","phone":"04423456789","state":"California","companyName":"Cumulogic","city":"Albany","country":"United State","isActive":true,"id":3,"email":"thomas@yahoo.com","zipCode":null,"accountType":"Premium","websiteUrl":"www.cumulogic.com","firstName":"George","createdDate":"2014-04-11 03:39:19","addrLineOne":"Albany","loginName":" thomas@yahoo.com"}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200,203 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

2.6 Check username availability

| Http Method | URL | Description |
|-------------|--------------------|------------------------------|
| GET | /core/api/v1/users | To get user of specific name |

Query Parameters:

| Filter Name | Type | Description |
|-------------|--------|---|
| Username | String | Username which is to be checked for availability Put username as name in URL |

Example Request :

```
$ curl -i "http://localhost:8080/core/api/v1/users?username= thomas@yahoo.com" -X GET  
-H "CL-AUTH-TOKEN:2f023e50-1712-4284-a0f7-40026748cd8a"
```

Example Response:

```
{"success":true,"errors":{},"response":{"users":[{"lastName":"Thomas","phone":"044234567  
89","state":"California","companyName":"Cumulogic","city":"Albany","country":"United
```

```
State","isActive":true,"id":3,"email":"thomas@yahoo.com","zipCode":null,"accountType":"Premium","websiteUrl":"www.cumulogic.com","firstName":"George","createdDate":"2014-04-11 03:39:19","addrLineOne":"Albany","loginName":" thomas@yahoo.com"]}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

2.7 Update password

| Http Method | URL | Description |
|-------------|--------------------|------------------------|
| POST | /core/api/v1/users | To update the password |

Request Parameters:

| Attribute | Type | Description | Required |
|-----------|---------|--|----------|
| ID | Integer | Put user ID as ID in URL | Yes |
| <action> | String | Action name for activating user: Put 'changePassword' as action name | Yes |
| Password | String | New password to be set | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/users" -X POST -H "CL-AUTH-TOKEN:b698429b-b2d8-4c73-b3cb-f8d70f822f12" -d '{"changePassword": {"id": "3", "password": "cumulogic"}}'
```

Example Response:

```
{"success":true,"errors":{},"response":{"users":[]}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

2.8 Check assigned/unassigned users

| Http Method | URL | Description |
|-------------|--------------------|--------------------------------------|
| GET | /core/api/v1/users | To get all assigned/unassigned users |

Query Parameters:

| Filter Name | Type | Description | Required |
|---------------|---------|---------------------------|----------|
| targetCloudID | Integer | Unique ID of target cloud | Yes |

| | | | |
|----------|---------|---|-----|
| Assigned | Integer | To get all assigned users for targetcloud assigned=1 assigned=2 | Yes |
|----------|---------|---|-----|

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/users?targetCloudId=26&assigned=1" -X GET -H "CL-AUTH-TOKEN:6a38369f-f7fa-4742-bf56-ff57d126633e"
```

Example Response:

```
{"success":true,"errors":{},"response":{"users":[{"lastName":"Thomas","phone":"04423456789","state":"California","companyName":"Cumulogic","city":"Davis","country":"United State","isActive":true,"id":3,"email":"thomas@yahoo.com","zipCode":null,"accountType":"Premium","websiteUrl":"www.cumulogic.com","firstName":"George","createdDate":"2014-04-11 03:39:19","addrLineOne":"davis","loginName":" thomas@yahoo.com"}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

2.9 Sign up

| Http Method | URL | Description |
|-------------|---|---------------------------------|
| POST | /core/api/v1/users | Sign up for creating a new user |

Request Parameters:

| Attribute | Type | Description | Required |
|-------------------|--------|--|----------|
| Email | String | Email of user | Yes |
| First name | String | First name of the user | Yes |
| Last name | String | Last name of user | Yes |
| Company name | String | Company name of user | No |
| Password | String | Password for the newly created account | Yes |
| Captcha response | String | Captcha response | Yes |
| Captcha challenge | String | Encoded captcha challenge | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/users" -X POST -d{"signup":{"firstName":"Paul","lastName":"Lee","email":"paul@yahoo.com","password":"12345","companyName":"cumulogic","captchaResponse":"12 3843248","captchaChallenge":"03AHJ_Vut3Vgw4I56H5i4zllrplotAwbXBeD-A9a-eDvPECzFD4bVUNPZJuZPyhBaUJb-O9ryGQnpRrP5cGWCNb95XhijCsFvGbjMyC8e4zfginyk5JQHKgtFt7uR0OYHb8yNJ2skAqOZ8CF3ECRFL_ZkB2dw_XlaT4O5xMe0Fre_jRh-giegameXSfNkG24hy2GgFCjD3-vv0hCeBow0LX9xwksBdDyPsBdw"}}
```

Example Response: JSON

```
{"success":true,"errors":{},"response":{"users":[{"id":103,"loginName":"archana@cumulogic.com"}]}}
```

Response Code(s):

| | |
|--------------------------|-----|
| Normal Response Code(s): | 200 |
|--------------------------|-----|

| | |
|--------------------------------|--|
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

2.10 Get all users

| Http Method | URL | Description |
|-------------|--------------------|------------------|
| GET | /core/api/v1/users | To get all users |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/users" -X GET -H "CL-AUTH-TOKEN:2f023e50-1712-4284-a0f7-40026748cd8a"
```

Example Response:

```
{
  "success": true,
  "errors": {},
  "response": {
    "users": [
      {
        "lastName": "Admin",
        "phone": "04423456789",
        "state": "California",
        "companyName": "Cumulogic",
        "city": "cupertino",
        "country": "United State",
        "isActive": true,
        "id": 1,
        "email": "admin@yahoo.com",
        "zipCode": null,
        "accountType": "Admin",
        "websiteUrl": "www.cumulogic.com",
        "firstName": "Admin",
        "createdDate": "2014-04-11 03:39:19",
        "addrLineOne": "cupertino",
        "loginName": "admin@yahoo.com"
      },
      {
        "lastName": "Thomas",
        "phone": "04423456789",
        "state": "California",
        "companyName": "Cumulogic",
        "city": "Albany",
        "country": "United State",
        "isActive": true,
        "id": 3,
        "email": "thomas@yahoo.com",
        "zipCode": null,
        "accountType": "Premium",
        "websiteUrl": "www.cumulogic.com",
        "firstName": "George",
        "createdDate": "2014-04-11 03:39:19",
        "addrLineOne": "Albany",
        "loginName": "thomas@yahoo.com"
      }
    ]
  }
}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

2.11 Update MyAccount

| Http Method | URL | Description |
|-------------|--------------------|--|
| POST | /core/api/v1/users | Updates the myaccount user information |

Request Parameters:

| Attribute | Type | Description | Required |
|------------------|---------|-----------------------------|----------|
| ID | Integer | Unique ID of user | Yes |
| First name | String | First name of the user | Yes |
| Last name | String | email ID of the user | Yes |
| Phone | String | Last name of user | Yes |
| City | Integer | Contact number | Yes |
| Address line one | String | City of user | Yes |
| Company name | String | Company name of user | Yes |
| Country | String | Country the user belongs to | Yes |
| State | String | State the user belongs to | Yes |
| Website URL | String | Website of user | No |

| | | | |
|----------|--------|--|-----|
| Password | String | Password for the newly created account | Yes |
|----------|--------|--|-----|

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/users" -X POST -H "CL-AUTH-TOKEN:a29db2d5-d55b-4ef4-9f2f-18cf4f191316" -d '{"updateMyAccount": {"id": 40,"firstName": "John","lastName": "Johnson","phone": "217361253","city": "San francisco","addrLineOne": "Albany","companyName": "Cumulogic","country": "United State","state": "California","websiteUrl": "www.cumulogic.com"}}'
```

Example Response:

```
{"success":true,"errors":{},"response":{"users":[]}}
```

Response Codes:

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

3. Email template

3.1 Get email template

| Http Method | URL | Description |
|-------------|------------------------------------|--|
| GET | /core/api/v1/emailtemplate?id=<id> | To get template format for a particular email template |

Query Parameters:

| Filter Name | Type | Description | Required |
|-------------|---------|--|----------|
| <ID> | Integer | Unique ID of the existing email template | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/emailtemplate/24" -X GET -H "CL-AUTH-TOKEN:b6ffeb0b-7fe9-4fca-9255-ed2bbc270c54"
```

Example Response:

```
{
  "success": true,
  "errors": {},
  "response": {
    "emailTemplates": [
      {
        "template": "Hello [FIRST_NAME],<br><br>Your CumuLogic Services account is now active.<br><br>Please [LOGIN_URL] to get started with CumuLogic Services.<br><br>Please refer to <a href='\"http://www.cumulogic.com/documentation/CumuLogic_User_Guide.pdf\">CumuLogic User Guide</a> to get started and feel free to contact support@cumulogic.com if you have questions or need assistance.<br><br>You can also stay in touch via <a href='\"http://twitter.com/cumulogic\">Twitter </a>and/or <a href='\"http://www.linkedin.com/company/cumulogic\">LinkedIn</a>. <br><br></div>Best regards,<br><br></div><div>CumuLogic Team</div>",
        "id": 24,
        "subject": "Your CumuLogic services account is now active",
        "serviceType": 14
      }
    ]
  }
}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

3.2 Get All Email Template(s)

| Http Method | URL | Description |
|-------------|---|---------------------------|
| GET | /core/api/v1/emailtemplate | To get all email template |

No Query Parameters are needed for this call

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/emailtemplate" -X GET -H "CL-AUTH-TOKEN:b6ffeb0b-7fe9-4fca-9255-ed2bbc270c54"
```

Example Response:

```
{"success":true,"errors":{},"response":{"emailTemplates":[{"template":"<br>Dear [USERNAME],<br><br>The database instance [SERVICE_NAME] has started successfully, however the monitoring configuration for the servers has failed. You can use the database service, but the monitoring features will not be available. We recommended you to re-provision your database service.<br><br> Best Regards, <br><br> CumuLogic Team","id":14,"subject":"Monitoring Configuration Failed","serviceType":1}],"template":"Hello [FIRST_NAME],<div><br>Thank you for registering for CumuLogic Services.<br><br>Please [ACTIVATION_URL] to activate your
```

account.</div><div>
Your feedback is very important to us. Please feel free to contact us at anytime for questions or comments at support@cumulogic.com.

</div><div>Best regards,

</div><div>CumuLogic Team</div>","id":15,"subject":"Cumulogic Platform Services account validation required","serviceType":2},{ "template":"Hello [FIRST_NAME],<div>
Your CumuLogic Services account is now active.

Please [LOGIN_URL] to get started with CumuLogic Services.

Please refer to CumuLogic User Guide to get started and feel free to contact support@cumulogic.com if you have questions or need assistance.

You can also stay in touch via Twitter and/or LinkedIn.

</div>Best regards,<div>
</div><div>CumuLogic Team</div>","id":24,"subject":"Your CumuLogic services account is now active","serviceType":14}}}

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

3.3 Edit email template

| Http Method | URL | Description |
|-------------|---|--------------------------------|
| PUT | <a href="/core/api/v1/emailtemplate/<id>">/core/api/v1/emailtemplate/<id> | Update existing email template |

Request Parameters:

| Attribute | Type | Description | Required |
|--------------|---------|--|----------|
| ID | Integer | Unique ID of the template to be edited | Yes |
| Subject | String | Subject of the email template | Yes |
| Template | String | Content of email template format | Yes |
| Service type | Integer | Template type | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/emailtemplate/23" -X PUT -H "CL-AUTH-TOKEN:b6ffeb0b-7fe9-4fca-9255-ed2bbc270c54" -d '{"template": "Dear [USERNAME],Host [HOST_NAME] of [SERVICE_TYPE] Instance [SERVICE_NAME] recovery service failed to start a new node. sorry Your Service Instance [SERVICE_NAME] will be unavailable. Please re-provision.Best Regards,CumuLogicTeam","subject": "Failed to recover sql instance1","serviceType": 13}'
```

Example Response:

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | QueryFailed(1009), entityDoesNotExist (1012), invalidParameterValue(1013), invalidRequest(1000) |

4. Service type

4.1 Get all service types

| Http Method | URL | Description |
|-------------|---|--------------------------|
| GET | /core/api/v1/servicetype | To get all service types |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/servicetype" -X GET -H "CL-AUTH-TOKEN:b6ffeb0b-7fe9-4fca-9255-ed2bbc270c54"
```

Example Response:

```
{"success":true,"errors":{},"response":{"servicetype":[{"id":1,"name":"Production"}, {"id":2,"name":"Development"}, {"id":3,"name":"Testing"}, {"id":4,"name":"Staging"}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200,203 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

5. Support settings

5.1 Get all support settings

| Http Method | URL | Description |
|-------------|------------------------------|-----------------------------|
| GET | /core/api/v1/supportsettings | To get all support settings |

There are no query parameters for this API call.

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/supportsettings" -X GET -H "CL-AUTH-TOKEN:b6ffeb0b-7fe9-4fca-9255-ed2bbc270c54"
```

Example Response:

```
{"success":true,"errors":{},"response":{"supportsettings":[{"id":1,"name":"ADMIN_EMAIL","value":"support@cumulogic.com"}, {"id":2,"name":"CORPORATE_ADDRESS","value":"Cumulo  
gic123"}, {"id":3,"name":"ADMIN_CONTACT_NO","value":"408-372-7676"}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

5.2 Update Support Settings

| Http Method | URL | Description |
|-------------|---|---|
| PUT | <a href="/core/api/v1/supportsettings/<id>">/core/api/v1/supportsettings/<id> | To update the existing support settings |

Request Parameters:

| Attribute | Type | Description | Required |
|-----------|--------|-----------------------------------|----------|
| Name | String | Name of the setting to be edited | Yes |
| Value | String | Value of the setting to be edited | Yes |

Example Request:-

```
$ curl -i "http://localhost:8080/core/api/v1/supportsettings/2" -X PUT -H "CL-AUTH-TOKEN:2f023e50-1712-4284-a0f7-40026748cd8a" -d '{"name":"CORPORATE_ADDRESS","value":"CumuLogic123"}'
```

Example Response:

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeout (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | QueryFailed(1009), entityDoesNotExist (1012), invalidParameterValue(1013), invalidRequest(1000) |

6. Target cloud

6.1 Get target cloud(s) by user ID:

| HTTP Method | URL | Description |
|-------------|--------------------------|--|
| GET | /core/api/v1/targetcloud | To get all target clouds for specific user |

Request Format: Query Parameters

| Filter Name | Type | Description | Required |
|-------------|---------|-------------------------|----------|
| User ID | Integer | ID of the existing user | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/targetcloud?userId=2" -X GET -H "CL-AUTH-TOKEN:56b0cc21-b171-4c3c-a155-63ec579473f0"
```

Response Format:

```
{"success":true,"errors":{},"response":{"targetClouds":[{"tenantId":null,"iaasProvider":{"iaasProviderName":"Amazon EC2","iaasProviderId":1,"accessKey":"4MV87724S7B","password":null,"id":46,"username":null,"accountNumber":"7867-2347-17","imageId":23,"secretKey":"sfhTKVKgv3t8YMqQ75tFYjvIKXoQgokcBc","createdBy":2,"isDefault":0,"name":"CL-Ec2","createdDate":"2014-05-10 02:50:39","endpointUri":"https://ec2.us-east-1.aws.com "}]}}
```

Response Code(s):

| | |
|-------------------------|---|
| Normal Response Code(s) | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |

| | |
|--------------------------------|---|
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |
|--------------------------------|---|

6.2 Get target cloud details by target cloud ID

| HTTP Method | URL | Description |
|-------------|----------------------------------|--------------------------------------|
| GET | /core/api/v1/targetcloud?id=<id> | To get specific target cloud details |

Request format: Query parameter

| Filter Name | Type | Description | Required |
|-------------|---------|--|----------|
| ID | Integer | Unique ID of the existing target cloud | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/targetcloud/46" -X GET -H "CL-AUTH-TOKEN:b6ffeb0b-7fe9-4fca-9255-ed2bbc270c54"
```

Response Format:

```
{"success":true,"errors":{},"response":{"targetClouds":[{"tenantId":null,"iaasProvider":{"iaasProviderName":"Amazon EC2","iaasProviderId":1,"accessKey":"4MV87724S7B","password":null,"id":46,"username":null,"accountNumber":"7867-2347-17","imageId":23,"secretKey":"sfhTKVKgv3t8YMqQ75tFYjvIKXoQgokcBc","createdBy":2,"isDefault":0,"name":"CL-Ec2","createdDate":"2014-05-10 02:50:39","endpointUri":"https://ec2.us-east-1.aws.com "}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s) | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

6.3 Get default target cloud

| HTTP Method | HTTP Method | Description |
|-------------|--------------------------|--|
| GET | /core/api/v1/targetcloud | To get details of default target cloud |

Request format: Query parameter

| Filter Name | Type | Description | Required |
|-------------|---------|----------------------------|----------|
| Default | Integer | Set value as 1 | Yes |
| User ID | Integer | Unique ID of existing user | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/targetcloud?default=1&userId=2" -X GET -H "CL-AUTH-TOKEN:56b0cc21-b171-4c3c-a155-63ec579473f0"
```

Example Response:

```
{"success":true,"errors":{},"response":{"targetClouds":[{"tenantId":null,"accessKey":"ffghfgdfgduRsSkCXwy6bjLytpsdgdfgdN-
```

```
Wlqffdrdgrdd","iaasProvider":{"iaasProviderName":"CloudStack","iaasProviderId":3},"password":null,"id":50,"username":null,"imageId":14,"secretKey":"dfrdgtDSEfdgfdgdgDdD","createdBy":2,"isDefault":1,"name":"CONTEGIX_TC","createdDate":"2014-05-13 03:39:19","endpointUri":"https://cloud.com/client/api"}}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s) | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413) , requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

6.4 Add target cloud

| HTTP Method | URL | Description |
|-------------|--------------------------|------------------------|
| POST | /core/api/v1/targetcloud | Add a new target cloud |

Request Format:Request Parameter:

Parameters [Specific to iaas-provider]:

IaaS Provider: HPCS

| Filter Name | Type | Description | Required |
|-------------|--------|------------------------|----------|
| accessKey | String | Access key of the iaas | Yes |
| secretKey | String | Secret key of the iaas | Yes |

| | | | |
|----------------|---------|-------------------------------|-----|
| Name | String | Name of target cloud | Yes |
| endpointUri | String | Endpoint uri of iaas provider | Yes |
| Tenanted | String | Tenant ID of iaas provider | Yes |
| iaasProviderID | Integer | iaaS provider ID | Yes |

iaaS Provider: AWS

| Filter Name | Type | Description | Required |
|----------------|---------|---------------------------------|----------|
| accessKey | String | Access key of the iaaS | Yes |
| secretKey | String | Secret key of the iaaS | Yes |
| Name | String | Name of target cloud | Yes |
| accountNumber | String | Account number of iaas provider | Yes |
| iaasProviderID | Integer | iaaS provider ID | Yes |

iaaS Provider: CloudStack

| Filter Name | Type | Description | Required |
|------------------|---------|---------------------------------|----------|
| accessKey | String | Access key of the iaaS | Yes |
| secretKey | String | Secret key of the iaaS | Yes |
| Name | String | Name of target cloud | Yes |
| endpointUri | String | Endpoint uri of iaas provider | Yes |
| AvailabilityZone | String | Availability zone provider name | Yes |
| iaasProviderID | Integer | iaaS provider ID | Yes |

IaaS Provider: OpenStack

| Filter Name | Type | Description | Required |
|----------------|---------|-------------------------------|----------|
| accessKey | String | Access key of the iaaS | Yes |
| secretKey | String | Secret key of the iaaS | Yes |
| Name | String | Name of target cloud | Yes |
| endpointUri | String | Endpoint uri of iaaS provider | Yes |
| iaasProviderID | Integer | iaaS provider ID | Yes |

IaaS Provider: VMPOOL

| Filter Name | Type | Description | Required |
|----------------|---------|---|----------|
| Name | String | Name of target cloud | Yes |
| ID | Integer | Unique ID of virtualmachinepool Ex. "virtualMachinePool": { "id": "1" } | Yes |
| iaasProviderID | Integer | iaaS provider ID | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/targetcloud" -X POST -H "CL-AUTH-TOKEN:56b0cc21-b171-4c3c-a155-63ec579473f0" -d'{"add": {"accessKey": "AIXM8rs0huRsSkCXwy6bjLytpYly126WN-WlqfR1EZSizsfdsgd","endpointUri": "https://cloud.cont.com/client/api","iaasProviderId": "3", "secretKey": "dfrdgtDSEFDsgfdgdsdDRHTFYSDGD","name": "target123","tenantId": ""}}
```

Example Response: JSON

```
{"success":true,"errors":{},"response":{"targetclouds":[{"id":55,"name":"target123"}]}}
```

Response Code(s):

| Response Type | Response Code |
|-------------------------------|--|
| Normal Response Code(s) | 200 |
| Error Response Code(s) | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s) | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

6.5 Update target cloud

| HTTP Method | URL | Description |
|-------------|-------------------------------|----------------------------------|
| PUT | /core/api/v1/targetcloud/<id> | Update the existing target cloud |

Request format: Request parameter

| Filter Name | Type | Description | Required |
|-------------|---------|--------------------------|----------|
| ID | Integer | Existing target cloud ID | Yes |
| accessKey | String | Access key of the iaaS | Yes |

| | | | |
|-----------|--------|------------------------|-----|
| secretKey | String | Secret key of the iaas | Yes |
|-----------|--------|------------------------|-----|

Parameters [specific to iaas-provider]:

iaaS provider: HPCS

| Filter Name | Type | Description | Required |
|----------------|---------|-------------------------------|----------|
| endpointUri | String | Endpoint uri of iaas provider | Yes |
| Tenanted | String | tenantId of iaas provider. | Yes |
| iaasProviderID | Integer | iaaS provider ID | Yes |

iaaS provider: AWS

| Filter Name | Type | Description | Required |
|----------------|---------|---------------------------------|----------|
| AccountNumber | String | Account number of iaas provider | Yes |
| iaasProviderID | Integer | iaaS provider ID | Yes |

iaaS provider: CloudStack

| Filter Name | Type | Description | Required |
|------------------|---------|-------------------------------|----------|
| Name | String | Name of target cloud | Yes |
| endpointUri | String | Endpoint uri of iaas provider | Yes |
| availabilityZone | String | Availabilit zone for provider | Yes |
| iaasProviderID | Integer | iaaS provider ID | Yes |

iaaS Provider:OpenStack

| Filter Name | Type | Description | Required |
|----------------|---------|-------------------------------|----------|
| endpointUri | String | Endpoint uri of iaas provider | Yes |
| iaasProviderID | Integer | iaaS provider ID | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/targetcloud/55" -X PUT -H "CL-AUTH-TOKEN:56b0cc21-b171-4c3c-a155-63ec579473f0" -d'{"accessKey": "AIXM8rs0huRsSkCXwy6bjLytvpYly126WN-WlqfR1EZSizsfdsgd","endpointUri": "https://cloud.cont.com/client/api ","iaasProviderId": "3", "secretKey": "dfrdgtDSEFDsgfdgdsdDRHTFYSDGD"}'
```

Example Response:

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| Response Type | Response Code |
|-------------------------------|--|
| Normal Response Code(s) | 200 |
| Error Response Code(s) | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s) | QueryFailed(1009), entityDoesNotExist (1012), invalidParameterValue(1013), invalidRequest(1000) |

6.6 Delete Target Cloud

| HTTP Method | URL | Description |
|-------------|-------------------------------|----------------------------------|
| DELETE | /core/api/v1/targetcloud/<id> | Delete the existing targetcloud. |

Request Format:Query Parameter

| Filter Name | Type | Description | Required |
|-------------|---------|---------------------------------|----------|
| ID | Integer | ID of the existing target cloud | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/targetcloud/48" -X DELETE -H "CL-AUTH-TOKEN:22290fd3-64c1-453f-a305-41476d97dc41"
```

Example Response:

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityDoesNotExist(1012), invalidRequest(1000) |

6.7 Update Target Cloud As Default

| HTTP Method | URL | Description |
|-------------|--------------------------|---|
| POST | /core/api/v1/targetcloud | Update the existing targetcloud as default. |

Request Format:Request Parameter

| Filter Name | Type | Description | Required |
|-------------|--------|----------------------------|----------|
| ID | String | Unique ID of target cloud | Yes |
| Created by | String | Unique ID of existing user | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/targetcloud" -X POST -H "CL-AUTH-TOKEN:1be9764f-b1fe-4d04-a496-3b5316b0c2b1" -d '{"updateAsdefault":{"id":"49","createdBy":"43"}}'
```

Example Response:

```
{"success":true,"errors":{},"response":{"targetclouds":[]}}
```

Response Code(s):

| Response Type | Response Code |
|-------------------------------|--|
| Normal Response Code(s) | 200 |
| Error Response Code(s) | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s) | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

6.8 Assign To User

| HTTP Method | URL | Description |
|-------------|--------------------------|-----------------------------|
| POST | /core/api/v1/targetcloud | Assign target cloud to user |

Request Format:Request Parameter

| Filter Name | Type | Description | Required |
|------------------|------------|--|----------|
| ID | String | Unique ID of target cloud | Yes |
| userTargetClouds | Array list | Ex. "userTargetClouds": [{ "user": { "id": "40" } }, { "user": { "id": "2" } }] | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/targetcloud" -X POST -H "CL-AUTH-TOKEN:df84f19b-b967-46c2-b077-97c4ea4ae55e" -d '{"assignToUser": {"id": "50"}, "userTargetClouds": [{"user": {"id": "40"}}, {"user": {"id": "2"}}}]'
```

Example Response:

```
{"success":true,"errors":{},"response":{"targetclouds":[]}}
```

Response Code(s):

| Response Type | Response Code |
|-------------------------|---------------|
| Normal Response Code(s) | 200 |

| | |
|-------------------------------|--|
| Error Response Code(s) | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s) | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

7 Entity: Virtual machine pool

7.1 Get Virtual Machine Pool(s)

| HTTP Method | URL | Description |
|-------------|---------------------------------|-------------------------------|
| GET | /core/api/v1/virtualmachinepool | To get all virtualmachinepool |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/virtualmachinepool" -X GET -H "CL-AUTH-TOKEN:86fbd8bb-3c8f-45c3-a471-88b531e41ec1"
```

Example Response:

```
{"success":true,"errors":{},"response":{"virtualmachinepool":[{"virtualmachine":{"createdBy":1,"status":1,"virtualMachineId":430,"privateDnsName":"domU-12-31-39-10-2C-A3.compute-1.internal","publicIpAddress":"54.81.99.238","hostName":"54.81.99.238","createdDate":"2014-06-19"}}

```

```

19:26:27", "privateIpAddress": "10.198.47.81"}, {"createdBy": 1, "status": 1, "virtualMachineId": 4
31, "privateDnsName": "domU-12-31-39-0F-DA-D2.compute-
1.internal", "publicIpAddress": "23.22.231.71", "hostName": "23.22.231.71", "createdDate": "201
4-06-19
19:26:27", "privateIpAddress": "10.193.217.28"}], "id": 34, "description": "gd_vmpool", "name": "g
d_vmpool", "keyPairValue": "-----BEGIN RSA PRIVATE KEY-----
\r\nMIIEpAlBAAKCAQEAIcmgmCQRwRjHogT32/KQOgiBdVtXH5tvDVsakHunNyLOS4zAugjb
AE+x6jB\r\nEct3ueP4TwOER1ZWHqodW/DhaRHtvlZMriTMD+3Rj2y4hBeO6xXoz/cTERJtoBN
nOO+XwHGCEcTK\r\n3l5lhmad0t9VcHPNey9FFAgVen093yAFom0y9EG2S120F1LI8WqIMJR8
sK6A7ildzNT+P96ujPO1\r\n3UVJHRHACIO8uPk19E5Zoib3qK7gqAWbbwSlnAwZeTEvXccZ/LX
phIN95oa3zvbPB3A4ISjWIKp7\r\nVMsm9B15ybujuR92jKxK++ZVuvQL5o/Ck3Ri2EKsfUWxG
sXBaYKwIDAQABAoIBADoPErNo525F\r\nlncF8dHBijFZ6iroPRLdbPLkw5HD+Na7vM8Y1CfSZz
KYZ7q4/NAhuyZCEBxDMBbb11iyd8KJAoJC\r\nngrokJRVv/ehsXIqgDLPkNV2LtkmlAsqh7Te2Cz
ekEQP0Fm8+XqvocG4NaJoZtkGgrv4VBQ3v80o\r\nPsz08lbZLEIsH2m/HHrFpemYRUuckGi4a6
Xgmtv45Bs9/cjXXG3a5y0xEKQYNwOr5zb6yUxemD2W\r\n0VQuWZOXH9uZgkTYXHdK8AYpe
XobxT3oabtCB593lu2TTMvz1WEheRfW++GPOcYkIxd5WmoFtL91\r\nNowWzFu1LsZo+ZHx2
GM4Kw8VC9ECgYEAvZDziMFb17vaWVGgyS/rOIX+zZooTTqs39kF/fegsUA\r\nFk+yj/zjfeqdt
MswtnKj5CTK1AU6owij519SrULRifMZe7vFQDtf/MLejLnkCFIhn66yPGiQxEL7\r\nHzmG+6s+D
uSyQwo2YDb1FIFWJeJOPdpltUBJPWDjwLEVncelVpkCgYEAuhNTIYHJ9WgkTS7dNtjt\r\nrd/9bc
UQDIfUAWQUo//s+3+ovi5A9W9mHizz3QV2S9bS5bfU96EeLt7XPKQAaCOyhRzjHUGVmitv\r\n
nLMOG32Sm6r8DwT09gLturw9zNrB95E0EettnZvPA1MN5UpxUaVBtU8pF6n3jm/cSFRTuYXIF
02MC\r\nngYBw715nZurUh/H5rHxlfApFcR1gaHXFNMVu0sJLFTzXI459vFVhOvS/s3rpLvSCfaHu
VY0Q61yX\r\nqLrDKSMUXe6ABPh1otRNBBKXf3Ee3dBguP9UJrvBCXrVsiHx5E/qt4ZtUQpyri7e
+17gQOzV/REo\r\nAGj/pqD7IDGbpUR971XmYQKBgQCpxQxTshEjeJr2WLW4WRF8vllq5ksZ6
wQhKMC0SMJdTEIXenht\r\n9e+DxJvSojMmSqAsuybxr+yr8uEKEM/WNAui65JDIRz7hnlurA6Y
L2sjbpfbsbQV6Olhp7Hy//xM9\r\nngw99UETtZ6Qd+yLsPnm9hIXbFwfnRGiwWilDXg+IING/5QK
BgQCob349hbD7aGGBiBkMAVHg8cYY\r\nnjy1UvdrDoPj0/LnvAYrtR3t3meA0VjX4R9I8ZeVHFs
ur2ib6/lphO9iLYZ5TRmuHo2JmZXZdzjO\r\n95azqBnULUZZ0UjACK5QIQR+qQDtNbDjiyyKhe
g9IYK/qha2Th5bMz3v9Dcjot71R/Umqw==\r\n-----END RSA PRIVATE KEY-----
\r\n", "createdDate": "2014-06-19 19:26:27"}]}}

```

Response Code(s):

| | |
|--------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |

| | |
|--------------------------------|---|
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |
|--------------------------------|---|

7.2 Get virtual machine by ID

| HTTP Method | URL | Description |
|-------------|---|---|
| GET | /core/api/v1/virtualmachinepool?id=<id> | To get specific virtualmachinepool details by virtualmachinepool ID |

| Filter Name | Type | Description | Required |
|-------------|---------|---------------------------------------|----------|
| ID | Integer | ID of the existing virtualmachinepool | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/virtualmachinepool/34" -X GET -H "CL-AUTH-TOKEN:86fbd8bb-3c8f-45c3-a471-88b531e41ec1"
```

Example Response:

```
{
  "success": true,
  "errors": {},
  "response": {
    "virtualmachinepool": [
      {
        "virtualmachine": [
          {
            "createdBy": 1,
            "status": 1,
            "virtualMachinId": 430,
            "privateDnsName": "domU-12-31-39-10-2C-A3.compute-1.internal",
            "publicIpAddress": "54.81.99.238",
            "hostName": "54.81.99.238",
            "createdDate": "2014-06-19 19:26:27",
            "privateIpAddress": "10.198.47.81"
          },
          {
            "createdBy": 1,
            "status": 1,
            "virtualMachinId": 432,
            "privateDnsName": "ip-10-190-91-44.ec2.internal",
            "publicIpAddress": "54.89.192.195",
            "hostName": "54.89.192.195",
            "createdDate": "2014-06-19 19:26:27",
            "privateIpAddress": "10.190.91.44"
          }
        ],
        "id": 34,
        "description": "gd_vmpool",
        "name": "gd_vmpool",
        "keyPairValue": "-----BEGIN RSA PRIVATE KEY-----\nMIIEpAIBAAKCAQEAicm gmCQRwRjHogT32/KQOgiBdVtXH5tvDV sakHunNyLOS n4zAugjbAE+x6jB\nEct3ueP4TwOER1ZWHqodW/DhaRHtvIZMriTMD+3Rj2y4hBeO6xXoz/cTERJtoBN"
      }
    ]
  }
}
```

```
nOO+XwHGCEcTK\r\n3i5lhmad0t9VcHPNey9FFAgVen093yAFom0y9EG2S120F1LI8WqIMJR8
sK6A7ildzNT+P96ujPO1\r\n3UVJHRHACIO8uPk19E5Zoib3qK7gqAWbbwSlnAwZeTEvXccZ/LX
phIN95oa3zvbPB3A4ISjWIKp7\r\nVMsm9B15ybujuR92jKxK++ZVuvQL5o/Ck3Ri2EKsfUWxG
sXBaYKwIDAQABAoIBADoPErNo525F\r\nlncF8dHBijFZ6iroPRLdbPLkw5HD+Na7vM8Y1CfSZz
KYZ7q4/NAhuyZCEBxDMBbb11iyd8KJAoJC\r\nngrokJRVv/ehsXIqgDLPkNV2LktkmlAsqh7Te2Cz
ekEQP0Fm8+XqvocG4NaJoZtkGgrv4VBQ3v80o\r\nPsz08lbZLEIsH2m/HHrFpemYRUuckGi4a6
Xgmtv45Bs9/cjXXG3a5y0xEKQYNwOr5zb6yUxemD2W\r\n0VQuWZOXH9uZgkTYXHdK8AYpe
XobxT3oabtCB593lu2TTMvz1WEheRfW++GPOcYkIxd5WmoFtL91\r\nNowWzFu1LsZo+ZHx2
GM4Kw8VC9ECgYEAzDziMFb17vaWVGgyS/rOIX+zZooTTqs39kF/fegsUA\r\nFk+yj/zjfeqdt
MswtnKj5CTK1AU6owlj519SrULRifMZe7vFQDtf/MLejLnkCFIhn66yPGiQxEL7\r\nHzmG+6s+D
uSyQwo2YDb1FIFWjeJOPdpltUBJPWDjwLEVncelVpkCgYEAuhNTIYHJ9WgkTS7dNtjt\r\nnrd/9bc
UQDIfUAWQUo//s+3+ovi5A9W9mHizz3QV2S9bS5bfU96EeLt7XPKQAaCOyhRzjHUGVmit\r\n
nLMOG32Sm6r8DwT09gLturw9zNrB95E0EettnZvPA1MN5UpxUaVBtU8pF6n3jm/cSFRTuYXIF
02MC\r\nngYBw715nZurUh/H5rHxlfApFcR1gaHXFNmVu0sJLFTzXI459vFVhOvS/s3rpLvSCfaHu
VY0Q61yX\r\nlqLrDKSMUXe6ABPh1otRNBBKXf3Ee3dBguP9UJrvBCXrVsiHx5E/qt4ZtUQpyri7e
+17gQOzV/REo\r\nAGj/pqD7IDGbpUR971XmYQKBgQCpxQxTshEjeJr2WLW4WRF8vlqq5ksZ6
wQhKMC0SMJdTEIXEnht\r\n9e+DxJvSojMmSqAsuybxr+yr8uEKEM/WNAui65JDIRz7hnlurA6Y
L2sjbpfbsbQV6Olhp7Hy//xM9\r\nngw99UETtZ6Qd+yLsPnm9hIXbFwfnRGiwWiiDXg+IING/5QK
BgQCob349hbD7aGGBiBkMAVHg8cYY\r\nnjy1UvdrDoPj0/LnvAYrtR3t3meA0VjX4R9I8ZeVHFs
ur2ib6/lphO9iLYZ5TRmuHo2JmZXZdzjO\r\n95azqBnULUZZ0UjACK5QIQR+qQDtNbDjIyyKhe
g9IYK/qha2Th5bMz3v9Dcjot71R/Umqw==\r\n-----END RSA PRIVATE KEY-----
\n","createdDate":"2014-06-19 19:26:27"}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeout (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

7.3 Delete virtual machine pool

| HTTP Method | URL | Description |
|-------------|--------------------------------------|--|
| DELETE | /core/api/v1/virtualmachinepool/<id> | Delete the existing virtualmachinepool |

Example Request:Request Parameter

| Filter Name | Type | Description | Required |
|-------------|---------|----------------------------------|----------|
| ID | Integer | Existing virtual machine pool ID | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/virtualmachinepool/34" -X DELETE -H "CL-AUTH-TOKEN:86fbd8bb-3c8f-45c3-a471-88b531e41ec1"
```

Example Response Code(s):

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityDoesNotExist(1012), invalidRequest(1000) |

7.4 Add virtual machine pool

| HTTP Method | URL | Description |
|-------------|---------------------------------|------------------------------|
| POST | /core/api/v1/virtualmachinepool | Add a new virtualmachinepool |

1] Request parameters: Request parameter

| Filter Name | Type | Description | Required |
|-------------|--------|-----------------------------------|----------|
| Name | String | Name of virtualmachinepool | Yes |
| Description | String | Description of virtualmachinepool | No |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/virtualmachinepool" -X POST -H "CL-AUTH-TOKEN:86fbd8bb-3c8f-45c3-a471-88b531e41ec1" -d{"create":{"description": "gd_vmpool","name": "gd_vmpool"}}
```

Note: After This do the PUT Request to upload the files

| HTTP Method | URL | Description |
|-------------|--|------------------------------------|
| PUT | /core/api/v1/virtualmachinepool/<id>/files | Update the existing created vmpool |

Request parameters: Request parameter

| Filter Name | Type | Description | Required |
|-------------|---------|--|----------|
| ID | Integer | Unique ID of the existing virtualmachinepool | Yes |

| | | | |
|----------------------------|--------|-----------------------------------|-----|
| keyPairFilePath | String | Key pair file path | Yes |
| virtualMachineJsonFilePath | String | Virtual machine Json file path | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/virtualmachinepool/36/files" -X PUT -H "CL-AUTH-TOKEN:86fbd8bb-3c8f-45c3-a471-88b531e41ec1" -F
"testtxt"=@"C:\\Users\\Archana\\Desktop\\CumuLogickey.pem" -F
"testtxt12"=@"C:\\Users\\Archana\\Desktop\\cumulogicvmpool.json"
```

Example Response:-

```
{"success":true,"errors":{},"response":{}}
```

7.5 Add virtual machine

| HTTP Method | URL | Description |
|-------------|---------------------------------|--|
| POST | /core/api/v1/virtualmachinepool | Add a new virtual machine to existing virtual machine pool |

Request parameter: Request parameter

| Filter Name | Type | Description | Required |
|----------------------|------------|---|----------|
| <action> | String | Action name "addMachines" | Yes |
| ID | String | Existing virtual machine pool ID | Yes |
| virtualMachineBOList | Array List | List of virtual machines desired to add | Yes |
| hostName | String | Host name Public IP address | Yes |
| privateDnsName | String | Private DNS name | Yes |
| privateIpAddress | String | Private IP address | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/virtualmachinepool" -X POST -H "CL-AUTH-TOKEN:86fbd8bb-3c8f-45c3-a471-88b531e41ec1" -d{"addMachines":{"id": "39","virtualMachineList": [{"hostName": "54.89.194.155","privateDnsName": "domU-12-31-39-03-2A-0A.compute-1.internal","privateIpAddress": "10.249.41.244"}]}}
```

Example Response Code(s):

```
{"success":true,"errors":{},"response":{"virtualmachinepool":[{"virtualmachine":{"createdBy":null,"status":1,"virtualMachineId":451,"privateDnsName":"domU-12-31-39-03-2A-0A.compute-1.internal","publicIpAddress":"54.89.194.155","hostName":"54.89.194.155","createdDate":"2014-06-19 20:10:15","privateIpAddress":"10.249.41.244"}],"id":39,"createdBy":1,"description":"gd_vmpool","name":"gd_vmpool","keyPairValue":null,"createdDate":"2014-06-19 20:10:15"}]}}
```

Response Code(s):

| | |
|--------------------------|-----|
| Normal Response Code(s): | 200 |
|--------------------------|-----|

| | |
|--------------------------------|--|
| Error Response Code(s): | computeFault (400, 500,...), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

7.6 Remove virtual machine

| HTTP Method | URL | Description |
|-------------|---------------------------------|-------------------------------------|
| POST | /core/api/v1/virtualmachinepool | Delete the existing virtual machine |

Request Parameter: Query Parameter

| Filter Name | Type | Description | Required |
|-------------|---------|---------------------------------------|----------|
| <action> | String | Action name removeMachines | Yes |
| ID | Integer | ID of the existing virtual machine ID | Yes |

Example Request:

```
curl -i "http://111.125.178.70:8080/core/api/v1/virtualmachinepool" -X POST -H "CL-AUTH-TOKEN:86fbd8bb-3c8f-45c3-a471-88b531e41ec1" -d' {"removeMachines": {"id": "439" }}'
```

Example Response:

```
{"success":true,"errors":{},"response":{"virtualmachinepool":[]}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

8. Entity: IaaS Provider

8.1 Get all available IaaS provider

| HTTP Method | URL | Description |
|-------------|---------------------------|-------------------------------|
| GET | /core/api/v1/iaasprovider | To get all iaas provider list |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/iaasprovider" -X GET -H "CL-AUTH-TOKEN:17c59a07-765a-43bf-b873-9250032a8d8a"
```

Example Response:

```
{"success":true,"errors":{},"response":{"iaasprovider":[{"id":1,"name":"Amazon EC2"}, {"id":2,"name":"AWS Compatible"}, {"id":3,"name":"CloudStack"}, {"id":4,"name":"VCloud"}, {"id":5,"name":"HPCS"}]}
```

```
{ "id":6,"name":"OpenStack"}, {"id":7,"name":"CloudStack_2.X"}, {"id":8,"name":"RackSpace"}, {"id":10,"name":"VM Pool"} ] ] }
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

8.2 Get IaaS provider by user

| Filter Name | Type | Description |
|-------------|---------|------------------|
| userID | Integer | Existing User ID |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/iaasprovider?userId=2" -X GET -H "CL-AUTH-TOKEN:17c59a07-765a-43bf-b873-9250032a8d8a"
```

Example Response:

```
{ "success":true,"errors":{},"response":{"iaasprovider":[{"id":1,"name":"Amazon EC2"}, {"id":2,"name":"AWS Compatible"}, {"id":3,"name":"CloudStack"}, {"id":5,"name":"HPCS"} ] ] }
```

Response Code(s):

| | |
|--------------------------|-----|
| Normal Response Code(s): | 200 |
|--------------------------|-----|

| | |
|--------------------------------|---|
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

9. Entity: Global Settings

9.1 Get all global settings

| HTTP Method | URL | Description |
|-------------|-----------------------------|----------------------------|
| GET | /core/api/v1/globalsettings | To get all global settings |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/globalsettings" -X GET -H "CL-AUTH-TOKEN:6af416ab-33f4-4450-be14-779448ceb21e"
```

Example Response:

```
{"success":true,"errors":{},"response":{"globalsettings":[{"id":32,"name":"CUMULOGIC_SERVER_URL","value":"15.185.155.167"},{"id":14,"name":"CUMULOGIC_PAASSTATIC_LOCATION","value":"/paas/PaaSStatic/"}, {"id":10,"name":"SMTP_HOST","value":"smtpout.secureserver.net"}, {"id":11,"name":"SMTP_USER","value":"support@cumulogic.com"}, {"id":12,"name":"SMTP_PASSWORD","value":"cumulogic1"}, {"id":13,"name":"SMTP_PAAS_URL","value":"http://15.185.185.167:8080"}, {"id":6,"name":"MYSQL_USERNAME","value":"root"}, {"id":7,"name":"MYSQL_PASSWORD","value":"cumulogic"}, {"id":8,"name":"MYSQL_SERVER","value":"127.0.0.1"}, {"id":15,"name":"HYPERIC_CLIENT_LIB","value":"/paas/hqapi1-client-5.0.0"}, {"id":25,"name":"HYPERIC_SERVER_PASSWORD","value":"hqadmin"}, {"id":26,"name":"
```

```

HYPERIC_SERVER_USERNAME","value":"hqadmin"},{"id":27,"name":"HYPERIC_SERVER_HOST_NAME","value":"166.78.62.13"},{"id":20,"name":"ADMIN_EMAIL","value":"support@cumulogic.com"},{"id":34,"name":"SESSION_TIME_OUT","value":"1440"},{"id":35,"name":"PAAS_SERVER","value":"0.0.0.0"},{"id":36,"name":"DEFAULT_SNAPSHOT_FREQUENCY","value":"1440"}, {"id":37,"name":"ACTIVE_DIRECTORY_URL","value":"LDAP://PSBI28DC04.cumulogic.com"}, {"id":38,"name":"CHEF_SERVER_URL","value":"https://api.cumulogic.com/organizations/site"}, {"id":39,"name":"CIDR_BITS","value":"/32"}, {"id":40,"name":"PAAS_SERVER_CIDR_IP","value":"0.0.0.0/0"}, {"id":41,"name":"LDAP_SERVER_URL","value":"LDAP://X.X.X.X"}, {"id":42,"name":"HYPERIC_SERVER_CIDR_IP","value":"0.0.0.0/0"}, {"id":43,"name":"DEFAULT_CIDR_IP","value":"0.0.0.0/0"}, {"id":44,"name":"PAAS_SERVER_CIDR_IP2","value":"0.0.0.0/0"}, {"id":45,"name":"ADMIN_NOTIFICATION_EMAIL","value":"support@cumulogic.com"}, {"id":46,"name":"DEFAULT_VM_TERMINATION_REMINDER","value":"7"}, {"id":47,"name":"JOB_HANDLE_URI","value":"http://localhost:8080/jobmanager/api/v1"}, {"id":48,"name":"REMOTE_INSTALLABLE_LOCATION","value":"/paas/PaaSStatic/installables"}, {"id":49,"name":"IS_LOCAL_INSTALLABLE","value":"true"}]]}

```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

9.2 Update global settings

| HTTP Method | URL | Description |
|-------------|----------------------------------|----------------------------------|
| PUT | /core/api/v1/globalsettings/<id> | Update the global settings value |

Request parameter: Request parameter

| Filter Name | Type | Description | Required |
|-------------|---------|----------------------------------|----------|
| ID | Integer | Unique setting ID | Yes |
| Name | String | Existing virtual machine pool ID | Yes |
| Value | String | Private IP address | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/globalsettings/12" -X PUT -H "CL-AUTH-TOKEN:6af416ab-33f4-4450-be14-779448ceb21e" -d{"name": "SMTP_PASSWORD","value": "cumulogic"}
```

Example Response Code(s):

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | QueryFailed(1009), entityDoesNotExist (1012), invalidParameterValue(1013), invalidRequest(1000) |

10. Entity: Images

10.1 Get all image(s)

| HTTP Method | URL | Description |
|-------------|---------------------|-------------------|
| GET | /core/api/v1/images | To get all images |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/images" -X GET -H "CL-AUTH-TOKEN:3f720e32-c54b-42a8-bf51-cb76c72e165c"
```

Example Response:

```
{ "success": true, "errors": {}, "response": { "image": { "osVersion": null, "isObsolete": 0, "number": "369086", "architecture": null, "type": 1, "instancetype": { "id": 30, "instanceTypeId": "101", "price": "1$", "name": "small", "instanceTypeValue": "small - small - 1$", "configuration": "small"}, "id": 13, "createdBy": 1, "description": "", "name": "falcon_image", "keypairName": "", "availabilityZone": "region-a.geo-1", "availabilityZoneName": "zone#2", "createdDate": "2014-04-25 04:40:14", "iaasprovider": { "id": 5, "name": "HPCS"}, "osName": null, "endpointUri": "https://region.identity.hp.com:35357/v2.0/" } } }
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

10.2 Get image by ID

| HTTP Method | URL | Description |
|-------------|-----------------------------|--------------------------|
| GET | /core/api/v1/images?id=<id> | To get image by image ID |

Request Parameter:Request Parameter

| Filter Name | Type | Description | Required |
|-------------|---------|-------------------|----------|
| ID | Integer | Existing image ID | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/images/19" -X GET -H "CL-AUTH-TOKEN:7cbc7d86-0113-40f1-8151-7a67115ad3b2"
```

Example Response:

```
{"success":true,"errors":{},"response":{"image":{"osVersion":null,"isObsolete":0,"number":"ami-3736045e","architecture":null,"type":2,"instancetype":{"id":41,"instanceTypeId":"t1.micro","price":"$2","name":"micro","instanceTypeValue":"micro - micro - $2","configuration":"micro"},"id":19,"createdBy":1,"description":"AmazonBase","name":"AmazonBase","keypairName":"","availabilityZone":"E-1","availabilityZoneName":"AmazonBase","createdDate":"2014-05-10 02:44:20","iaasprovider":{"id":1,"name":"Amazon EC2"},"osName":null,"endpointUri":"https://ec2.1.aws.com"}}}}
```

Response Code(s):

| | |
|--------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |

| | |
|--------------------------------|---|
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |
|--------------------------------|---|

10.3 Get image by IaaS provider ID

| HTTP Method | URL | Description |
|-------------|---------------------|----------------------------------|
| GET | /core/api/v1/images | To get image by IaaS provider ID |

Request Parameter: Request Parameter

| Filter Name | Type | Description | Required |
|----------------|---------|------------------|----------|
| iaasProviderID | Integer | IaaS provider ID | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/images?iaasProviderId=3" -X GET -H "CL-AUTH-TOKEN:2f023e50-1712-4284-a0f7-40026748cd8a"
```

Example Response:

```
{
  "success": true,
  "errors": {},
  "response": {
    "image": {
      "osVersion": null,
      "isObsolete": 1,
      "number": "e2645fca-3c35-48fa-9075-9ade8c90d642",
      "architecture": null,
      "type": 2,
      "instancetype": {
        "id": 5,
        "instanceTypeId": "193",
        "price": "$6",
        "name": "2xlarge",
        "instanceTypeValue": "2xlarge - 2xlarge - $6",
        "configuration": "2xlarge"
      },
      "id": 1,
      "createdBy": 1,
      "description": "image",
      "name": "Contegixl image",
      "keypairName": "",
      "availabilityZone": "1",
      "availabilityZoneName": "East-1",
      "createdDate": "2014-05-18 10:33:02",
      "iaasprovider": {
        "id": 3,
        "name": "CloudStack",
        "osName": null,
        "endpointUri": "https://cloud.com/client/api"
      }
    }
  }
}
```

Response Code(s):

| | |
|--------------------------|-----|
| Normal Response Code(s): | 200 |
|--------------------------|-----|

| | |
|--------------------------------|---|
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

10.4 Get image by endpoint URI

| HTTP Method | URL | Description |
|-------------|---------------------|-------------------------------|
| GET | /core/api/v1/images | To get image by end point uri |

Request parameter: Request parameter

| Filter Name | Type | Description | Required |
|-------------|--------|----------------------|----------|
| endpointUri | String | Endpoint uri of iaas | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/images?endpointUri=https://cloud.com/client/api"
-X GET -H "CL-AUTH-TOKEN:2f023e50-1712-4284-a0f7-40026748cd8a"
```

Example Response:

```
{"success":true,"errors":{},"response":{"image":{"osVersion":null,"isObsolete":0,"number":"e2645fca-3c35-48fa-9075-9ade8c90d642","architecture":null,"type":2,"instancetype":{"id":5,"instanceTypeId":"193","price":"$6","name":"2xlarge","instanceTypeValue":"2xlarge - 2xlarge - $6","configuration":"2xlarge"},"id":1,"createdBy":1,"description":"image","name":"Contegixl image","keypairName":"","availabilityZone":"1","availabilityZoneName":"East-1","createdDate":"2014-05-18"}}
```

10:33:02", "iaasprovider": {"id": 3, "name": "CloudStack"}, "osName": null, "endpointUri": "https://cloud.com/portal/api"}]}}

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

10.5 Import image

| HTTP Method | URL | Description |
|-------------|---------------------|-----------------|
| POST | /core/api/v1/images | Add a new image |

Common Request Parameters:Request Parameter

| Filter Name | Type | Description | Required |
|-------------------|--------|-------------------------------|----------|
| Name | String | Name of the image want to add | Yes |
| Description | String | Description of the image | No |
| osName | String | Name the operating system | No |
| osVersion | String | Version the operating system | No |
| imageProviderName | String | Name of the iaaS provider | No |
| Username | String | User name | No |
| Password | String | Password | No |

| | | | |
|----------------------|------------|--|-----|
| Architecture | String | Architecture of the OS | No |
| Type | Integer | Image type base image or master image | Yes |
| instanceTypeList | Array List | Array list of instance types | Yes |
| Name | String | Name of instance type | Yes |
| instanceTypeId | Integer | Instance type ID Ex for HPCS – 100,101,... etc. | Yes |
| Configuration | String | Configuration of instance type | Yes |
| Price | Integer | Price of instance | Yes |
| keypairName | String | Key pair name | No |
| availabilityZone | String | The availability of zone in which user wants to create the image | Yes |
| availabilityZoneName | String | Name of the availability zone which user wants to set | Yes |

Parameters[Specific to IaaS-provider]:

IaaS Provider: HPCS, Rackspace, CloudStack, vCloud, OpenStack

| Filter Name | Type | Description | Required |
|----------------------|---------|---|----------|
| endpointUri | String | End point URI | Yes |
| availabilityZone | String | The availability zone in which user wants to create the image | Yes |
| availabilityZoneName | String | Name of the availability zone which user wants to set | Yes |
| Number | Integer | Image number on IaaS | Yes |

**Parameters[Specific to IaaS-provider]:
IaaS Provider: AWS**

| Filter Name | Type | Description | Required |
|----------------------|---------|---|----------|
| availabilityZone | String | The availability zone in which user wants to create the image | Yes |
| availabilityZoneName | String | Name of the availability zone which user want to set | Yes |
| Number | Integer | Image number on IaaS | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/images" -X POST -H "CL-AUTH-TOKEN:c6e0df5f-6334-449b-a895-ff0d21c5a9c3" -d '{"import":{"name":"contegiximage","description":"test_image","availabilityZone":"1","availabilityZoneName":"STL01","organizationName":"","number":"e2645fca-3c35-48fa-9075-9ade8c90d642","endpointUri":"https://cloud.com/client/api","IaaSProvider":{"id":"3"},"IaaSProviderName":"cloudstack","architecture":"x86_64","type":"1","keypairName":"","instanceTypeList":[{"name":"xsmall","instanceTypeId":"179","configuration":"xsmall","price":"10$"}, {"name":"small","instanceTypeId":"183","configuration":"small","price":"15"}]}'
```

Example Response:

```
{ "success": true, "errors": {}, "response": { "image": { "id": 5, "name": "contegiximage", "availabilityZone": "1", "IaaSProvider": { "id": 3, "name": "CloudStack" }, "endpointUri": "https://cloud.com/client/api" } } }
```

Response Code(s):

| | |
|--------------------------|-----|
| Normal Response Code(s): | 201 |
|--------------------------|-----|

| | |
|--------------------------------|--|
| Error Response Code(s): | computeFault (400, 500,...), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

10.6 Import and obsolete old image

| HTTP Method | URL | Description |
|-------------|---------------------|------------------|
| POST | /core/api/v1/images | Add a new Image. |

Common request parameters: Request parameter

| Filter Name | Type | Description | Required |
|-------------------|--------|-------------------------------|----------|
| Name | String | Name of the image want to add | Yes |
| Description | String | Description of the image | No |
| osName | String | Name the operating system | No |
| osVersion | String | Version the operating system | No |
| imageProviderName | String | Name of the iaaS provider | No |
| Username | String | User name | No |
| Password | String | Password | No |
| Architecture | String | Architecture of the OS | No |

| | | | |
|------------------|------------|--|-----|
| Type | Integer | Image type base image or master image | Yes |
| instanceTypeList | Array List | Array list of instance types | Yes |
| Name | String | Name of instance type | Yes |
| instanceTypeId | Integer | Instance type ID Ex for HPCS – 100,101,... etc. | Yes |
| Configuration | String | Configuration of instance type | Yes |
| Price | Integer | Price of instance | Yes |
| keypairName | String | Key pair name | No |

Parameters[Specific to IaaS-provider]:

IaaS Provider: HPCS, Rackspace, CloudStack, vCloud, OpenStack

| Filter Name | Type | Description | Required |
|----------------------|---------|---|----------|
| endpointUri | String | End point URI | Yes |
| availabilityZone | String | The availability zone in which user wants to create the image | Yes |
| availabilityZoneName | String | Name of the availability zone which user want to set | Yes |
| Number | Integer | Image number on IaaS | Yes |

Example Request: JSON

```
curl -i "http://localhost:8080/core/api/v1/images" -X POST -H "CL-AUTH-TOKEN:2f023e50-1712-4284-a0f7-40026748cd8a" -d '{"importAndObsoleteOldImage":{"id": "1","name": "contegix","description": "test_image","availabilityZone": "1", "availabilityZoneName": "zone1", "organizationName": "", "number": "10f012fgf3434sdfdg80", "endpointUri": "https://cloud.com/client/api", "IaaSProvider":{"id": "3"},"IaaSProviderName": "contegix","architecture": "x86_64", "type": "2","keypairName": "", "instanceTypeList": [ { "name": "xsmall","instanceTypeId": "100","configuration": "xsmall","price": "10" }, { "name": "small", "instanceTypeId": "101", "configuration": "small","price": "15" }, { "name": "medium",
```

```
"instanceTypeId": "102", "configuration": "medium", "price": "20"}, {"name": "large",
"instanceTypeId": "103", "configuration": "large", "price": "25" } ] } }
```

Example Response:

```
{"success":true,"errors":{},"response":{"image":{"id":15,"name":"contegix","availabilityZone":
":1","iaasprovider":{"id":3,"name":null},"endpointUri":"https://cloud.com/client/api"}}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

10.7 Delete image

| HTTP Method | URL | Description |
|-------------|--------------------------|---------------------------|
| DELETE | /core/api/v1/images/<id> | Delete the existing image |

Request format: Query parameter

| Filter Name | Type | Description | Required |
|-------------|---------|--------------------------|----------|
| ID | Integer | ID of the existing image | Yes |

Example Request:

```
$ curl -i "http://166.78.143.203:8080/core/api/v1/images/12" -X DELETE -H "CL-AUTH-TOKEN:2f023e50-1712-4284-a0f7-40026748cd8a"
```

Example Response:

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityDoesNotExist(1012), invalidRequest(1000) |

10.8 Get all images by target cloud

| HTTP Method | URL | Description |
|-------------|---------------------|---------------------------------|
| GET | /core/api/v1/images | To get image by target cloud ID |

Request parameter: Request parameter

| Filter Name | Type | Description | Required |
|---------------|---------|---------------------------------|----------|
| targetCloudID | Integer | ID of the existing target cloud | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/images?targetCloudId=5" -X GET -H "CL-AUTH-TOKEN:a6f90068-1790-47c1-9cba-49ef6f75d7c4"
```

Example Response:

```
{"success":true,"errors":{},"response":{"image":{"osVersion":null,"isObsolete":0,"number":"376931","architecture":"x86_64","type":1,"instancetype":{"id":59,"instanceTypeId":"103","price":"25","name":"large","instanceTypeValue":"large - large - 25","configuration":"large"},"id":16,"createdBy":2,"description":"test_image","name":"Test image","keypairName":"","availabilityZone":"az-a.geo-1","availabilityZoneName":"zone1","createdDate":"2014-03-10 00:18:23","iaasprovider":{"id":5,"name":"HPCS"},"osName":null,"endpointUri":"https://region-a.geo-1.cloudsvc.com:65/v2.0/"}}}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

11. Instance Types

11.1 Get all instance types:

| HTTP Method | URL | Description |
|-------------|---------------------------|---------------------------|
| GET | /core/api/v1/instancetype | To get all instance types |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/instancetype" -X GET -H "CL-AUTH-TOKEN:a6f90068-1790-47c1-9cba-49ef6f75d7c4"
```

Example Response:

```
{"success":true,"errors":{},"response":{"instancetype":[{"id":56,"instanceTypeId":"100","imageId":16,"price":"10","name":"xsmall","availabilityZone":"az-1-a.geo-1","instanceTypeValue":"xsmall - xsmall - 10","configuration":"xsmall"}, {"id":57,"instanceTypeId":"101","imageId":16,"price":"15","name":"small","availabilityZone":"az-1-a.geo-1","instanceTypeValue":"small - small - 15","configuration":"small"}, {"id":58,"instanceTypeId":"102","imageId":16,"price":"20","name":"medium","availabilityZone":"az-1-a.geo-1","instanceTypeValue":"medium - medium - 20","configuration":"medium"}, {"id":59,"instanceTypeId":"103","imageId":16,"price":"25","name":"large","availabilityZone":"az-1-a.geo-1","instanceTypeValue":"large - large - 25","configuration":"large"}, {"id":60,"instanceTypeId":"100","imageId":17,"price":"10","name":"xsmall","availabilityZone":"az-2-a.geo-1","instanceTypeValue":"xsmall - xsmall - 10","configuration":"xsmall"}, {"id":61,"instanceTypeId":"101","imageId":17,"price":"15","name":"small","availabilityZone":"az-2-a.geo-1","instanceTypeValue":"small - small - 15","configuration":"small"}, {"id":62,"instanceTypeId":"102","imageId":17,"price":"20","name":"medium","availabilityZone":"az-2-a.geo-1","instanceTypeValue":"medium - medium - 20","configuration":"medium"}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

11.2 Get all instance types by AZ and image ID

| HTTP Method | URL | Description |
|-------------|---------------------------|--|
| GET | /core/api/v1/instancetype | To get all instance types by availability zone |

Request Format:Query Parameter

| Filter Name | Type | Description | Required |
|------------------|---------|---|----------|
| imageID | Integer | Unique ID of available image | Yes |
| availabilityZone | String | Availability zone of the existing image | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/instancetype?imageID=16&availabilityZone=az-1-a.geo-1" -X GET -H "CL-AUTH-TOKEN:a6f90068-1790-47c1-9cba-49ef6f75d7c4"
```

Example Response:

```
{"success":true,"errors":{},"response":{"instancetype":[{"id":56,"instanceTypeID":"100","imageID":16,"price":10,"name":"xsmall","availabilityZone":"az-1-a.geo-1","instanceTypeValue":"xsmall - xsmall - 10","configuration":"xsmall"}, {"id":57,"instanceTypeID":"101","imageID":16,"price":15,"name":"small","availabilityZone":"az-1-a.geo-1","instanceTypeValue":"small - small - 15","configuration":"small"}, {"id":58,"instanceTypeID":"102","imageID":16,"price":20,"name":"medium","availabilityZone":"az-1-a.geo-1","instanceTypeValue":"medium - medium - 20","configuration":"medium"}, {"id":59,"instanceTypeID":"103","imageID":16,"price":25,"name":"large","availabilityZone":"az-1-a.geo-1","instanceTypeValue":"large - large - 25","configuration":"large"}]}}
```

Response Code(s):

| | |
|--------------------------|-----|
| Normal Response Code(s): | 200 |
|--------------------------|-----|

| | |
|--------------------------------|---|
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

11.3 Get all instance types by ID and image ID

| HTTP Method | URL | Description |
|-------------|---------------------------|--|
| GET | /core/api/v1/instancetype | To get all Instance types by availability zone |

Request Format:Query Parameter

| Filter Name | Type | Description | Required |
|-------------|---------|--------------------------------|----------|
| imageID | Integer | Unique ID of available image | Yes |
| ID | Integer | Unique ID of the instance type | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/instancetype?imageid=61&instanceTypeId=186" -X GET -H "CL-AUTH-TOKEN:a6f90068-1790-47c1-9cba-49ef6f75d7c4"
```

Example Response:

```
{"success":true,"errors":{},"response":{"instancetype":[{"id":186,"instanceTypeId":"103","imageid":61,"price":"25","name":"large","availabilityZone":"az-2-a.geo-1","instanceTypeValue":"large - large - 25","configuration":"large"}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

12. Newrelic

12.1 Get Newrelic account

| HTTP Method | URL | Description |
|-------------|-----------------------|-------------------------|
| GET | /core/api/v1/newrelic | To get newrelic account |

Request format: Query parameter

| Filter Name | Type | Description | Required |
|-------------|---------|------------------|----------|
| userID | Integer | Existing User ID | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/newrelicaccount?userId=2" -X GET -H "CL-AUTH-TOKEN:6b611801-88be-412b-8041-0e85cc87aca1"
```

Example Response:

```
{"success":true,"errors":{},"response":{"newrelicaccount":[{"id":4,"createdBy":2,"licenseKey":
"asfs4dsf2f5aa2b2245f549sadsdf232s2sd256","createdDate":"2014-05-12 02:50:45"}]}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

12.2 Create Newrelic account

| HTTP Method | URL | Description |
|-------------|------------------------------|----------------------------|
| POST | /core/api/v1/newrelicaccount | Add a new newrelic account |

Common Request Parameters:

| Filter Name | Type | Description | Required |
|-------------|--------|---------------------------------|----------|
| licenseKey | String | Newrelic account license key | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/newrelicaccount" -X POST -H "CL-AUTH-
TOKEN:6b611801-88be-412b-8041-0e85cc87aca1" -d '{"create":{"licenseKey": "
asfs4dsf2f5aa2b2245f549sadsdf232s2sd256"}'}
```

Example Response:

```
{ "success": true, "errors": {}, "response": { "newrelicaccount": [ { "id": 4, "createdBy": 2, "licenseKey": " asfs4dsf2f5aa2b2245f549sadsdf232s2sd256", "createdDate": "2014-05-12 02:50:45" } ] } }
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

12.3 Delete Newrelic account

| HTTP Method | URL | Description |
|-------------|-----------------------------------|--------------------------------------|
| DELETE | /core/api/v1/newrelicaccount/<id> | Delete the existing newrelic account |

Request format: Query parameter

| Filter Name | Type | Description | Required |
|-------------|---------|-------------------------------------|----------|
| ID | Integer | ID of the existing newrelic account | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/newrelicaccount/2" -X DELETE -H "CL-AUTH-
```

TOKEN:6b611801-88be-412b-8041-0e85cc87aca1"

Example Response:

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityDoesNotExist(1012), invalidRequest(1000) |

13. Service Access Group

13.1 Get service access group

| HTTP Method | URL | Description |
|-------------|---------------------------------|-----------------------------|
| GET | /core/api/v1/serviceaccessgroup | To get service access group |

Request format: Query parameter

| Filter Name | Type | Description | Required |
|-------------|---------|--------------------------|----------|
| ID | Integer | Existing access group ID | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/serviceaccessgroup?id=4256" -X GET -H "CL-AUTH-TOKEN:6b611801-88be-412b-8041-0e85cc87aca1"
```

Example Response:

```
{"success":true,"errors":{},"response":{"serviceaccessgroup":[{"id":4256,"serviceaccessgroupproperties":[{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":2144,"toPort":2144,"serviceAccessGroupPropertiesId":17728},{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":22,"toPort":22,"serviceAccessGroupPropertiesId":17729},{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":27019,"toPort":27019,"serviceAccessGroupPropertiesId":17730},{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":27017,"toPort":27017,"serviceAccessGroupPropertiesId":17731}],"createdBy":2,"description":"MongoDB-Management","name":"default_MongoDB_775","serviceEngineName":"MongoDB 2.2.0"}]}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

13.2 Get all service access group by owner type & created by

| HTTP Method | URL | Description |
|-------------|---------------------------------|---------------------------------|
| GET | /core/api/v1/serviceaccessgroup | To get all service access group |

Request Format:Query Parameter

| Filter Name | Type | Description | Required |
|-------------|---------|------------------|----------|
| userID | Integer | Existing user ID | Yes |
| ownerType | Integer | Service type | Yes |

Example Request:

```
$ curl -i  
"http://15.185.185.167:8080/core/api/v1/serviceaccessgroup?ownerType=2&userId=2" -X  
GET -H "CL-AUTH-TOKEN:89fec161-b687-4e4f-b31b-027cbb611573"
```

Example Response:

```
{"success":true,"errors":{},"response":{"serviceaccessgroup":[{"id":4260,"serviceaccessgroup  
pproperties":[{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":2144,"toPort":2144,"serviceAc  
cessGroupPropertiesId":17744},{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":22,"toPort":  
22,"serviceAccessGroupPropertiesId":17745},{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort  
":27019,"toPort":27019,"serviceAccessGroupPropertiesId":17746},{"cidrIp":"0.0.0.0/0","prot  
ocol":"tcp","fromPort":27017,"toPort":27017,"serviceAccessGroupPropertiesId":17747}],"  
createdBy":2,"description":"MongoDB-  
Management","name":"default_MongoDB_138","serviceEngineName":"MongoDB  
2.2.0"},{"id":4261,"serviceaccessgroupproperties":[{"cidrIp":"0.0.0.0/0","protocol":"tcp","fro  
mPort":2144,"toPort":2144,"serviceAccessGroupPropertiesId":17748},{"cidrIp":"0.0.0.0/0","p  
rotocol":"tcp","fromPort":22,"toPort":22,"serviceAccessGroupPropertiesId":17749},{"cidrIp":  
"0.0.0.0/0","protocol":"tcp","fromPort":27019,"toPort":27019,"serviceAccessGroupPropertie  
sId":17750},{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":27017,"toPort":27017,"serviceAc  
cessGroupPropertiesId":17751}],"  
createdBy":2,"description":"MongoDB-  
Management","name":"default_MongoDB_996","serviceEngineName":"MongoDB 2.2.0"}]}
```

Response Code(s):

| | |
|--------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |

| | |
|--------------------------------|---|
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |
|--------------------------------|---|

13.3 Get service access group by engine

| HTTP Method | URL | Description |
|-------------|---------------------------------|---------------------------------------|
| GET | /core/api/v1/serviceaccessgroup | To get service access group by engine |

Request Format:Query Parameter

| Filter Name | Type | Description | Required |
|-------------|---------|---------------------|----------|
| hostType | Integer | Service engine type | Yes |
| userID | Integer | User ID | Yes |

Example Request:

```
curl -i "http://localhost:8080/core/api/v1/serviceaccessgroup?hostType=2&userId=2" -X GET
-H "CL-AUTH-TOKEN:6b611801-88be-412b-8041-0e85cc87aca1"
```

Example Response: JSON

```
{
  "success": true,
  "errors": {},
  "response": {
    "serviceaccessgroup": [
      {
        "id": 4256,
        "serviceaccessgroupproperties": [
          {
            "cidrIp": "0.0.0.0/0",
            "protocol": "tcp",
            "fromPort": 2144,
            "toPort": 2144,
            "serviceAccessGroupPropertiesId": 17728
          },
          {
            "cidrIp": "0.0.0.0/0",
            "protocol": "tcp",
            "fromPort": 22,
            "toPort": 22,
            "serviceAccessGroupPropertiesId": 17729
          },
          {
            "cidrIp": "0.0.0.0/0",
            "protocol": "tcp",
            "fromPort": 27019,
            "toPort": 27019,
            "serviceAccessGroupPropertiesId": 17730
          },
          {
            "cidrIp": "0.0.0.0/0",
            "protocol": "tcp",
            "fromPort": 27017,
            "toPort": 27017,
            "serviceAccessGroupPropertiesId": 17731
          }
        ],
        "createdBy": 2,
        "description": "MongoDB-Management",
        "name": "default_MongoDB_775",
        "serviceEngineName": "MongoDB 2.2.0"
      },
      {
        "id": 4257,
        "serviceaccessgroupproperties": [
          {
            "cidrIp": "0.0.0.0/0",
            "protocol": "tcp",
            "fromPort": 2144,
            "toPort": 2144,
            "serviceAccessGroupPropertiesId": 17732
          }
        ]
      }
    ]
  }
}
```

```
rotocol":"tcp","fromPort":22,"toPort":22,"serviceAccessGroupPropertiesId":17733},{ "cidrlp":
"0.0.0.0/0","protocol":"tcp","fromPort":27019,"toPort":27019,"serviceAccessGroupPropertie
sId":17734},{ "cidrlp":"0.0.0.0/0","protocol":"tcp","fromPort":27017,"toPort":27017,"serviceAc
cessGroupPropertiesId":17735}], "createdBy":2,"description":"MongoDB-
Management","name":"default_MongoDB_899","serviceEngineName":"MongoDB 2.2.0"]}]}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

13.4 Create service access group

| HTTP Method | URL | Description |
|-------------|---------------------------------|--------------------------------|
| POST | /core/api/v1/serviceaccessgroup | Add a new service access group |

Request parameters: Request parameter

| Filter Name | Type | Description | Required |
|-------------|--------|--------------------------|----------|
| Name | String | Access group name | Yes |
| Description | String | Access group description | No |

| | | | |
|------------------------------------|------------|--|-----|
| hostType | Integer | Service engine type ID | Yes |
| serviceAccessGroupPropertiesBOList | Array List | Array list of access group properties | Yes |
| Protocol | String | Which protocol - tcp or udp | Yes |
| fromPort | Integer | Is the start range of port | Yes |
| toPort | Integer | Up to range of port | Yes |
| cidrlp | String | For allocating IP addresses and routing Internet Protocol packets it is 0.0.0.0/0 or any other | Yes |
| ownerType | Integer | Service type | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/serviceaccessgroup" -X POST -H "CL-AUTH-TOKEN:6b611801-88be-412b-8041-0e85cc87aca1" -d '{"create":{"name":"test","description":"createdBy","ownerType":11,"hostType":"3","serviceAccessGroupPropertiesBOList":[{"protocol":"tcp","fromPort":"8080","toPort":"8080","cidrlp":"0.0.0.0/0","isDefault":"0"}, {"protocol":"tcp","fromPort":"2144","toPort":"2144","cidrlp":"0.0.0.0/0","isDefault":"0"}, {"protocol":"tcp","fromPort":"22","toPort":"22","cidrlp":"0.0.0.0/0","isDefault":"0"}],"serviceEngineName":"Jboss-7.0.2"}'}
```

Example Response:

```
{"success":true,"errors":{},"response":{"serviceaccessgroup":[{"id":4259,"serviceaccessgroupproperties":true,"createdBy":2,"description":"createdBy","name":"test","serviceEngineName":"Jboss-7.0.2"}]}}
```

Response Code(s):

| | |
|--------------------------|-----|
| Normal Response Code(s): | 201 |
|--------------------------|-----|

| | |
|--------------------------------|--|
| Error Response Code(s): | computeFault (400, 500,...), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

13.5 Delete service access group

| HTTP Method | URL | Description |
|-------------|--------------------------------------|----------------------------------|
| DELETE | /core/api/v1/serviceaccessgroup/<id> | Delete the existing access group |

Request Format:Query Parameter

| Filter Name | Type | Description | Required |
|-------------|---------|------------------------------------|----------|
| ID | Integer | ID of the existing access group ID | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/serviceaccessgroup/4256" -X DELETE -H "CL-AUTH-TOKEN:4c346e43-1f1f-4037-8089-f2dad4fe7bc3"
```

Example Response:

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), invalidParameterValue(1013), invalidRequest(1000) |

13.6 Apply service access group

| HTTP Method | URL | Description |
|-------------|---------------------------------|--------------------|
| POST | /core/api/v1/serviceaccessgroup | Apply access group |

Request format: Query parameter

| Filter Name | Type | Description | Required |
|-------------|---------|------------------------------------|----------|
| ID | Integer | ID of the existing access group ID | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/serviceaccessgroup" -X POST -H "CL-AUTH-TOKEN:5ebd84eb-9ea9-474b-803a-947e84e81b66" -d '{"apply":{"id": 4309}}'
```

Example Response: JSON

```
{"success":true,"errors":{},"response":{"serviceaccessgroup":[]}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

13.7 Update service access group

| HTTP Method | URL | Description |
|-------------|--------------------------------------|--|
| PUT | /core/api/v1/serviceaccessgroup/<id> | Update the existing service access group |

Request parameters: Request parameter

| Filter Name | Type | Description | Required |
|------------------------------------|------------|---------------------------------------|----------|
| Name | String | Access group name | Yes |
| Description | String | Access group description | No |
| hostType | Integer | Service engine type ID | Yes |
| serviceAccessGroupPropertiesBOList | Array List | Array list of access group properties | Yes |
| Protocol | String | Which protocol - tcp or udp | Yes |

| | | | |
|----------|---------|--|-----|
| fromPort | Integer | The start range of port | Yes |
| toPort | Integer | Up to range of port | Yes |
| cidrlp | String | For allocating IP addresses and routing Internet Protocol packets it is 0.0.0.0/0 or any other | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/serviceaccessgroup/4309" -X PUT -H "CL-AUTH-TOKEN:5ebd84eb-9ea9-474b-803a-947e84e81b66" -d '{"serviceAccessGroupPropertiesBOList":[{"cidrlp":"0.0.0.0/0","protocol":"tcp","fromPort":8080,"toPort":8080,"serviceAccessGroupPropertiesId":17933},{"cidrlp":"0.0.0.0/0","protocol":"tcp","fromPort":2144,"toPort":2144,"serviceAccessGroupPropertiesId":17934},{"cidrlp":"0.0.0.0/0","protocol":"tcp","fromPort":22,"toPort":22,"serviceAccessGroupPropertiesId":17935},{"cidrlp":"0.0.0.0/0","protocol":"tcp","fromPort":2100,"toPort":2100,"serviceAccessGroupPropertiesId":17944},{"cidrlp":"0.0.0.0/0","protocol":"tcp","fromPort":55,"toPort":55,"serviceAccessGroupPropertiesId":17945},{"cidrlp":"0.0.0.0/0","protocol":"tcp","fromPort":4444,"toPort":4444,"serviceAccessGroupPropertiesId":17946},{"cidrlp":"0.0.0.0/0","protocol":"tcp","fromPort":33,"toPort":33,"serviceAccessGroupPropertiesId":17947}], "hostType":2,"ownerType":1,"name":"Falcon_test","description":"falcon_test1","serviceEngineName":"MySQL5.5.27"}
```

Example Response:

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------|-----|
| Normal Response Code(s): | 200 |
|--------------------------|-----|

| | |
|--------------------------------|--|
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | QueryFailed(1009), entityDoesNotExist (1012), invalidParameterValue(1013), invalidRequest(1000) |

14. Availability zone

14.1 Get availability zone by target cloud ID

| HTTP Method | URL | Description |
|-------------|-------------------------------|---|
| GET | /core/api/v1/availabilityzone | To get availability zone by target cloud ID |

Request format: Query parameter

| Filter Name | Type | Description | Required |
|---------------|---------|--------------------------|----------|
| targetCloudID | Integer | Existing target cloud ID | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/availabilityzone?targetCloudId=50" -X GET -H "CL-AUTH-TOKEN:a29db2d5-d55b-4ef4-9f2f-18cf4f191316"
```

Example Response:

```
{"success":true,"errors":{},"response":{"availability_zones":[{"keypairValue":null,"availability
```

Zone":{"zone1","availabilityZoneName":"ContZone"}}}

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

15. Entity: Appdynamic

15.1 Get appdynamic account

| HTTP Method | URL | Description |
|-------------|-------------------------|----------------------------|
| GET | /core/api/v1/appdynamic | To get Appdynamics account |

Request Format:Query Parameter-

| Filter Name | Type | Description | Required |
|-------------|---------|------------------|----------|
| | | | |
| userId | Integer | Existing User ID | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/appdynamic?userId=2" -X GET -H "CL-AUTH-TOKEN:9708bec8-5fc6-44bc-b181-b9d99b7e83e5"
```

Example Response:

```
{
  "success": true,
  "errors": {},
  "response": {
    "appdynamic": [
      {
        "id": 1,
        "createdBy": 2,
        "name": "test",
        "controllerUrl": "cumulogic.appdynamics.com",
        "licenseKey": "eeg26dfh78",
        "createdDate": "2014-05-21 07:07:35"
      }
    ]
  }
}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |

15.2 Create Appdynamic account

| HTTP Method | URL | Description |
|-------------|-------------------------|------------------------------|
| POST | /core/api/v1/appdynamic | Add a new appdynamic account |

Common Request Parameters:

| Filter Name | Type | Description | Required |
|---------------|--------|--------------------------------|----------|
| licenseKey | String | Appdynamic account license key | Yes |
| Name | String | Account name | Yes |
| controllerUri | String | Controller URI | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/appdynamic" -X POST -H "CL-AUTH-TOKEN:1282681a-eb2e-4d87-b176-67a341d4c56a" -d'{"create":{"name":"test","controlerUrl": "cumulogic.appdynamics.com","licenseKey": "eeg26dfh78"}}'
```

Example Response:

```
{"success":true,"errors":{},"response":{"appdynamic":[{"id":1,"createdBy":2,"name":"test","controlerUrl":"cumulogic.appdynamics.com","licenseKey":"eeg26dfh78","createdDate":"2014-05-20 13:06:37"}]}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unprocessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityAlreadyExists(1011), invalidParameterValue(1013), invalidRequest(1000), invalidAction(1001) |

15.3 Delete appdynamic account

| HTTP Method | URL | Description |
|-------------|------------------------------|--|
| DELETE | /core/api/v1/appdynamic/<id> | Delete the existing appdynamic account |

Request Format: Query Parameter

| Filter Name | Type | Description | Required |
|-------------|------|-------------|----------|
| | | | |

| | | | |
|----|---------|---------------------------------------|-----|
| ID | Integer | ID of the existing appdynamic account | Yes |
|----|---------|---------------------------------------|-----|

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/appdynamic/1" -X DELETE -H "CL-AUTH-TOKEN:1282681a-eb2e-4d87-b176-67a341d4c56a"
```

Example Response: JSON

```
{"success":true,"errors":{},"response":{}}
```

Response Code(s):

| | |
|--------------------------------|--|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500,...), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), itemNotFound (404), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed(1009), entityDoesNotExist(1012), invalidRequest(1000) |

16. Access Group Template

16.1 Get access group template by host type

| HTTP Method | URL | Description |
|-------------|--------------------------------------|------------------------------|
| GET | /core/api/v1/ accessGroupTemplate | To get access group template |

Request Format:Query Parameter-

| Filter Name | Type | Description | Required |
|-------------|---------|--------------------|----------|
| hostType | Integer | Existing host type | Yes |

Example Request:

```
$ curl -i "http://localhost:8080/core/api/v1/accessGroupTemplate?hostType=1" -X GET -H "CL-AUTH-TOKEN:1282681a-eb2e-4d87-b176-67a341d4c56a"
```

Example Response:

```
{"success":true,"errors":{},"response":{"accessgroup":[{"id":1,"description":"Mysql-Management","name":"Mysql-Management","accessgroupproperties":[{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":2144,"toPort":2144,"serviceAccessGroupPropertiesId":240},{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":22,"toPort":22,"serviceAccessGroupPropertiesId":241},{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":3306,"toPort":3306,"serviceAccessGroupPropertiesId":242},{"cidrIp":"0.0.0.0/0","protocol":"tcp","fromPort":1186,"toPort":1186,"serviceAccessGroupPropertiesId":243}]}}}
```

Response Code(s):

| | |
|--------------------------------|---|
| Normal Response Code(s): | 200 |
| Error Response Code(s): | computeFault (400, 500), serviceUnavailable (503), unauthorized (401), forbidden (403), overLimit (413), requestTimeOut (408), unProcessableEntity (422), locked (423) |
| Custom Error Response Code(s): | queryFailed (1009), entityDoesNotExist (1012), invalidRequest (1000), invalidParamter (1010) |