

# **vDC Automation Standard Edition**

## **ポータル Web API リファレンス**

---

# 免責事項

本書の内容はすべて日本電気株式会社が所有する著作権に保護されています。

本書の内容の一部または全部を無断で転載および複製することは禁止されています。

本書の内容は将来予告なしに変更することがあります。

日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任を負いません。

日本電気株式会社は、本書の内容に関し、その正確性、有用性、確実性その他いかなる保証もいたしません。

# 商標

- SigmaSystemCenter、JobCenter、WebSAM は日本電気株式会社の登録商標です。
- Microsoft、Windows、Windows ロゴ、Windows Server、Internet Explorer、SQL Server および Azure は 米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- Linux は Linus Torvalds 氏の米国およびその他の国における登録商標または商標です。
- Intel、Itanium は、Intel 社の米国およびその他の国における登録商標または商標です。
- Apache、Apache Tomcat、Tomcat は、Apache Software Foundation の登録商標または商標です。
- Firefox は Mozilla Foundation の登録商標または商標です。
- Google Chrome は Google Inc.の登録商標または商標です。
- PostgreSQL は、PostgreSQL の米国およびその他の国における登録商標または商標です。
- Amazon Web Services、およびその他の AWS 商標は、米国その他の諸国における Amazon.com, Inc.またはその関連会社の商標です。
- OpenStack のワードマークは、米国とその他の国における OpenStack Foundation の登録商標/サービスマークまたは商標/サービスマークのいずれかであり、OpenStack Foundation の許諾の下に使用されています。
- Oracle と Java は Oracle Corporation およびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

その他、本書に記載のシステム名、会社名、製品名は、各社の登録商標もしくは商標です。

なお、® マーク、™ マークは本書に明記しておりません。

---

# 目次

<b>第 1 章 Web API について</b> .....	<b>1</b>
1.1 権限 .....	1
1.2 共通の HTTP ステータスコード .....	1
<b>第 2 章 Web API リファレンス</b> .....	<b>2</b>
2.1 テナント API .....	2
2.1.1 テナント一覧取得 .....	2
2.1.2 テナント作成 .....	4
2.1.3 テナント削除 .....	5
2.2 ユーザ API .....	7
2.2.1 ユーザ一覧取得 .....	7
2.2.2 ユーザの作成(v1.0) .....	9
2.2.3 ユーザの作成(v1.1) .....	12
2.2.4 ユーザ削除 .....	15
2.2.5 ユーザ詳細表示 .....	16
2.3 ロール管理 API .....	18
2.3.1 ロール一覧取得 .....	19
2.3.2 ロール作成 .....	21
2.3.3 ロール詳細 .....	24
2.3.4 ロール削除 .....	26
2.4 リクエスト API .....	27
2.4.1 リクエスト一覧取得 .....	28
2.4.2 リクエスト承認 .....	30
2.4.3 リクエストエラー状態解除 .....	32
2.4.4 リクエスト実行 .....	33
2.4.5 リクエストキャンセル .....	35
2.4.6 リクエスト却下 .....	36
2.4.7 サーバ作成リクエストの作成 .....	37
2.4.8 サーバ削除リクエストの作成 .....	44
2.5 サーバ API .....	45
2.5.1 サーバ一覧取得 .....	45
2.5.2 サーバ詳細情報取得 .....	48
2.5.3 サーバ起動 .....	51
2.5.4 サーバ停止 .....	52
2.5.5 サーバ電源 OFF .....	53
2.5.6 サーバ再起動 .....	55
2.5.7 サーバ同期 .....	56
2.6 スコープ管理 API .....	57
2.6.1 スコープ一覧取得(v1.0) .....	57
2.6.2 スコープ一括追加(v1.0) .....	59
2.6.3 スコープ一括更新(v1.0) .....	66
2.6.4 スコープ一括削除(v1.0) .....	69
2.6.5 スコープ一括追加(v2.0) .....	71

---

2.6.6	スコープ一括更新(v2.0)	77
2.6.7	スコープ一括削除(v2.0)	80
2.7	ログインユーザ管理 API	83
2.7.1	ユーザの詳細情報を取得する	83
2.8	ユーザ管理基盤 API	87
2.8.1	ユーザ認証	88
2.8.2	認証トークン検証	89
2.8.3	認証トークン破棄	91
2.9	プラグインインスタンス API	92
2.9.1	プラグインインスタンス一覧取得	93
2.9.2	プラグインインスタンス操作	95

# 第1章

## Web API について

本章では、Web API の概要について説明します。

WebAPI は各ユーザが払い出された API キーを設定し使用します。

新規ユーザへの API キーの払い出しについては、別紙『vDC Automation Standard Edition ポータル利用者マニュアル(リソース管理編)』「ユーザを登録する (管理者)」を参照してください。

また、Ver1.1 のユーザ作成 API では、レスポンスボディに API キーが含まれますので、API 実行でキーを払い出すことも可能です。詳細は、「[2.2.3 ユーザの作成\(v1.1\) \(12 ページ\)](#)」を参照してください。

既存のユーザに API キーを払い出す場合は、別紙『vDC Automation Standard Edition ポータル利用者マニュアル(リソース管理編)』「ログインユーザ情報を変更する」から API キーの設定をおこなってください。

### 1.1 権限

API の使用には権限が必要です。詳細は各 API の節を参照してください。

### 1.2 共通の HTTP ステータスコード

API の成功・失敗は、HTTP ステータスコードで通知します。

全 API で共通の HTTP ステータスコードは次の通りです。

コード	意味	説明
400	Bad Request	渡されたパラメータが異なるなど、要求が正しくない場合に返却される
401	Unauthorized	適切な認証情報を提供せず、保護されたリソースに対しアクセスをした場合に返却される
403	Forbidden	権限が不足した場合に返却される
404	Not Found	指定された URL のリソースが見つからない
405	Method Not Allowed	要求したリソースがサポートしていない HTTP メソッドを利用した場合に返却される
500	Internal Server Error	API 実行時に予期しないエラーが発生した場合に返却される

**注:** 上記以外にも各 API で HTTP ステータスコードを定義します。詳細は各 API の仕様を参照してください。

# 第2章

## Web API リファレンス

本章では、Web API の仕様について説明します。

### 2.1 テナント API

テナント API について説明します。

#### 【権限】

テナント API に必要とする権限は以下です。

API 機能	権限	説明
テナント一覧の表示	ROLE_TENANT_LIST_SHOW	テナント一覧情報の取得と検索
テナント作成	ROLE_TENANT_CREATE	テナント作成リクエスト
テナント削除	ROLE_TENANT_DELETE	テナントを削除する
テナント詳細表示	ROLE_TENANT_DETAIL_SHOW	テナント詳細情報を取得する

#### 2.1.1 テナント一覧取得

テナントの一覧取得・検索をします。検索条件についてはリクエストパラメータを参照してください。

#### URL

```
GET /cloudportal/api/v1.0/tenants[?[valid=<true|false>]]
```

#### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_TENANT_LIST_SHOW の権限を持つこと

## リクエストパラメータ

リクエストパラメータ名	省略可否	説明
valid	可	<ul style="list-style-type: none"> <li>テナントの状態を指定する 指定された状態のテナントを取得する <ul style="list-style-type: none"> <li>- 有効 : true</li> <li>- 無効 : false</li> </ul> </li> <li>valid を指定しない場合、有効・無効の両方のテナントを取得する</li> </ul>

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功
400	Bad Request	不正なテナント

## レスポンスボディ

JSON キー	説明
tenantId	テナント ID
tenantName	テナント名
enabled	テナントの状態(有効/無効)
uploadTime	テナントの登録時間 (yyyy/MM/dd HH:mm:ss)
remarks	テナントの備考

## 実行例

### 【リクエスト】

```
GET /cloudportal/api/v1.0/tenants HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  [
    {
```

```

    "tenantId": "TenantA",
    "tenantName": "TenantA",
    "enabled": true,
    "uploadTime": "2018/06/27 11:05:24",
    "remarks": ""
  },
  {
    "tenantId": "TenantB",
    "tenantName": "TenantB",
    "enabled": true,
    "uploadTime": "2018/07/04 17:47:30",
    "remarks": "TenantB備考"
  },
  {
    "tenantId": "TenantC",
    "tenantName": "TenantC",
    "enabled": false,
    "uploadTime": "2018/07/04 17:47:41",
    "remarks": "TenantC備考"
  }
]
}

```

## 2.1.2 テナント作成

指定されたパラメータでテナントを作成します。

### URL

```
POST /cloudportal/api/v1.0/tenants
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_TENANT_CREATE の権限を持つこと

### リクエストパラメータ

なし



## リクエストボディ

キー	型	省略可否	説明
tenantId	string	不可	テナント ID
tenantName	string	不可	テナント名
enabled	boolean	不可	状態 <ul style="list-style-type: none"> <li>有効: true</li> <li>無効: false</li> </ul>
remarks	string	可	備考

## レスポンスコード

コード	意味	説明
201	Created	作成成功
400	Bad Request	リクエスト情報が不正
409	Conflict	テナント ID に指定されたテナントが既に存在する場合

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v1.0/tenants HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
{
  "tenantId": "TenantA",
  "tenantName": "TenantA",
  "enabled": true,
  "remarks": "グループ用テナント"
}
```

### 【レスポンス】

```
HTTP/1.1 201 Created
```

### 2.1.3 テナント削除

指定したテナントを削除します。

## URL

```
DELETE /cloudportal/api/v1.0/tenants/{tenantId}
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_TENANT_DELETE の権限を持つこと

## リクエストパラメータ

リクエストパラメータ	省略可否	説明
tenantId	不可	テナント ID

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
204	NoContent	削除成功
404	Not Found	指定したテナント ID が存在しない
409	Conflict	テナントに所属しているユーザが存在している テナントに所属しているサーバが存在している テナントに所属しているネットワークが存在している テナント向けのお知らせがある テナント向けのメンテナンスがある 承認待ち、実行中のリクエストがある 監視エージェントがある テナント拡張チェックを実施失敗

## 実行例

### 【リクエスト】

```
DELETE /cloudportal/api/v1.0/tenants/TenantA HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

**【レスポンス】**

```
HTTP/1.0 204 NoContent
```

**注**

監視ライセンスが有効になっている場合には、テナント削除 API は使用できません。  
事前に、監視ライセンスを無効に設定してください。

## 2.2 ユーザ API

ユーザ API について説明します。

**【権限】**

ユーザ API に必要とする権限は以下です。

API 機能	権限	説明
ユーザー一覧表示	ROLE_USER_LIST	ユーザー一覧情報の取得と検索
ユーザ作成	ROLE_USER_CREATE	ユーザ作成リクエスト
ユーザ削除	ROLE_USER_DELETE	ユーザ削除リクエスト
ユーザ詳細表示	ROLE_USER_DETAIL	ユーザ詳細情報の取得

### 2.2.1 ユーザー一覧取得

ユーザー一覧を取得します。

#### URL

```
GET /cloudportal/api/v1.0/users
```

#### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_USER_LIST の権限を持つこと

## リクエストパラメータ

なし

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功

## レスポンスボディ

JSON キー	説明
name	氏名
userId	ユーザ ID
roleType	ロール種別 <ul style="list-style-type: none"> <li>• ROLE_SYSTEM_ADMIN</li> <li>• ROLE_TENANT_ADMIN</li> <li>• ROLE_TENANT_USER</li> <li>• テナント固有ロール ※</li> </ul>
enabled	状態 <ul style="list-style-type: none"> <li>• 有効 : true</li> <li>• 無効 : false</li> </ul>
tenantId	所属するテナントのテナント ID

## 実行例

### 【リクエスト】

```
GET /cloudportal/api/v1.0/users HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "users" : [
    {
      "name": "テナントユーザA",
      "userId": "TenantA_User",
```

```

    "roleType": "ROLE_TENANT_USER",
    "enabled": true,
    "tenantId": "TenantA"
  },
  {
    "name": "テナント管理者A",
    "userId": "TenantA_Admin",
    "roleType": "ROLE_TENANT_ADMIN",
    "enabled": true,
    "tenantId": "TenantA"
  },
  {
    "name": "テナントユーザB",
    "userId": "TenantB_User",
    "roleType": "ROLE_TENANT_USER",
    "enabled": true,
    "tenantId": "TenantB"
  }
]
}

```

## ヒント

※ テナント固有ロールの詳細は vDC Automation Standard Edition ポータル利用者マニュアル (リソース管理編)の「ロールについて」を参照してください。

## 2.2.2 ユーザの作成(v1.0)

指定されたパラメータでユーザを作成します。

## URL

```
POST /cloudportal/api/v1.0/users
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_USER_CREATE の権限を持つこと

## リクエストパラメータ

なし

## リクエストボディ

キー	型	省略可否	説明
name	string	不可	ユーザ氏名 null 指定不可
tenantId	string	不可	所属するテナントのテナント ID <ul style="list-style-type: none"> <li>• roleType が ROLE_SYSTEM_ADMIN の場合、null を指定</li> <li>• roleType が ROLE_TENANT_ADMIN の場合、テナント ID を指定</li> <li>• roleType が ROLE_TENANT_USER の場合、テナント ID を指定</li> </ul>
userId	string	不可	ユーザ ID null 指定不可
approval	boolean	可	リクエストの承認が可能か否か <ul style="list-style-type: none"> <li>• true : リクエスト承認可能</li> <li>• false : リクエスト承認不可</li> <li>• 省略時は false を設定</li> </ul>
roleType	string	不可	ロール <ul style="list-style-type: none"> <li>• ROLE_SYSTEM_ADMIN</li> <li>• ROLE_TENANT_ADMIN</li> <li>• ROLE_TENANT_USER</li> <li>• テナント固有ロール ※</li> </ul>
customRoleTypes	string[]	可	カスタムロール
email	string	不可	E-Mail アドレス null 指定不可
password	string	不可	ユーザのパスワード
remarks	string	可	備考
enabled	boolean	不可	状態 <ul style="list-style-type: none"> <li>• true : 有効</li> <li>• false : 無効</li> </ul>

## レスポンスコード

コード	意味	説明
201	Created	作成成功
400	Bad Request	<p>ユーザ状態の指定が不正</p> <p>ロール種別が ROLE_SYSTEM_ADMIN かつテナント ID を指定した場合</p> <p>指定テナントで利用できないテナント固有ロールを指定した場合</p> <p>操作ユーザが利用できないテナント固有ロールを指定した場合</p> <p>操作ユーザが所属するテナントとは別テナントを指定した場合</p> <p>ロール種別に</p> <ul style="list-style-type: none"> <li>• ROLE_SYSTEM_ADMIN</li> <li>• ROLE_TENANT_ADMIN</li> <li>• ROLE_TENANT_USER</li> <li>• テナント固有ロール※</li> </ul> <p>以外が指定された場合</p>
404	Not Found	<p>ロール種別に 「ROLE_TENANT_ADMIN」を指定した場合に、存在しないテナント ID を指定する場合</p>
409	Conflict	ユーザ ID が既に存在する場合

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/users HTTP/1.1
Content-Type:application/json; charset=utf-8
ApiKey:ABCDEFGHIJKLMNQRSTUUV1234567890
{
  "name": "テナント管理者A",
  "tenantId": "TenantA",
  "userId": "TenantA_user",
  "approval": true,
  "roleType": "ROLE_TENANT_USER",
  "customRoleTypes": ["CUSTOM_ROLE_1", "CUSTOM_ROLE_2"],
  "email": "TenantA_user@local.com",
  "password": "123456",
  "remarks": "テナントユーザAの備考",
  "enabled": true
}
```

## 【レスポンス】

```
HTTP/1.1 201 Created
```

## ヒント

※テナント固有ロールの詳細は vDC Automation Standard Edition ポータル利用者マニュアル (リソース管理編)の「ロールについて」を参照してください。

## 2.2.3 ユーザの作成(v1.1)

指定されたパラメータでユーザを作成します。

## URL

```
POST /cloudportal/api/v1.1/users
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_USER_CREATE の権限を持つこと

## リクエストパラメータ

なし

## リクエストボディ

キー	型	省略可否	説明
name	string	不可	ユーザ氏名 null 指定不可
tenantId	string	不可	所属するテナントのテナント ID <ul style="list-style-type: none"> <li>• roleType が ROLE_SYSTEM_ADMIN の場合、null を指定</li> <li>• roleType が ROLE_TENANT_ADMIN の場合、テナント ID を指定</li> </ul>



キー	型	省略可否	説明
			<ul style="list-style-type: none"> <li>roleType が ROLE_TENANT_USER の場合、テナント ID を指定</li> </ul>
userId	string	不可	ユーザ ID null 指定不可
enableApiKey	boolean	不可	true:apiKey を自動的に生成する false:apiKey を自動的に生成しない
approval	boolean	可	リクエストの承認が可能か否か <ul style="list-style-type: none"> <li>true: リクエスト承認可能</li> <li>false: リクエスト承認不可</li> <li>省略時は false を設定</li> </ul>
roleType	string	不可	ロール <ul style="list-style-type: none"> <li>ROLE_SYSTEM_ADMIN</li> <li>ROLE_TENANT_ADMIN</li> <li>ROLE_TENANT_USER</li> <li>テナント固有ロール ※</li> </ul>
customRoleTypes	string[]	可	カスタムロール種別を指定 複数ある場合はカンマ区切りで指定
email	string	不可	E-Mail アドレス null 指定不可
password	string	不可	ユーザのパスワード
remarks	string	可	備考
enabled	boolean	不可	状態 <ul style="list-style-type: none"> <li>true: 有効</li> <li>false: 無効</li> </ul>

## レスポンスコード

コード	意味	説明
201	Created	作成成功
400	Bad Request	ユーザ状態の指定が不正 ロール種別が ROLE_SYSTEM_ADMIN かつテナント ID を指定した場合

コード	意味	説明
		指定テナントで利用できないテナント固有ロールを指定した場合 操作ユーザが利用できないテナント固有ロールを指定した場合 操作ユーザが所属するテナントとは別テナントを指定した場合 ロール種別に <ul style="list-style-type: none"> <li>• ROLE_SYSTEM_ADMIN</li> <li>• ROLE_TENANT_ADMIN</li> <li>• ROLE_TENANT_USER</li> <li>• テナント固有ロール※</li> </ul> 以外が指定された場合
404	Not Found	ロール種別に「ROLE_TENANT_ADMIN」を指定した場合に、存在しないテナントIDを指定する場合
409	Conflict	ユーザ ID が既に存在する場合

## レスポンスボディ

JSON キー	説明
ApiKey	リクエストの enableApiKey が true の場合：作成するユーザの API キー(自動生成された 32 文字の文字列)を返却 リクエストの enableApiKey が false の場合：null を返却

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v1.1/users HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

```
{
  "name": "テナント管理者A",
  "tenantId": "TenantA",
  "userId": "TenantA_Admin",
  "enableApiKey": true,
  "approval": true,
  "roleType": "ROLE_TENANT_ADMIN",
  "customRoleTypes": ["CUSTOM_ROLE_1", "CUSTOM_ROLE_2"],
  "email": "TenantA_Admin@local.com",
  "password": "123456",
  "remarks": "テナント管理者Aの備考",
  "enabled": true
}
```

### 【レスポンス】

- enableApiKey:true の場合

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
{
  "ApiKey" : "eebb4c15b286462597d3ee5980c9fb67"
}
```

- enableApiKey:false の場合

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
{
  "ApiKey" : null
}
```

## ヒント

※テナント固有ロールの詳細は vDC Automation Standard Edition ポータル利用者マニュアル (リソース管理編)の「ロールについて」を参照してください。

## 2.2.4 ユーザ削除

指定したユーザを削除します。

### URL

```
DELETE /cloudportal/api/v1.0/users/{userId}
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_USER_DELETE の権限を持つこと

### リクエストパラメータ

リクエストパラメータ	省略可否	説明
userId	不可	ユーザ ID

### リクエストボディ

なし

## レスポンスコード

コード	意味	説明
204	NoContent	削除成功
400	Bad Request	削除したいユーザがビルトインユーザの場合 APIを持っているユーザの操作するロール種別は以下の場合 <ul style="list-style-type: none"> <li>システム管理者タイプは、テナント代行してないシステム管理者のみ</li> <li>テナント管理者タイプ・テナントユーザタイプは、テナント代行していないシステム管理者もしくは、テナント代行者、テナント管理者</li> </ul> SysMgrG 用属性情報の取得失敗
404	Not Found	指定したユーザ ID が存在しない場合

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
DELETE /cloudportal/api/v1.0/users/TenantA_user01 HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.0 204 NoContent
```

## 2.2.5 ユーザ詳細表示

指定したユーザ詳細情報を取得する

## URL

```
GET /cloudportal/api/v1.0/]users/{userId}
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_USER_DETAIL の権限を持つこと

## リクエストパラメータ

リクエストパラメータ	省略可否	説明
userId	不可	ユーザ ID

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	Success	取得成功
404	Not Found	指定したユーザ ID が存在しない場合

## レスポンスボディ

JSON キー	説明
tenantId	テナント ID
userId	ユーザ ID
name	ユーザの氏名
enableApiKey	API キー true : API キーがある false : API キーがない
roleType	ロール ・ ROLE_SYSTEM_ADMIN ・ ROLE_TENANT_ADMIN ・ ROLE_TENANT_USER テナント固有ロール ※
approval	リクエスト承認できるかどうか true : 承認可能 false : 承認不可

JSON キー	説明
customRoleType	追加するロール（複数の場合カンマ「,」区切りで記載）※
enabled	状態 <ul style="list-style-type: none"> <li>• true : 有効</li> <li>• false : 無効</li> </ul>
email	e-mail
remarks	備考

## 実行例

### 【リクエスト】

```
GET /cloudportal/api/v1.0/users/TenantA_user01 HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.0 200 OK
Content-Type: application/json; charset=utf-8
{
  "user" : {
    "tenantId" : "TenantA",
    "userId" : "TenantA_user01",
    "name" : "テストユーザ01",
    "enableApikey" : true,
    "roleType" : "ROLE_TENANT_USER",
    "approval" : true,
    "customRoleTypes" : [CUSROM_ROLE_01,CUSROM_ROLE_02],
    "enabled" : true,
    "email" : "user01@vdca.local",
    "remarks" : "テストユーザ01の備考"
  }
}
```

### ヒント

※ カスタムロール、テナント固有ロールの詳細は vDC Automation Standard Edition ポータル利用者マニュアル(リソース管理編)の「ロールについて」を参照してください。

## 2.3 ロール管理 API

ロール管理 API について説明します。

### 【権限】

ロール管理に必要とする権限は以下です。

API 機能	権限	説明
ロール一覧の表示	ROLE_SCOPEDROLE_LIST	ロール一覧情報の取得と検索
ロール詳細の表示	ROLE_SCOPEDROLE_SHOW	ロール詳細情報の取得
ロール作成	ROLE_SCOPEDROLE_CREATE	ロール作成
ロール削除	ROLE_SCOPEDROLE_DELETE	ロール削除

### 2.3.1 ロール一覧取得

ロールの一覧取得・検索をします。検索条件についてはリクエストパラメータを参照してください。

#### URL

```
GET /portal/cloudportal/api/v1.0/roles[?[id=<id>][name=<name>]]
```

#### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPEDROLE_LIST の権限を持つこと

#### リクエストパラメータ

リクエストパラメータ名	省略可否	説明
id	可	ロールの ID を指定する。指定された値を含むロールにフィルタリングして取得する。
name	可	ロールの表示名を指定する。指定された値を含むロールにフィルタリングして取得する。

#### リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功

## レスポンスボディ

JSON キー	説明
roles[].id	ロールの ID
roles[].name	ロールの表示名
roles[].description	ロールの説明
roles[].tenantId	ロールが所属するテナントのテナント ID

### 注

API を実行したユーザが所属するテナントのテナント固有ロールのみ返却されます。

API を実行したユーザがシステム管理者の場合、すべてのテナントが対象となります。

## 実行例

### 【リクエスト】

```
GET /portal/cloudportal/api/v1.0/roles
```

### 【レスポンス】

```
HTTP/1.1 200
Cache-Control: no-store
Pragma:
Expires:
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Set-Cookie: JSESSIONID=ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890; Path=/portal; HttpOnly
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Fri, 13 Sep 2019 10:46:35 GMT

{
  "roles": [
    {
      "id": "ServerManager",
      "name": "サーバ管理者",
      "description": "作成されたサーバを管理する権限のみ有するロール",
      "tenantId": "TenantA"
    },
    {
      "id": "NetworkManager",
      "name": "ネットワーク管理者",
      "description": "作成されたネットワークを管理する権限のみ有するロール",
      "tenantId": "TenantA"
    }
  ]
}
```



```

    },
    {
      "id": "RequestManager",
      "name": "リクエスト管理者",
      "description": "リクエスト作成に必要な権限のみ有するロール.",
      "tenantId": "TenantA"
    }
  ]
}

```

## 2.3.2 ロール作成

リクエストボディで指定された内容でロールを作成します。

### URL

```
POST /portal/cloudportal/api/v1.0/roles
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPEDROLE_CREATE の権限を持つこと

### リクエストパラメータ

なし

### リクエストボディ

キー	型	省略可否	説明
id	string	不可	ロールの ID
name	string	不可	ロール表示名
description	string	可	ロールの説明
products[].id	string	不可	付与する権限が所属する ファンクションが所属する プロダクトの ID 以下のプロダクトが指定 可能 <ul style="list-style-type: none"> <li>• CloudPortal</li> <li>• vDCA</li> </ul>

キー	型	省略可否	説明
products[].functions[].id	string	不可	付与する権限が所属する ファンクションの ID 以下のファンクションが 指定可能 <ul style="list-style-type: none"> <li>CloudPortal 配下の ファンクションの場合、『利用者マニュアル(リソース管理編)』付録『CloudPortal プロダクトにて扱う機能および権限』の機能名を参照</li> <li>vDCA 配下のファンクションの場合、『利用者マニュアル(リソース管理編)』付録『vDCA プロダクトにて扱う機能および権限』の機能名を参照</li> </ul>
products[].functions[].allows[]	string[]	不可	付与する権限のリスト 以下の権限が指定可能 <ul style="list-style-type: none"> <li>CloudPortal 配下の ファンクションの場合、『利用者マニュアル(リソース管理編)』付録『CloudPortal プロダクトにて扱う機能および権限』の権限名を参照</li> <li>vDCA 配下のファンクションの場合、『利用者マニュアル(リソース管理編)』付録『vDCA プロダクトにて扱う機能および権限』の権限名を参照</li> </ul>

## レスポンスコード

コード	意味	説明
201	Created	作成成功
400	Bad Request	リクエストボディに指定したパラメータに誤りがある
404	Not Found	リクエストボディに指定したプロダクトまたはファンクションまたは権限が存在しない
409	Conflict	指定された ID のロールが既に登録されている

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
POST /portal/cloudportal/api/v1.0/roles
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216

{
  "id": "MonitoringAdmin",
  "name": "監視管理者用ロール",
  "description": "監視管理者用のロール",
  "products": [
    {
      "id": "SysMgrG",
      "functions": [
        {
          "id": "status",
          "scopes": [
            {
              "id": "status",
              "allows": [
                "MSC_GENERAL_REFERENCE"
              ]
            }
          ],
          "allows": [
            "MSC_GENERAL_REFERENCE"
          ]
        }
      ]
    }
  ]
}
```

### 【レスポンス】

```
HTTP/1.1 201
Cache-Control: no-store
Pragma:
Expires:
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Set-Cookie: JSESSIONID=ABCDEFGHIJKLMNQRSTUUV1234567890; Path=/portal; HttpOnly
Content-Length: 0
Date: Sun, 15 Sep 2019 23:57:52 GMT
```

**注**

リクエストボディのロール表示名について、重複した値の登録は可能ですが、既に登録済みのロールと識別しづらくなる可能性があります。事前に、ロール管理画面から既存のロール名を確認していただき、重複しない一意となる値の設定を推奨します。

### 2.3.3 ロール詳細

URL に含まれる roleId で指定したロールの詳細を取得をします。

#### URL

```
GET /portal/cloudportal/api/v1.0/roles/{roleId}
```

#### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPEDROLE_SHOW の権限を持つこと

#### リクエストパラメータ

リクエストパラメータ名	省略可否	説明
roleId	不可	ロールの ID を指定する。

#### リクエストボディ

なし

#### レスポンスコード

コード	意味	説明
200	OK	成功
404	Not Found	指定された ID のロールが存在しない 操作ユーザが参照できないテナント固有ロールを指定した場合

## レスポンスボディ

JSON キー	説明
role.id	ロールの ID
role.name	ロールの表示名
role.description	ロールの説明
role.tenantId	ロールが所属するテナントのテナント ID
role.products[].id	権限が付与されているプロダクトの ID
role.products[].functions[].id	権限が付与されているプロダクト内の関数の ID
role.products[].functions[].allows	付与されている権限の一覧

## 実行例

### 【リクエスト】

```
GET /portal/cloudportal/api/v1.0/roles/ServerManager
```

### 【レスポンス】

```
HTTP/1.1 200
Cache-Control: no-store
Pragma:
Expires:
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Set-Cookie: JSESSIONID=ABCDEFGHIJKLMNQPQRSTUVWXYZ1234567890; Path=/portal; HttpOnly
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Date: Fri, 13 Sep 2019 10:46:35 GMT
```

```
{
  "role": {
    "id": "ServerManager",
    "name": "サーバ管理者",
    "description": "作成されたサーバを管理する権限のみ有するロール",
    "tenantId": "TenantA",
    "products": [
      {
        "id": "CloudPortal",
        "functions": [
          {
            "id": "LoginUser",
            "allows": [
              "ROLE_LOGINUSER_EDIT",
              "ROLE_LOGINUSER_PASSWORD_EDIT"
            ]
          }
        ]
      }
    ]
  },
  {
    "id": "vDCA",
```

```

    "functions": [
      {
        "id": "Server",
        "allows": [
          "ROLE_SERVER_MENU_SHOW",
          "ROLE_SERVER_LIST_SHOW",
          "ROLE_SERVER_DETAIL_SHOW",
          "ROLE_SERVER_SCREENSHOT_SHOW",
          "ROLE_SERVER_STARTUP",
          "ROLE_SERVER_SHUTDOWN",
          "ROLE_SERVER_POWEROFF",
          "ROLE_SERVER_REBOOT",
          "ROLE_SERVER_VM_CONSOLE",
          "ROLE_SERVER_GENERIC_OPERATION"
        ]
      }
    ]
  }
}

```

### 2.3.4 ロール削除

URL に含まれる `roleId` で指定したロールを削除をします。

#### URL

```
DELETE /portal/cloudportal/api/v1.0/roles/{roleId}
```

#### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPEDROLE_DELETE の権限を持つこと

#### リクエストパラメータ

リクエストパラメータ名	省略可否	説明
roleId	不可	削除対象となるロールの ID を指定する。

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
204	OK	削除成功
400	Bad Request	指定された ID のロールが割り当てられているユーザが存在するため、削除できない
404	Not Found	指定された ID のロールが存在しない

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
DELETE /portal/cloudportal/api/v1.0/roles/ServerManager
```

### 【レスポンス】

```
HTTP/1.1 204
Cache-Control: no-store
Pragma:
Expires:
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Set-Cookie: JSESSIONID=ABCDEFGHIJKLMNQPQRSTUVWXYZ1234567890; Path=/portal; HttpOnly
Date: Mon, 16 Sep 2019 00:03:02 GMT
```

## 2.4 リクエスト API

リクエスト API について説明します。

### 【権限】

リクエスト API に必要とする権限は以下です。

API 機能	権限	説明
リクエスト一覧の表示	ROLE_REQUEST_LIST_SHOW	リクエスト一覧情報の取得と検索

API 機能	権限	説明
サーバ作成リクエストの作成	ROLE_REQUEST_CREATE および ROLE_REQUEST_SERVER_CREATE	サーバ作成リクエストの申請
サーバ削除リクエストの作成	ROLE_REQUEST_CREATE および ROLE_REQUEST_SERVER_DELETE	サーバ削除リクエストの申請
リクエスト実行	ROLE_REQUEST_EXECUTE	サーバリクエストの実行
リクエストキャンセル	ROLE_REQUEST_CANCEL	サーバリクエストのキャンセル
リクエスト承認(代理承認)	ROLE_REQUEST_APPROVE または ROLE_REQUEST_AGENCY_APPROVE	サーバリクエストの承認 (代理承認)
リクエスト却下(代理却下)	ROLE_REQUEST_REJECT または ROLE_REQUEST_AGENCY_REJECT	サーバリクエストの却下 (代理却下)
リクエストエラー状態解除	ROLE_REQUEST_ERRORCLEAR	サーバリクエストのエラー 状態解除

## 2.4.1 リクエスト一覧取得

リクエスト一覧を取得します。

### URL

```
GET /cloudportal/api/v1.0/requests[?[status=<status>][&type=<kind>]]
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_REQUEST_LIST_SHOW の 権限を持つこと

### リクエストパラメータ

リクエストパラメータ名	省略可否	説明
status	可	リクエストの状態 省略する場合、全ての状態のリクエスト
kind	可	省略する場合、全てのタイプのリクエスト

status で利用できる値



表 2-1 status で利用できる値

値	説明
FAILED	失敗
ADMITWAIT	承認待ち
REJECT	却下
DOING	実行中
COMPLETE	完了
EXECUTIONWAIT	実行待機
CANCELED	キャンセル

kind で利用できる値

表 2-2 kind で利用できる値

値	説明
SERVERCREATE	サーバ作成
SERVERCHANGE	サーバ変更
SERVERDELETE	サーバ削除
STACKCREATE	スタック作成
STACKDELETE	スタック削除
LOGICALNETWORKCREATE	論理ネットワーク作成
LOGICALNETWORKDELETE	論理ネットワーク削除

## レスポンスコード

コード	意味	説明
200	OK	成功

## レスポンスボディ

JSON キー	説明
requests[].tenantId	テナント ID
requests[].requestId	作成したリクエストの ID
requests[].status	リクエストの状態 「表 2-1 status で利用できる値 (29 ページ)」を参照
requests[].kind	リクエスト種別 「表 2-2 kind で利用できる値 (29 ページ)」を参照
requests[].nextApprover	次の承認者
requests[].progress	進捗率
requests[].requestDatetime	リクエスト日時 (サーバのタイムゾーンに合わせて表示する)
requests[].approver	申請者

## 実行例

### リクエストボディ

```
GET /cloudportal/api/v1.0/requests HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "requests": [
    {
      "tenantId": "TenantA",
      "requestId": "R000000001",
      "status": "COMPLETE",
      "kind": "LOGICALNETWORKCREATE",
      "nextApprover": null,
      "progress": 100,
      "requestDatetime": "2020/03/06 10:43:44",
      "approver": null
    },
    {
      "tenantId": "TenantA",
      "requestId": "R000000002",
      "status": "COMPLETE",
      "kind": "LOGICALNETWORKCREATE",
      "nextApprover": null,
      "progress": 100,
      "requestDatetime": "2020/03/06 10:43:44",
      "approver": null
    },
    {
      "tenantId": "TenantA",
      "requestId": "R000000003",
      "status": "COMPLETE",
      "kind": "SERVERCREATE",
      "nextApprover": "TenantA_Admin1",
      "progress": 100,
      "requestDatetime": "2020/03/06 14:16:37",
      "approver": "管理者"
    }
  ]
}
```

### 2.4.2 リクエスト承認

指定したリクエストを承認します。

## URL

```
PUT /cloudportal/api/v1.0/requests/{requestId}/approve
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_REQUEST_APPROVE または ROLE_REQUEST_AGENCY_APPROVE の権限を持つこと

## リクエストパラメータ

リクエストパラメータ名	省略可否	説明
requestId	不可	承認したいリクエストのリクエスト ID

## リクエストボディ

キー	省略可否	説明
admitComment	可	申請者へのコメント

## レスポンスコード

コード	意味	説明
200	OK	成功
400	Bad Request	リクエストの状態は承認できない場合
404	Not Found	指定したリクエストが存在しない場合

## レスポンスボディ

JSON キー	説明
status	リクエストの状態 ※1

※1 リクエストの状態: 1. DOING、2. ADMITWAIT、3. EXECUTIONWAIT

## 実行例

### 【リクエスト】

```
PUT /cloudportal/api/v1.0/requests/R000000001/approve HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
{
  "admitComment": "同意します。"
}
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "status": "ADMITWAIT"
}
```

## 2.4.3 リクエストエラー状態解除

指定したリクエストエラー状態を解除します。

### URL

```
PUT /cloudportal/api/v1.0/requests/{requestId}/errorclear
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_REQUEST_ERRORCLEAR の権限を持つこと

### リクエストパラメータ

リクエストパラメータ名	省略可否	説明
requestId	不可	エラー状態を解除したいリクエストのリクエスト ID

### リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功
400	Bad Request	リクエストの状態が「エラー」ではない場合
404	Not Found	指定したリクエストが存在しない場合

## レスポンスボディ

JSON キー	説明
status	リクエストの状態 EXECUTIONWAIT を返却

## 実行例

### 【リクエスト】

```
PUT /cloudportal/api/v1.0/requests/R00000001/errorclear HTTP/1.0
Content-Type: application/json; charset=utf-8
apikey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "status": "ADMITWAIT"
}
```

## 2.4.4 リクエスト実行

指定したリクエストを実行します。

### URL

```
PUT /cloudportal/api/v1.0/requests/{requestId}/execute
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列)

リクエストヘッダ名	省略可否	説明
		※API 利用者は ROLE_REQUEST_EXECUTE の権 限を持つこと

## リクエストパラメータ

リクエストパラメータ名	省略可否	説明
requestId	不可	実行したいリクエストのリクエ スト ID

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功
400	Bad Request	リクエストの状態は「却下」、「キャン セル」、「エラー」である。リク エストが承認中である場合
404	Not Found	指定したリクエストが存在しない 場合

## レスポンスボディ

JSON キー	説明
status	リクエストの状態 ※1

※1 リクエストの状態: 1. DOING

## 実行例

### 【リクエスト】

```
PUT /cloudportal/api/v1.0/requests/R00000001/execute HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "status": "DOING"
}
```

## 2.4.5 リクエストキャンセル

指定したリクエストをキャンセルします。

### URL

```
PUT /cloudportal/api/v1.0/requests/{requestId}/cancel
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_REQUEST_CANCEL の権限を持つこと

### リクエストパラメータ

リクエストパラメータ名	省略可否	説明
requestId	不可	キャンセルしたいリクエストのリクエスト ID

### リクエストボディ

なし

### レスポンスコード

コード	意味	説明
200	OK	成功
400	Bad Request	リクエストの状態は「承認待ち」、「実行待ち」、「エラー」以外の状態である場合
404	Not Found	指定したリクエストが存在しない場合

### レスポンスボディ

JSON キー	説明
status	リクエストの状態 ※1

※1 リクエストの状態: 1. CANCELED

## 実行例

### 【リクエスト】

```
PUT /cloudportal/api/v1.0/requests/R00000001/cancel HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "status": "CANCELED"
}
```

## 2.4.6 リクエスト却下

指定したリクエストを却下します。

リクエストの状態が「承認待ち」または「実行待ち」の場合、リクエスト却下が可能です。

## URL

```
PUT /cloudportal/api/v1.0/requests/{requestId}/reject
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_REQUEST_REJECT または ROLE_REQUEST_AGENCY_REJECT の権限を持つこと

## リクエストパラメータ

リクエストパラメータ名	省略可否	説明
requestId	不可	却下したいリクエストのリクエスト ID



## リクエストボディ

JSON キー	省略可否	説明
admitComment	可	申請者へのコメント

## レスポンスコード

コード	意味	説明
200	OK	成功
400	Bad Request	リクエストの状態が「承認待ち」、「実行待ち」以外の場合
404	Not Found	指定したリクエストが存在しない場合

## レスポンスボディ

JSON キー	説明
status	リクエストの状態 ※1

※1 リクエストの状態: 1. REJECT

## 実行例

### 【リクエスト】

```
PUT /cloudportal/api/v1.0/requests/R00000001/reject HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
{
  "admitComment": "申請したリソースが不正です。却下します。"
}
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "status": "REJECT"
}
```

## 2.4.7 サーバ作成リクエストの作成

リクエスト作成 API では、リクエスト種別が「サーバ作成」、「サーバ削除」であるリクエストの作成をサポートしています。

本 API の実行にあたり、システム管理者は予めサーバ作成リクエストの雛形となる情報をファイルに記述し、所定のフォルダに配置しておく必要があります。本書ではこれをリク

エストテンプレートと表現します。APIの実行者はリクエストテンプレートを指定し、テンプレートに記載されていない必要なパラメータを補完するかたちでAPIを実行します。

パブリッククラウド(AWS/Azure)はリクエスト作成APIに対応していません。

リクエストテンプレートはJSONフォーマットで作成します。文字コードはUTF-8を指定し、拡張子がjsonとなるファイルを以下のディレクトリ配下に作成してください。ファイル名は任意です。

```
C:\Program Files (x86)\NEC\vDCA\MoM\FW\Tomcat\conf\NEC\requestTemplates
```

## ヒント

- ファイルパスの"requestTemplates"ディレクトリ配下に、任意のサブディレクトリを作ることが可能です。
- ファイルパスの"requestTemplates"ディレクトリ配下に、複数のリクエストテンプレートファイルを作ることが可能です。
- リクエストテンプレートは、プロパティ"name"の値で識別します。複数のリクエストテンプレートファイルを配置する場合、重複しない値を設定してください。
- リクエストテンプレートは、保存後、画面更新で反映されます。サービス「Apache Tomcat 8.5 ServiceGovernor」再起動は不要です。

リクエストテンプレートにて指定可能なキーは以下の通りです。本書ではJSONキーを.(ピリオド)で連結した形で表現します。

例えば、以下の request.info.templateName キーの値を確認・変更したい場合

キー	型	省略可否	説明
request.info	object	不可	リクエスト詳細情報
request.info.templateName	string	不可	作成元のVMテンプレート名

以下のようにファイル上の位置を読み替えて設定を確認・変更してください。

```
{
  "name": "templetel",
  "request" {
    "info" : {
      "templateName": "RHEL7",
      (略)
    }
  }
}
```

キー	型	省略可否	説明
name	string	不可	リクエストテンプレート名 他のリクエストテンプレートと重複がない一意IDを指定
tenants	string	可	リクエストテンプレートを利用できるテナント 複数のテナントに公開する、特定のテナントのみ公開しないといった設定も可能

キー	型	省略可否	説明
			省略した場合、全テナント利用可能 記述ルールについては『vDC Automation Standard Edition ポータル利用者マニュアル(リソース管理編) カタログの公開制限設定』を参照
request	object	不可	リクエストテンプレートのパラメータ
request.info	object	不可	リクエスト詳細情報
request.info.templateName	string	不可	作成元の VM テンプレート名 SigmaSystemCenter に登録されている VM テンプレートの名前を指定
request.info.resourcePool	string	不可	作成先のリソースプール名 サブリソースプールを指定したい場合には、 $\${ルールリソースプール名}/\${サブリソースプール名}$ で指定 専有リソースプールを指定した場合、併せて公開制限設定を行う 記述ルールについては『vDC Automation Standard Edition ポータル利用者マニュアル(リソース管理編) カタログの公開制限設定』を参照
request.info.specName	string	不可	スペック名 C:\Program Files (x86)\NEC\vDCA\MoM\FW\Tomcat\conf\NEC\webframework.properties に記述されている VM テンプレートにて選択可能なスペックの名前を指定
request.info.groupPath	string	可	作成先の業務グループ $/\$(グループ)$ を設定。 省略時には $\${テナント ID}/\_default$ が設定される
request.hardware	object	可 *1	申請のマシン情報
request.hardware.disks[]	object	可 *1	ディスク設定の配列
request.hardware.disks[n].type	string	不可	ディスクの種類(systemdisk   extendeddisk)
request.hardware.disks[n].diskSize	number	可 *2	ディスクサイズ(GB)指定
request.hardware.disks[n].tag	string	可	ディスクタグ 省略時にはディスクタグのデフォルトパラメータが設定される 利用できないディスクタグを指定された場合はディスクタグのデフォルトパラメータが設定される
request.networks[]	object	不可 *7 *8	ネットワーク設定の配列
request.networks[n].name	string	不可	論理ネットワーク名
request.networks[n].networkProfile	object	可	DNS/WINS 情報
request.networks[n].networkProfile.primaryDNS	string	可	プライマリ DNS
request.networks[n].networkProfile.secondaryDNS	string	可	セカンダリ DNS

キー	型	省略可否	説明
request.networks[n].networkProfile.tertiaryDNS *4	string	可	ターシャリ DNS
request.networks[n].networkProfile.primaryWINS *3	string	可	プライマリ WINS
request.networks[n].networkProfile.secondaryWINS *3	string	可	セカンダリ WINS
request.hostProfile	object	不可	ホスト設定
request.hostProfile.account	string	可 *5	管理者アカウント
request.hostProfile.password	string	不可	管理者パスワード
request.hostProfile.organization	string	不可	OS に設定するユーザの所属 *3
request.hostProfile.timezone *3	string	不可	OS のタイムゾーン 以下のサイトに記載されている値を設定 <a href="https://support.microsoft.com/en-us/help/973627/microsoft-time-zone-index-values">https://support.microsoft.com/en-us/help/973627/microsoft-time-zone-index-values</a>
request.hostProfile.productId *3	string	可	OS のプロダクトキー
request.hostProfile.domainType *3	string	不可	ワークグループまたはドメインを指定 • WORKGROUP • DOMAIN
request.hostProfile.networkName *3	string	不可	所属するワークグループ名またはドメイン名
request.hostProfile.domainAccount *3	string	可 *6	ドメインアカウント
request.hostProfile.domainPassword *3	string	可 *6	ドメインパスワード
request.hostProfile.domainSuffix *4	string	可	ドメインサフィックス
request.extendedParams	object	可	拡張リクエストパラメータ設定
request.extendedParams[n].name	string	不可	拡張リクエストパラメータ表示名 拡張リクエストパラメータを設定する時省略不可
request.extendedParams[n].parameterKey	string	不可	作成拡張リクエストパラメータのキー 拡張リクエストパラメータを設定する時省略不可
request.extendedParams[n].parameterValue	string	不可	作成拡張リクエストパラメータの値 拡張リクエストパラメータを設定する時省略不可

## ヒント

- \*1 省略時にはシステムディスクのみが設定されます。システムディスクのサイズはスペックで設定された値が採用されます。
- \*2 hardware.disks[n].type に systemdisk を設定した場合、diskSize は省略可能です。diskSize にはスペックで設定された値が採用されます。
- \*3 VM テンプレートの OS タイプが WindowsClient または WindowsServer の場合のみ有効です。

- \*4 VM テンプレートの OS タイプが Linux の場合のみ有効です。
- \*5 省略時には VM テンプレートの OS タイプが WindowsClient または WindowsServer の場合は Administrator, Linux の場合は root が設定されます。
- \*6 hostProfile.domainType が domain の場合のみ必須です。

---

## 注

- \*7 networks[]の先頭には必ずテナント管理 LAN を設定してください。
  - \*8 接続される論理ネットワークの IP アドレスは自動設定のみをサポートします。
- 

以下、リクエストテンプレートの設定例です。

```
{
  "name": "commandServer",
  "tenants": [
    "TenantA",
    "TenantB"
  ],
  "request": {
    "info": {
      "templateName": "templateA",
      "osName": "Red Hat Enterprise Linux 6 (64bit)",
      "resourcePool": "RP/Sub_TenantA",
      "groupPath": "/bbb",
      "specName": "specName001"
    },
    "hardware": {
      "disks": [
        {
          "type": "systemdisk",
          "diskSize": "10240"
        },
        {
          "type": "extendeddisk",
          "diskSize": "1024"
        }
      ]
    },
    "networks": [
      {
        "name": "manegementLAN",
        "networkProfile": {
          "primaryDNS": "192.168.1.1",
          "secondaryDNS": "192.168.1.2",
          "tertiaryDNS": "192.168.1.3",
          "primaryWINS": "192.168.1.4",
          "secondaryWINS": "192.168.1.5"
        }
      },
      {
        "name": "manegementLAN"
      }
    ]
  }
}
```

```

    "hostProfile": {
      "organization": "aaa",
      "productKey": "bbb",
      "domainType": "DOMAIN",
      "networkName": "ccc",
      "domainAccount": "ddd",
      "domainPassword": "eee",
      "timezone": "80000050",
      "account": "root",
      "password": "vdc",
      "domainSuffix": "domain"
    },
    "extendedParams": [
      {
        "name": "VmCreateKey01",
        "parameterKey": "param01",
        "parameterValue": "aaa"
      }
    ]
  }
}

```

## URL

```
POST /cloudportal/api/v1.0/requests/server/create/{templateName}
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_REQUEST_CREATE および ROLE_REQUEST_SERVER_CREATE の権限を持つこと

## リクエストパラメータ

リクエストパラメータ名	省略可否	説明
templateName	不可	リクエストテンプレート名

## リクエストボディ

キー	省略可否	説明
applyComment	可	申請コメント

キー	省略可否	説明
isAutoExecute	不可	リクエスト実行種別(自動:true/手動:false)
serverName	不可	サーバ名
comment	可	サーバ備考

## レスポンスコード

コード	意味	説明
201	Created	作成成功
400	Bad Request	リクエスト情報不正
404	Not Found	API 利用者ユーザが存在しない リクエストテンプレート定義ファイルが存在しない 指定されたネットワークを取得する際に、異常が発生する場合
409	Conflict	同一テナント内に当該リクエスト申請のサーバ名と同じ名前のサーバが既に存在している

## レスポンスボディ

JSON キー	説明
requestId	作成したリクエストの ID

## 実行例

### 【リクエスト】

```
POST
/cloudportal/api/v1.0/requests/server/create/requestTemplate HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
{
  "applyComment": "XX用サーバを申請します。",
  "isAutoExecute": true,
  "serverName": "server001",
  "comment": "テストサーバ"
}
```

### 【レスポンス】

```
HTTP/1.1 201 Created
{
  "requestId" : "R00000010"
}
```

## 2.4.8 サーバ削除リクエストの作成

指定したサーバの削除リクエストを作成する

### URL

```
POST /cloudportal/api/v1.0/requests/server/delete/{serverId}
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列)

### リクエストパラメータ

リクエストパラメータ	省略可否	説明
serverId	不可	削除したいサーバのサーバ ID

### リクエストボディ

キー	型	省略可否	説明
applyComment	string	可	申請コメント
isAutoExecute	boolean	不可	リクエスト実行種別 <ul style="list-style-type: none"> <li>• true : 自動実行</li> <li>• false : 手動実行</li> </ul>

### レスポンスコード

コード	意味	説明
200	OK	削除成功
400	Bad Request	サーバ状態は停止以外状態
404	Not Found	指定したサーバ ID が存在しない場合
409	Conflict	指定したサーバは SigmaSystemCenter から削除済み



## レスポンスボディ

JSON キー	説明
requestId	サーバ削除リクエスト ID

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v1.0/requests/server/delete/S00000001 HTTP/1.0
Content-Type: application/json; charset=utf-8
{
  "applyComment" : "テストサーバを削除お願いします",
  "isAutoExecute" : true
}
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "requestId" : "R00000001"
}
```

## 2.5 サーバ API

サーバ API について説明します。

### 【権限】

サーバ API に必要とする権限は以下です。

API 機能	権限	説明
サーバー一覧の表示	ROLE_SERVER_LIST_SHOW	サーバー一覧情報の取得と検索
サーバ詳細の表示	ROLE_SERVER_DETAIL_SHOW	サーバ詳細情報の取得
サーバ起動	ROLE_SERVER_STARTUP	サーバの起動
サーバ停止	ROLE_SERVER_SHUTDOWN	サーバの停止
サーバ電源 OFF	ROLE_SERVER_POWEROFF	サーバの電源 OFF
サーバ再起動	ROLE_SERVER_REBOOT	サーバの再起動
サーバ同期	ROLE_SERVER_SYNCHRONIZE	サーバの同期

### 2.5.1 サーバー一覧取得

サーバー一覧を取得します。

## URL

```
GET /cloudportal/api/v1.0/servers
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SERVER_LIST_SHOW の権限を持つこと

### リクエストパラメータ

なし

### リクエストボディ

なし

### レスポンスコード

コード	意味	説明
200	OK	取得成功

### レスポンスボディ

JSON キー	説明
tenantId	テナント ID
serverId	サーバ ID
groupName	グループ名
serverName	サーバ名
location	ロケーション
status	サーバのステータス <ul style="list-style-type: none"> <li>• UNKNOWN 説明: サーバ状態不明</li> <li>• STARTED 説明: サーバ状態起動</li> <li>• START 説明: サーバ状態起動中</li> <li>• OFF</li> </ul>

JSON キー	説明
	説明：サーバ状態停止 ・ SHUTOFF 説明：サーバ状態停止中 ・ POWEROFF 説明：サーバ状態強制電源オフ中 ・ RESTART 説明：サーバ状態再起動中 ・ SUSPENDED 説明：サーバ状態サスペンド ・ SHELVED 説明：サーバ状態休止中 ・ DELETED 説明：サーバ状態削除済み ・ ERROR 説明：サーバ状態エラー
progress	進捗率
osName	OS 名
resourcePoolName	リソースプール名
comment	備考

## 注

API を実行したユーザが所属するテナントのサーバのみ返却されます。

API を実行したユーザがシステム管理者の場合、すべてのテナントが対象となります。

## 実行例

### 【リクエスト】

```
GET /cloudportal/api/v1.0/servers HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.0 200 OK
Content-Type: application/json; charset=utf-8
{
  "servers" : [
    {
      "tenantId": "TenantA",
      "serverId": "S000000001",
      "groupName": "",
      "serverName": "Win2012R2-01",
      "location": "Private",
      "status": "STARTED",
      "progress": 100,
      "osName": "Microsoft Windows Server 2012 R2 (64 ビット)",
```

```

    "resourcePoolName": "RP_TenantA",
    "remarks": ""
  },
  {
    "tenantId": "TenantA",
    "serverId": "S00000002",
    "groupName": "windowsGroup",
    "serverName": "Win2012R2-02",
    "location": "Private",
    "status": "STARTED",
    "progress": 100,
    "osName": "Microsoft Windows Server 2012 R2 (64 ビット)",
    "resourcePoolName": "RP_TenantA",
    "remarks": "テスト用サーバ。"
  }
]
}

```

## 2.5.2 サーバ詳細情報取得

指定したサーバ詳細情報を取得する

### URL

```
GET /cloudportal/api/v1.0/servers/{serverId}
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SERVER_DETAIL_SHOW の権限を持つこと

### リクエストパラメータ

リクエストパラメータ名	省略可否	説明
serverId	不可	サーバ ID

### リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功
404	Not Found	指定したサーバ ID が存在しない場合

## レスポンスボディ

JSON キー	説明
server.tenantId	テナント ID
server.serverId	サーバ ID
server.groupName	グループ名
server.serverName	サーバ名
server.location	ロケーション
server.status	状態 <ul style="list-style-type: none"> <li>• UNKNOWN</li> <li>• STARTED</li> <li>• START</li> <li>• OFF</li> <li>• SHUTDOWN</li> <li>• POWEROFF</li> <li>• RESTART</li> <li>• SUSPENDED</li> <li>• SHELVED</li> <li>• DELETED</li> <li>• ERROR</li> </ul>
server.progress	進捗率
server.osName	OS 名
server.resourcePool	リソースプール
server.comment	コメント
server.cpuCount	CPU 数
server.memorySize	メモリサイズ(MB)
server.resourcePoolPath	リソースプールパス
server.groupPath	業務グループパス
server.uuId	サーバ UUID
server.machineType	マシン種別
server.remarks	備考
server.requestCompleteDatetime	リクエスト完了日時(yyyy/MM/dd HH:mm:ss) タイムゾーン:API 利用ユーザのタイムゾーン 取得できない場合には default timezone を利用する
server.additionalInfo	付加情報

JSON キー	説明
server.displayTemplateName	テンプレート名
server.disk	サーバディスクオブジェクト
server.disk[n].deviceSlot	ディスクスロット
server.disk[n].type	タイプ
server.disk[n].diskType	ディスクタイプ
server.disk[n].name	ディスクファイル名
server.disk[n].diskSize	ディスク総量(KB)
server.network	サーバネットワークオブジェクト
server.network[n].interFace	インターフェース名
server.network[n].ipAddress	IP アドレス
server.network[n].macAddress	MAC アドレス

## 実行例

### 【リクエスト】

```
GET /cloudportal/api/v1.0/servers/S00000001 HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.0 200 OK
Content-Type: application/json; charset=utf-8
{
  "server": {
    "tenantId": null,
    "serverId": null,
    "groupName": null,
    "serverName": "ServerE",
    "location": "Private",
    "status": "STARTED",
    "progress": null,
    "osName": "Red Hat Enterprise Linux 6 (64ビット)",
    "resourcePool": null,
    "comment": null,
    "cpuCount": 1,
    "memorySize": "1024",
    "resourcePoolPath": "/RP/Sub_TenantA",
    "groupPath": "/bbb",
    "uuId": "42231e8d-263f-1a21-1709-971be07319f7",
    "machineType": "VM",
    "remarks": "APIで作成したサーバ",
    "requestCompleteDatetime": "2020/03/09 11:29:39",
    "additionalInfo": "",
    "displayTemplateName": "rhel-template",
    "disk": [
      {
        "deviceSlot": "SCSI0:0",
        "type": "systemdisk",
```

```

        "diskType": "thin",
        "name": "[portal_ds1] ServerE/ServerE_2.vmdk",
        "diskSize": "16777216"
    },
    {
        "deviceSlot": "SCSI0:1",
        "type": "extendeddisk",
        "diskType": "thin",
        "name": "[portal_ds1] ServerE/ServerE.vmdk",
        "diskSize": "1048576"
    }
],
"network": [
    {
        "interFace": "manegementLAN",
        "ipAddress": "192.168.24.140",
        "macAddress": "00:50:56:A3:8E:CF"
    },
    {
        "interFace": "manegementLAN2",
        "ipAddress": "192.168.24.150",
        "macAddress": "00:50:56:A3:9E:0D"
    }
]
}
}
}

```

### 2.5.3 サーバ起動

サーバを起動します。

#### URL

```
POST /cloudportal/api/v1.0/servers/{serverId}/startup
```

#### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SERVER_STARTUP の権限を持つこと

#### リクエストパラメータ

リクエストパラメータ名	省略可否	説明
serverId	不可	サーバ ID

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功
400	Not Startup	指定されたサーバは指定された電源操作できる状態ではありません
404	Not Found	指定サーバが存在しない

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v1.0/server/S00000001/startup HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.0 200 OK
```

## 2.5.4 サーバ停止

サーバを停止します。

## URL

```
POST /cloudportal/api/v1.0/servers/{serverId}/shutdown
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列)



リクエストヘッダ名	省略可否	説明
		※API 利用者は ROLE_SERVER_SHUTDOWN の 権限を持つこと

## リクエストパラメータ

リクエストパラメータ名	省略可否	説明
serverId	不可	サーバ ID

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功
400	Not Stop	指定されたサーバは指定された電源操作ができる状態ではありません
404	Not Found	指定サーバが存在しない

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v1.0/servers/S00000001/shutdown HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.0 200 OK
```

## 2.5.5 サーバ電源 OFF

サーバの電源を OFF にします。

## URL

```
POST /cloudportal/api/v1.0/servers/{serverId}/poweroff
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SERVER_POWEROFF の権限を持つこと

## リクエストパラメータ

リクエストパラメータ名	省略可否	説明
serverId	不可	サーバ ID

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功
400	Not Power Off	指定されたサーバは指定された電源操作ができる状態ではありません
404	Not Found	指定サーバが存在しない

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v1.0/servers/S00000001/poweroff HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

## 【レスポンス】

```
HTTP/1.0 200 OK
```

## 2.5.6 サーバ再起動

サーバを再起動します。

### URL

```
POST /cloudportal/api/v1.0/servers/{serverId}/reboot
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SERVER_REBOOT の権限を持つこと

### リクエストパラメータ

リクエストパラメータ名	省略可否	説明
serverId	不可	サーバ ID

### リクエストボディ

なし

### レスポンスコード

コード	意味	説明
200	OK	成功
400	Not Reboot	指定されたサーバは指定された電源操作ができる状態ではありません
404	Not Found	指定サーバが存在しない

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v1.0/servers/S00000001/reboot HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.0 200 OK
```

## 2.5.7 サーバ同期

サーバを同期します。

## URL

```
POST /cloudportal/api/v1.0/servers/synchronize
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SERVER_SYNCHRONIZE の権限を持つこと

## リクエストパラメータ

なし

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功
404	Not Found	利用者ユーザが存在しない場合

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v1.0/servers/synchronize HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.0 200 OK
```

## 2.6 スコープ管理 API

スコープ管理 API について説明します。

### 【権限】

スコープ管理に必要とする権限は以下です。

API 機能	権限	説明
スコープ一覧情報の取得	ROLE_SCOPE_SHOW	スコープ一覧情報の取得
スコープ一括追加	ROLE_SCOPE_CREATE	スコープ一括追加
スコープ一括更新	ROLE_SCOPE_UPDATE	スコープ一括更新
スコープ一括削除	ROLE_SCOPE_DELETE	スコープ一括削除

### 2.6.1 スコープ一覧取得(v1.0)

指定されたプロダクト名、ファンクション名配下にスコープの一覧を取得する

## URL

```
GET /cloudportal/api/v1.0/scopes/{product}/{function}
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPE_SHOW の権限を持つこと

## リクエストパラメータ

リクエストパラメータ	省略可否	説明
product	不可	スコープが所属するプロダクト名
function	不可	スコープが所属するプロダクト内のファンクション名

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	スコープの一覧取得が完了
401	Unauthorized	指定した API キーのユーザが存在しない
403	Forbidden	API キーから識別されるユーザは ROLE_SCOPE_SHOW 権限を有していない
404	Not Found	プロダクトまたはファンクションが存在しない
503	Service Unavailable	スコープの同期が行われていない

## レスポンスボディ

JSON キー	説明
scopes.id	スコープの ID
scopes.name	スコープの表示名
scopes.viewType	スコープツリー上のツリーアイテムに付与される CSS クラス
scopes.parent	親となるスコープの ID
scopes.tenantId	スコープが所属するテナントのテナント ID

## 実行例

### 【リクエスト】

```
GET /cloudportal/api/v1.0/scopes/vDCA/Server HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "scopes": [
    {
      "id": "msc_status_0000000000",
      "name": "groupA",
      "viewType": "SysMgrG_STS_GROUP",
      "parent": 0,
      "tenantId": "DefaultTenant"
    }
  ]
}
```

## 2.6.2 スコープ一括追加(v1.0)

指定されたプロダクト名、ファンクション名配下にスコープを一括追加します。

指定されたスコープのパラメータに一つでもエラーがある場合には、スコープの登録は一切行われません。スコープを同期するオプションが指定された場合には、プロダクト、ファンクション配下のスコープはすべて破棄され、指定されたスコープを一括登録します(同期します)。

## URL

```
POST /cloudportal/api/v1.0/scopes/{product}/{function}
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPE_CREATE の権限を持つこと

## リクエストパラメータ

リクエストパラメータ	省略可否	説明
product	不可	スコープが所属するプロダクト名
function	不可	スコープが所属するプロダクト内のファンクション名

## リクエストボディ

キー	型	省略可否	説明
scopes.id	string	不可	スコープの ID
scopes.name	string	可	スコープの表示名 以下の場合、未設定として扱う。 <ul style="list-style-type: none"> <li>・ キーの指定がない</li> <li>・ null</li> <li>・ 値が空文字</li> </ul>
scopes.viewType	string	可	スコープツリー上のツリーアイテムに付与される CSS クラス ツリーアイテムのアイコンを変更したい場合に指定 以下の場合、未設定として扱う。 <ul style="list-style-type: none"> <li>・ キーの指定がない</li> <li>・ null</li> <li>・ 値が空文字</li> </ul>
scopes.parent	string	可	スコープの親の ID 以下の場合、最上位スコープ直下(親スコープ無し)に登録 <ul style="list-style-type: none"> <li>・ キーの指定がない</li> <li>・ null</li> <li>・ 値が空文字</li> </ul>
scopes.tenantId	string	不可	スコープのテナント ID
sync	boolean	可	スコープを同期するか否かを指定 <ul style="list-style-type: none"> <li>・ true: プロダクト、ファンクション配下のスコープはすべて破棄され、指定されたスコープを一括登録(同期される)</li> <li>・ false、null およびキー省略時: 指定されたスコープのみ追加</li> </ul>



## レスポンスコード

コード	意味	説明
201	Created	スコープの登録が完了した
400	Bad Request	リクエストボディのパラメータに不備がある
401	Unauthorized	指定した API キーのユーザが存在しない
403	Forbidden	API キーから識別されるユーザは <b>ROLE_SCOPE_CREATE</b> 権限を有していない
404	Not Found	プロダクトまたはファンクションが存在しない。リクエストボディのパラメータが存在しない。※ <b>sync</b> に <b>true</b> を指定していない場合は、スコープ構成に乖離が生じている可能性があるためスコープの同期を行うことを推奨する。
409	Conflict	指定したスコープ ID のスコープが既に存在する。登録対象のスコープが同一リクエスト内で複数指定されている。※ <b>sync</b> に <b>true</b> を指定していない場合は、スコープ構成に乖離が生じている可能性があるためスコープの同期を行うことを推奨する。
503	Service Unavailable	スコープの同期が行われていない。なお、 <b>sync</b> に <b>true</b> を指定した場合には発生しない。

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v1.0/scopes/vDCA/Server HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
{
  "scopes": [
    {
      "id": "msc_status_0000000000",
      "name": "groupA",
      "viewType": "SysMgrG_STS_GROUP",
      "parent": 0,
      "tenantId": "DefaultTenant"
    }
  ],
}
```

```
"sync": true
}
```

## 【レスポンス】

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
```

## ヒント

- 外部コンポーネントがスコープ管理 API を使用する為の設定

### 1. ビルトインユーザを定義する

ビルトインユーザを定義する方法について説明します。

- C:\Program Files (x86)\NEC\vDCA\MoM\FW\Tomcat\conf\NEC\custom 配下の「application.json」ファイルをテキストエディタで開き、ビルトインユーザを JSON フォーマットで定義します。

表 2-3 ビルトインユーザ設定項目

設定項目	型	設定内容
users[n].name	string	ユーザ名 ファイル中で一意となるユーザ名を入力してください。 <ul style="list-style-type: none"> <li>3～16 文字</li> <li>半角英数と_ (アンダースコア)、- (ハイフン) が使用可能</li> </ul>
users[n].password	string	パスワード パスワードを入力してください。 <ul style="list-style-type: none"> <li>6～16 文字</li> <li>半角英数字と!"#\$%&amp;'()*+,-./:;&lt;=&gt;?@[\]^_`{ }~が使用可能</li> </ul>
users[n].displayName	string	ユーザの表示名 ユーザの表示名を入力してください。 <ul style="list-style-type: none"> <li>1～1024 文字</li> </ul>
users[n].roles	string	ユーザが使用するロール定義 "ROLE_API_EXECUTER"を指定してください。 以下、スコープのビルトインユーザに必要な権限が割り当てられます。 <ul style="list-style-type: none"> <li>"ROLE_API_EXECUTER"</li> <li>"ROLE_SCOPE_SHOW"</li> <li>"ROLE_SCOPE_CREATE"</li> <li>"ROLE_SCOPE_UPDATE"</li> <li>"ROLE_SCOPE_DELETE"</li> </ul>

以下、ビルトインユーザの設定例を示します。

```
{
```

```
"name": "CloudPortal",
"displayNames" : {
  "ja" : "Cloud Portal",
  "en" : "Cloud Portal"
},
"version": "V1.0",
"description": "CloudPortal",
"authorities": [省略],
"functions" : [省略],
"roles" : [省略],
"users" : [
  {
    "name" : "apiuser",
    "password" : "apiuser",
    "displayName" : "API実行ユーザ",
    "roles" : ["ROLE_API_EXECUTER"]
  }
],
"extensions" : [省略],
"menus" : [省略],
"loginUserMenus": [省略]
}
```

- b. 設定ファイルの文字コードを UTF-8 で保存してください。
- c. アプリケーションサーバ(Tomcat)を再起動します。

サーバーマネージャーの左側のパネルからローカルサーバーを選択し、サービスから「Apache Tomcat 8.5 ServiceGovernor」を右クリックし、[サービスの再起動]を選択します。

ポータル起動時にビルトインユーザが作成されます

## 注

ビルトインユーザ設定前に、Active Directory 連携（LDAP 認証）設定を無効にしてください。

Active Directory 連携（LDAP 認証）設定が有効な場合、ユーザ追加に失敗する可能性があります。

## 2. ビルトインユーザの API キーを発行する

ビルトインユーザの API キーを発行する手順を説明します。

- a. システム管理者でポータルにログインします。
- b. メニューバーからテナント情報>ユーザ管理をクリックし、ユーザ管理画面に遷移します。
- c. 前手順で作成したビルトインユーザをユーザ一覧から選択し、ユーザ変更アイコンをクリックします。



- d. ユーザ変更(入力)画面で、API キーのラジオボタン「設定する」にチェックし、次へボタンをクリックします。



- e. ユーザ変更(確認)画面で、発行した API キーが表示されるので、スクリーンショットやメモ帳など忘れないように記録してください。



- f. ユーザ変更(確認)画面で、確定ボタンを押してください。

### 3. API キーを外部コンポーネントの設定ファイルに設定する

前手順で発行した API キーを外部コンポーネントの設定ファイルに設定してください。

設定ファイルの詳細は「WebSAM SystemManager G WebConsole Option 環境構築ガイド」を参照してください。

- スコープ一覧ツリーのアイコン変更

スコープ一覧ツリーのアイコンを変更する方法を説明します。

1. viewType の style を roleViewType.css に記載します。

デフォルトインストールパスは、`\webapps\portal\styles\cloudportal\roles\roleViewType.css` です。

2. スコープ登録 API でスコープを登録する際に” `viewType`” を指定します。

`viewType` で指定する `style` が `roleViewType.css` に存在しない場合、アイコンは表示されません。

`viewType` を指定しない場合、デフォルトアイコンが表示されます。

`roleViewType.css` 設定例 :

```
.viewType01 {
    background: url('../.../images/cloudportal/○○.svg');
}
.viewType02 {
    background: url('../.../images/cloudportal/××.svg');
}
.viewType05 {
    background: url('../.../images/cloudportal/O×.svg');
}
```

コマンド実行例 :

```
curl -i -XPOST http://localhost:12080/portal/cloudportal/api/v1.0/scopes/{product}/{function}
-H "Content-Type:application/json"
-H "apikey:○○"
-d "{
  \"scopes\": [
    {\"id\": \"Scope1\", \"name\": \"Scope1\", \"viewType\": \"viewType01\", \"parent\": \"\", \"tenantId\": \"TenantA\"},
    {\"id\": \"Scope2\", \"name\": \"Scope2\", \"viewType\": \"viewType02\", \"parent\": \"Scope1\", \"tenantId\": \"TenantA\"},
    {\"id\": \"Scope3\", \"name\": \"Scope3\", \"viewType\": \"viewType03\", \"parent\": \"Scope2\", \"tenantId\": \"TenantA\"},
    {\"id\": \"Scope4\", \"name\": \"Scope4\", \"viewType\": \"\", \"parent\": \"Scope3\", \"tenantId\": \"TenantA\"},
    {\"id\": \"Scope5\", \"name\": \"Scope5\", \"viewType\": \"viewType05\", \"parent\": \"Scope4\", \"tenantId\": \"TenantA\"}
  ], \"sync\": true
}"
```

	参照権	更新権	削除権
▲ 🏠 All	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ 📁 Scope1	⬇	⬇	<input type="checkbox"/>
▲ 📁 Scope2	⬇	⬇	<input type="checkbox"/>
▲ 📁 Scope3	⬇	⬇	<input type="checkbox"/>
▲ 📁 Scope4	⬇	⬇	<input type="checkbox"/>
📁 Scope5	⬇	⬇	<input type="checkbox"/>

図 2-1 アイコン変更前

	参照権	更新権	削除権
All	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Scope2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Scope3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Scope4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Scope5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

図 2-2 アイコン変更後

### 2.6.3 スコープ一括更新(v1.0)

指定されたプロダクト名、ファンクション名配下の指定されたスコープを一括更新します。指定されたスコープの指定されたキーのみ更新を行います。指定されていないキーの変更は行わず、キーの値が空文字の場合は挙動が異なる点について注意して利用してください。指定されたスコープのパラメータに一つでもエラーがある場合には、スコープの更新は一切行われません。レスポンスには変更が生じたスコープを返却します。

## URL

```
PUT /cloudportal/api/v1.0/scopes/{product}/{function}
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPE_UPDATE の権限を持つこと

## リクエストパラメータ

リクエストパラメータ	省略可否	説明
product	不可	スコープが所属するプロダクト名
function	不可	スコープが所属するプロダクト内のファンクション名

## リクエストボディ

キー	型	省略可否	説明
scopes.id	string	不可	スコープの ID
scopes.name	string	可	スコープの表示名 キーの指定がないまたは値が <code>null</code> の場合には変更は行われ ない。 値が空文字の場合には、表示名が未設定として更新される。
scopes.viewType	string	可	スコープツリー上のツリーアイテムに付与される CSS クラス ツリーアイテムのアイコンを変更したい場合に指定。 キーの指定がないまたは値が <code>null</code> の場合には変更は行われ ない。 値が空文字の場合には、CSS クラスは未設定として更新される。
scopes.parent	string	可	親となるスコープの ID キーの指定がないまたは値が <code>null</code> の場合には変更は行われ ない。 値が空文字の場合には、最上位スコープ直下(親スコープ無し)に更新される。

## レスポンスコード

コード	意味	説明
200	OK	スコープの更新が完了した
400	Bad Request	リクエストボディのパラメータに不備がある
401	Unauthorized	指定した API キーのユーザが存在しない
403	Forbidden	API キーから識別されるユーザは <code>ROLE_SCOPE_UPDATE</code> 権限を有していない
404	Not Found	プロダクトまたはファンクションが存在しない。リクエストボディのパラメータが存在しない。 ※スコープ構成に乖離が生じている可能性があるためスコープの同期を行うことを推奨する。

コード	意味	説明
409	Conflict	更新対象の範囲が同一リクエスト内で複数指定されている
503	Service Unavailable	範囲の同期が行われていない。

## レスポンスボディ

JSON キー	説明
scopes.id	範囲の ID
scopes.name	範囲の表示名
scopes.viewType	範囲ツリー上のツリーアイテムに付与される CSS クラス
scopes.parent	範囲の親の ID
scopes.tenantId	範囲のテナント ID

## 実行例

### 【リクエスト】

```
PUT /cloudportal/api/v1.0/scopes/vDCA/Server HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
{
  "scopes": [
    {
      "id": "msc_status_0000000000",
      "name": "groupA",
      "viewType": "SysMgrG_STS_GROUP",
      "parent": 0
    }
  ]
}
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "scopes": [
    {
      "id": "msc_status_0000000000",
      "name": "groupA",
      "viewType": "SysMgrG_STS_GROUP",
      "parent": 0,
      "tenantId": "DefaultTenant"
    }
  ]
}
```



## 2.6.4 スコープ一括削除(v1.0)

指定されたプロダクト名、ファンクション名配下の指定されたスコープを一括削除します。

スコープ削除 API では子スコープが存在する場合、削除できません。リクエストボディの `force` を `true` を設定することで指定されたスコープ配下の子スコープも同時に削除します。また、指定されたスコープを一つでも削除できない場合には、スコープの削除は一切行いません。レスポンスには削除されたスコープを返却します。存在しないスコープを指定された場合、404/410 は返却せず、200 として扱います。その場合、存在しないスコープ ID はレスポンスに含まれないことに留意してください。

### URL

```
DELETE /cloudportal/api/v1.0/scopes/{product}/{function}
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPE_DELETE の権限を持つこと

### リクエストパラメータ

リクエストパラメータ	省略可否	説明
product	不可	スコープが所属するプロダクト名
function	不可	スコープが所属するプロダクト内のファンクション名

### リクエストボディ

キー	型	省略可否	説明
ids[]	string[]	不可	スコープの ID の配列
force	boolean	可	削除対象スコープが子スコープを保持している場合、強制的に削除するかどうかを指定する true : 強制的に削除する false : 削除は行わず、レスポンスコード 400 を返却する

キー	型	省略可否	説明
			省略時：削除は行わず、レスポンスコード 400 を返却する

## レスポンスコード

コード	意味	説明
200	OK	スコープの削除が完了した
400	Bad Request	削除対象スコープが子スコープを保持しているため、削除できない
401	Unauthorized	指定した API キーのユーザが存在しない
403	Forbidden	API キーから識別されるユーザは <code>ROLE_SCOPE_DELETE</code> 権限を有していない
404	Not Found	プロダクトまたはファンクションが存在しない
503	Service Unavailable	スコープの同期が行われていない

## レスポンスボディ

JSON キー	説明
<code>ids[]</code>	スコープの ID の配列

## 実行例

### 【リクエスト】

```
DELETE /cloudportal/api/v1.0/scopes/vDCA/Server HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
{
  "ids": [
    "msc_status_0000000000"
  ],
  "force": true
}
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "ids": [
    "msc_status_0000000000"
  ]
}
```

```

]
}

```

## 2.6.5 スコープ一括追加(v2.0)

指定されたプロダクト名、ファンクション名配下にスコープを一括追加します。

指定されたスコープのパラメータに一つでもエラーがある場合には、スコープの登録は一切行われません。スコープを同期するオプションが指定された場合には、プロダクト、ファンクション配下のスコープはすべて破棄され、指定されたスコープを一括登録します(同期します)。

※スコープ ID は、v1.0 ではファンクションごとに一意、v2.0 ではファンクション内のテナントごとに一意となります。

## URL

```
POST /cloudportal/api/v2.0/scopes/{product}/{function}
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPE_CREATE の権限を持つこと

## リクエストパラメータ

リクエストパラメータ	省略可否	説明
product	不可	スコープが所属するプロダクト名
function	不可	スコープが所属するプロダクト内のファンクション名

## リクエストボディ

キー	型	省略可否	説明
scopes.id	string	不可	スコープの ID
scopes.name	string	可	スコープの表示名 以下の場合、未設定として扱う。 ・キーの指定がない

キー	型	省略可否	説明
			<ul style="list-style-type: none"> <li>• null</li> <li>• 値が空文字</li> </ul>
scopes.viewType	string	可	<p>スコープツリー上のツリーアイテムに付与される CSS クラス</p> <p>ツリーアイテムのアイコンを変更したい場合に指定</p> <p>以下の場合、未設定として扱う。</p> <ul style="list-style-type: none"> <li>• キーの指定がない</li> <li>• null</li> <li>• 値が空文字</li> </ul>
scopes.parent	string	可	<p>スコープの親の ID</p> <p>以下の場合、最上位スコープ直下(親スコープ無し)に登録</p> <ul style="list-style-type: none"> <li>• キーの指定がない</li> <li>• null</li> <li>• 値が空文字</li> </ul>
scopes.tenantId	string	不可	スコープのテナント ID
sync	boolean	可	<p>スコープを同期するか否かを指定</p> <ul style="list-style-type: none"> <li>• true: プロダクト、ファンクション配下のスコープはすべて破棄され、指定されたスコープを一括登録(同期される)</li> <li>• false、null およびキー省略時: 指定されたスコープのみ追加</li> </ul>

## レスポンスコード

コード	意味	説明
201	Created	スコープの登録が完了した
400	Bad Request	リクエストボディのパラメータに不備がある
401	Unauthorized	指定した API キーのユーザが存在しない
403	Forbidden	API キーから識別されるユーザは <code>ROLE_SCOPE_CREATE</code> 権限を有していない
404	Not Found	<p>プロダクトまたはファンクションが存在しない。リクエストボディのパラメータが存在しない。</p> <p>※sync に true を指定していない場合は、スコープ構成に乖離が生</p>

コード	意味	説明
		じている可能性があるためスコープの同期を行うことを推奨する。
409	Conflict	指定したスコープ ID のスコープが既に存在する。登録対象のスコープが同一リクエスト内で複数指定されている。※syncに true を指定していない場合は、スコープ構成に乖離が生じている可能性があるためスコープの同期を行うことを推奨する。
503	Service Unavailable	スコープの同期が行われていない。なお、sync に true を指定した場合には発生しない。

## レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v2.0/scopes/vDCA/Server HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
{
  "scopes": [
    {
      "id": "msc_status_0000000000",
      "name": "groupA",
      "viewType": "SysMgrG_STS_GROUP",
      "parent": 0,
      "tenantId": "DefaultTenant"
    }
  ],
  "sync": true
}
```

### 【レスポンス】

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
```

## ヒント

- 外部コンポーネントがスコープ管理 API を使用する為の設定
  - ビルトインユーザを定義する
 

ビルトインユーザを定義する方法について説明します。

- a. C:\Program Files (x86)\NEC\vDCA\MoM\FW\Tomcat\conf\NEC\custom 配下の「application.json」ファイルをテキストエディタで開き、ビルトインユーザを JSON フォーマットで定義します。

表 2-4 ビルトインユーザ設定項目

設定項目	型	設定内容
users[n].name	string	ユーザ名 ファイル中で一意となるユーザ名を入力してください。 <ul style="list-style-type: none"> <li>3~16 文字</li> <li>半角英数と_ (アンダースコア)、- (ハイフン) が使用可能</li> </ul>
users[n].password	string	パスワード パスワードを入力してください。 <ul style="list-style-type: none"> <li>6~16 文字</li> <li>半角英数字と!"#\$%&amp;'()*+,-./:;&lt;=&gt;?@[ ]^_`{ }~が使用可能</li> </ul>
users[n].displayName	string	ユーザの表示名 ユーザの表示名を入力してください。 <ul style="list-style-type: none"> <li>1~1024 文字</li> </ul>
users[n].roles	string	ユーザが使用するロール定義 "ROLE_API_EXECUTER"を指定してください。 以下、スコープのビルトインユーザに必要な権限が割り当てられます。 <ul style="list-style-type: none"> <li>"ROLE_API_EXECUTER"</li> <li>"ROLE_SCOPE_SHOW"</li> <li>"ROLE_SCOPE_CREATE"</li> <li>"ROLE_SCOPE_UPDATE"</li> <li>"ROLE_SCOPE_DELETE"</li> </ul>

以下、ビルトインユーザの設定例を示します。

```

{
  "name": "CloudPortal",
  "displayNames": {
    "ja": "Cloud Portal",
    "en": "Cloud Portal"
  },
  "version": "V1.0",
  "description": "CloudPortal",
  "authorities": [省略],
  "functions": [省略],
  "roles": [省略],
  "users": [
    {
      "name": "apiuser",
      "password": "apiuser",
      "displayName": "API実行ユーザ",
      "roles": ["ROLE_API_EXECUTER"]
    }
  ],
  "extensions": [省略],

```

```
"menus" : [省略],
"loginUserMenus": [省略]
}
```

- b. 設定ファイルの文字コードを UTF-8 で保存してください。
- c. アプリケーションサーバ(Tomcat)を再起動します。

サーバーマネージャーの左側のパネルからローカルサーバーを選択し、サービスから「Apache Tomcat 8.5 ServiceGovernor」を右クリックし、[サービスの再起動]を選択します。

ポータル起動時にビルトインユーザが作成されます

## 注

ビルトインユーザ設定前に、Active Directory 連携 (LDAP 認証) 設定を無効にしてください。

Active Directory 連携 (LDAP 認証) 設定が有効な場合、ユーザ追加に失敗する可能性があります。

## 2. ビルトインユーザの API キーを発行する

ビルトインユーザの API キーを発行する手順を説明します。

- a. システム管理者でポータルにログインします。
- b. メニューバーからテナント情報>ユーザ管理をクリックし、ユーザ管理画面に遷移します。
- c. 前手順で作成したビルトインユーザをユーザ一覧から選択し、ユーザ変更アイコンをクリックします。

テナント	ユーザID	氏名	ロール	状態
	admin	管理者	システム管理者	有効
	admin2	管理者2	システム管理者	有効
	admin_system1	admin_system1	システム管理者	有効
	admin_system2	admin_system2	システム管理者	有効
	apluser	API実行ユーザ	システム管理者	有効

API実行ユーザ のユーザ詳細

テナント: admin  
 ユーザID: apluser  
 氏名: API実行ユーザ  
 ロール: システム管理者, リクエスト承認不可  
 APIキー: 設定なし  
 状態: 有効  
 e-mail:

- d. ユーザ変更(入力)画面で、API キーのラジオボタン「設定する」にチェックし、次へボタンをクリックします。

- e. ユーザ変更(確認)画面で、発行した API キーが表示されるので、スクリーンショットやメモ帳など忘れないように記録してください。

- f. ユーザ変更(確認)画面で、確定ボタンを押してください。

### 3. API キーを外部コンポーネントの設定ファイルに設定する

前手順で発行した API キーを外部コンポーネントの設定ファイルに設定してください。

設定ファイルの詳細は「WebSAM SystemManager G WebConsole Option 環境構築ガイド」を参照してください。

- スコープ一覧ツリーのアイコン変更

スコープ一覧ツリーのアイコンを変更する方法を説明します。

1. viewType の style を roleViewType.css に記載します。

デフォルトインストールパスは、\webapps\portal\styles\cloudportal\roles\roleViewType.css です。

2. スコープ登録 API でスコープを登録する際に” viewType” を指定します。

viewType で指定する style が roleViewType.css に存在しない場合、アイコンは表示されません。

viewType を指定しない場合、デフォルトアイコンが表示されます。

roleViewType.css 設定例 :

```
.viewType01 {
    background: url('.././../images/cloudportal/○○.svg');
}
.viewType02 {
    background: url('.././../images/cloudportal/××.svg');
}
.viewType05 {
```



```
background: url('../../../../../images/cloudportal/Ox.svg');
}
```

コマンド実行例：

```
curl -i -XPOST http://localhost:12080/portal/cloudportal/api/v1.0/scopes/{product}/{function}
-H "Content-Type:application/json"
-H "apikey:〇〇"
-d "{
  \"scopes\": [
    {\"id\": \"Scope1\", \"name\": \"Scope1\", \"viewType\": \"viewType01\", \"parent\": \"\", \"tenantId\": \"TenantA\"},
    {\"id\": \"Scope2\", \"name\": \"Scope2\", \"viewType\": \"viewType02\", \"parent\": \"Scope1\", \"tenantId\": \"TenantA\"},
    {\"id\": \"Scope3\", \"name\": \"Scope3\", \"viewType\": \"viewType03\", \"parent\": \"Scope2\", \"tenantId\": \"TenantA\"},
    {\"id\": \"Scope4\", \"name\": \"Scope4\", \"viewType\": \"\", \"parent\": \"Scope3\", \"tenantId\": \"TenantA\"},
    {\"id\": \"Scope5\", \"name\": \"Scope5\", \"viewType\": \"viewType05\", \"parent\": \"Scope4\", \"tenantId\": \"TenantA\"}
  ], \"sync\": true
}"
```

	参照権	更新権	削除権
All	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

図 2-3 アイコン変更前

	参照権	更新権	削除権
All	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scope5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

図 2-4 アイコン変更後

## 2.6.6 スコープ一括更新(v2.0)

指定されたプロダクト名、ファンクション名配下の指定されたスコープを一括更新します。

指定されたスコープの指定されたキーのみ更新を行います。指定されていないキーの変更は行わず、キーの値が空文字の場合は挙動が異なる点について注意して利用してください。また、`scopes[].tenantId` キーは変更できないことに注意してください。指定されたスコープのパラメータに一つでもエラーがある場合には、スコープの更新は一切行いません。レスポンスには変更が生じたスコープを返却します。

※スコープ ID は、v1.0 ではファンクションごとに一意、v2.0 ではファンクション内のテナントごとに一意となります。

## URL

```
PUT /cloudportal/api/v2.0/scopes/{product}/{function}
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPE_UPDATE の権限を持つこと

## リクエストパラメータ

リクエストパラメータ	省略可否	説明
product	不可	スコープが所属するプロダクト名
function	不可	スコープが所属するプロダクト内のファンクション名

## リクエストボディ

キー	型	省略可否	説明
scopes.id	string	不可	スコープの ID
scopes.name	string	可	スコープの表示名 キーの指定がないまたは値が null の場合には変更は行われ ない。 値が空文字の場合には、表示名が未設定として更新される。
scopes.viewType	string	可	スコープツリー上のツリーアイテムに付与される CSS クラス

キー	型	省略可否	説明
			ツリーアイテムのアイコンを変更したい場合に指定。 キーの指定がないまたは値が <code>null</code> の場合には変更は行われず。 値が空文字の場合には、CSS クラスは未設定として更新される。
<code>scopes.parent</code>	<code>string</code>	可	親となるスコープの ID キーの指定がないまたは値が <code>null</code> の場合には変更は行われず。 値が空文字の場合には、最上位スコープ直下(親スコープ無し)に更新される。
<code>scopes.tenantId</code>	<code>string</code>	不可	スコープが所属するテナントのテナント ID

## レスポンスコード

コード	意味	説明
200	OK	スコープの更新が完了した
400	Bad Request	リクエストボディのパラメータに不備がある
401	Unauthorized	指定した API キーのユーザが存在しない
403	Forbidden	API キーから識別されるユーザは <code>ROLE_SCOPE_UPDATE</code> 権限を有していない
404	Not Found	プロダクトまたはファンクションが存在しない。リクエストボディのパラメータが存在しない。 ※スコープ構成に乖離が生じている可能性があるためスコープの同期を行うことを推奨する。
409	Conflict	更新対象のスコープが同一リクエスト内で複数指定されている
503	Service Unavailable	スコープの同期が行われていない。

## レスポンスボディ

JSON キー	説明
<code>scopes.id</code>	スコープの ID
<code>scopes.name</code>	スコープの表示名

JSON キー	説明
scopes.viewType	スコープツリー上のツリーアイテムに付与される CSS クラス
scopes.parent	親となるスコープの ID
scopes.tenantId	スコープが所属するテナントのテナント ID

## 実行例

### 【リクエスト】

```
PUT /cloudportal/api/v2.0/scopes/vDCA/Server HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
{
  "scopes": [
    {
      "id": "msc_status_0000000000",
      "name": "groupA",
      "viewType": "SysMgrG_STS_GROUP",
      "parent": 0,
      "tenantId": "DefaultTenant"
    }
  ]
}
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "scopes": [
    {
      "id": "msc_status_0000000000",
      "name": "groupA",
      "viewType": "SysMgrG_STS_GROUP",
      "parent": 0,
      "tenantId": "DefaultTenant"
    }
  ]
}
```

### 2.6.7 スコープ一括削除(v2.0)

指定されたプロダクト名、ファンクション名配下の指定されたスコープを一括削除します。

スコープ削除 API では子スコープが存在する場合、削除できません。リクエストボディの `force` で `true` を設定することで指定されたスコープ配下の子スコープも同時に削除します。また、指定されたスコープを一つでも削除できない場合には、スコープの削除は一切行いません。レスポンスには削除されたスコープを返却します。存在しないスコープを指定され

た場合、404/410 は返却せず、200 として扱います。その場合、存在しないスコープ ID はレスポンスに含まれないことに留意してください。

※スコープ ID は、v1.0 ではファンクションごとに一意、v2.0 ではファンクション内のテナントごとに一意となります。

## URL

```
DELETE /cloudportal/api/v2.0/scopes/{product}/{function}
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者は ROLE_SCOPE_DELETE の権限を持つこと

## リクエストパラメータ

リクエストパラメータ	省略可否	説明
product	不可	スコープが所属するプロダクト名
function	不可	スコープが所属するプロダクト内のファンクション名

## リクエストボディ

キー	型	省略可否	説明
scopes.id	string	不可	スコープの ID
scopes.tenantId	string	不可	スコープが所属するテナントのテナント ID
force	boolean	可	削除対象スコープが子スコープを保持している場合、強制的に削除するか否かを指定する true : 強制的に削除する false : 削除は行わず、レスポンスコード 400 を返却する 省略時 : 削除は行わず、レスポンスコード 400 を返却する

## レスポンスコード

コード	意味	説明
200	OK	スコープの削除が完了した
400	Bad Request	削除対象スコープが子スコープを保持しているため、削除できない
401	Unauthorized	指定した API キーのユーザが存在しない
403	Forbidden	API キーから識別されるユーザは <code>ROLE_SCOPE_DELETE</code> 権限を有していない
404	Not Found	プロダクトまたはファンクションが存在しない
503	Service Unavailable	スコープの同期が行われていない

## レスポンスボディ

JSON キー	説明
scopes.id	スコープの ID
scopes.tenantId	テナント ID

## 実行例

### 【リクエスト】

```
DELETE /cloudportal/api/v2.0/scopes/vDCA/Server HTTP/1.1
Content-Type: application/json; charset=utf-8
ApiKey: ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
{
  "scopes": [
    {
      "id": "msc_status_0000000000",
      "tenantId": "DefaultTenant"
    }
  ],
  "force": true
}
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "scopes": [
    {
      "id": "msc_status_0000000000",
      "tenantId": "DefaultTenant"
    }
  ]
}
```

```
}  
]  
}
```

## 2.7 ログインユーザ管理 API

ログインユーザ管理 API について説明します。

### 【権限】

ログインユーザ管理に必要とする権限はありません。

### 2.7.1 ユーザの詳細情報を取得する

ユーザが保持するプロダクト名、ファンクション名配下の権限、またはスコープとそれに割り当てられている権限の情報を一覧で返却します。

with に products を指定された場合、レスポンスに loginuser.products[] まで含めます。ただし、products 指定は本バージョンではサポートしません。

with に functions を指定された場合、レスポンスに loginuser.products[].functions[] まで含めます。

with に scopes を指定された場合、レスポンスに loginuser.products[].functions[].scopes[] まで含めます。

with の指定がない場合、loginuser.products[] を返却しません。

with に products, functions または scopes を指定した場合、products[].id が必須となります。

with に functions または scopes を指定した場合、products[].functions[].id が必須となります。

with に scopes を指定し、products[].functions[].scopes[] にフィルタリングしたいスコープ ID を指定することで、指定されたスコープに関する情報のみをフィルタリングして返却します。

with に function または scopes を指定し、products[].functions[].recursive に true を指定し、products[].functions[].scopes[] にフィルタリングしたいスコープ ID を指定することで、指定されたスコープ配下の子スコープについても親スコープとともに返却します。

また、products[].functions[].recursive についてパラメータを指定しないおよび false を指定した場合には、指定されたスコープに関する情報のみを返却します。

## URL

```
POST /cloudportal/api/v1.0/loginuser
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
X-Authorization	不可	認証トークン

リクエストヘッダで指定する X-Authorization は、「[2.8 ユーザ管理基盤 API \(87 ページ\)](#)」を参照して設定してください。

## リクエストパラメータ

なし

## リクエストボディ

キー	型	省略可否	説明
with	string	可	レスポンスに指定するパラメータを含めるか否かを指定する。 指定がない場合、loginuser.products[]は返却しない。 <ul style="list-style-type: none"> <li>products : レスポンスに loginuser.products[]まで含める。ただし、products 指定は本バージョンではサポートしない。</li> <li>functions : レスポンスに loginuser.products[].functions[]まで含める。</li> <li>scopes : レスポンスに loginuser.products[].functions[].scopes[]まで含める。</li> </ul>
products[].id	string	不可	付与する権限が所属するファクションが所属する製品の ID 以下の製品が指定可能 <ul style="list-style-type: none"> <li>•CloudPortal</li> <li>•vDCA</li> </ul> with に products, functions または、scopes を指定した場合、必須
products[].functions[].id	string	不可	付与する権限が所属するファクションの ID 以下のファクションが指定可能 <ul style="list-style-type: none"> <li>•CloudPortal 配下のファクションの場合、『利用者マニュアル(リソース管理編)』付録</li> </ul>



キー	型	省略可否	説明
			『CloudPortal プロダクトにて扱う機能および権限』の機能名を参照 <ul style="list-style-type: none"> <li>•vDCA 配下のファンクションの場合、『利用者マニュアル(リソース管理編)』付録『vDCA プロダクトにて扱う機能および権限』の機能名を参照</li> </ul> <b>with</b> に <b>products, functions</b> または、 <b>scopes</b> を指定した場合、必須
<b>products[].functions[].scopes[]</b>	string	可	フィルタリングするスコープ ID のリスト。 <b>with</b> に <b>scopes</b> を指定した場合のみ有効となる。
<b>products[].functions[].scopes[].id</b>	string	可	フィルタリングするスコープ ID。 <b>with</b> に <b>scopes</b> を指定した場合のみ有効となる。
<b>products[].functions[].recursive</b>	boolean	可	再帰的に取得するか否かを指定する。 <ul style="list-style-type: none"> <li>• <b>true</b> : フィルタリングするスコープ ID の子スコープも再帰的に取得して返却</li> <li>• <b>false</b> : 指定されたスコープに関する情報のみを返却</li> <li>• 指定なし : 指定されたスコープに関する情報のみを返却</li> </ul> <b>with</b> に <b>functions</b> または <b>scopes</b> を指定した場合、かつ、 <b>products[].functions[].scopes[].id</b> を指定した場合のみ有効となる

## レスポンスコード

コード	意味	説明
200	OK	ログインユーザ詳細の取得に成功
400	Bad Request	クエリーのパラメータに不備がある
401	Unauthorized	認証トークンが指定されていない。認証トークンが発行されていない。認証トークンの有効期限が切れている。

コード	意味	説明
503	Service Unavailable	スコープの同期が行われていない。なお、sync に true を指定した場合には発生しない。

## レスポンスボディ

JSON キー	説明
loginuser.tenantId	ユーザが所属するテナントのテナント ID
loginuser.userId	ユーザのユーザ ID
loginuser.userName	ユーザの氏名
loginuser.role	ユーザに割り当てられたロール
loginuser.approval	ユーザがリクエスト承認かどうか
loginuser.email	ユーザの Email アドレス
loginuser.remarks	ユーザの備考情報
loginuser.products[].id	プロダクトの ID
loginuser.products[].functions[].id	ファンクションの ID
loginuser.products[].functions[].scopes[].id	スコープ ID
loginuser.products[].functions[].scopes[].name	スコープの表示名
loginuser.products[].functions[].scopes[].viewType	スコープツリー上のツリーアイテムに付与される CSS クラス
loginuser.products[].functions[].scopes[].parent	親となるスコープの ID
loginuser.products[].functions[].scopes[].allows[]	スコープに付与する権限の一覧
loginuser.products[].functions[].allows[]	スコープに付与する権限の一覧

## 実行例

### 【リクエスト】

```
POST /cloudportal/api/v1.0/loginuser
X-Authorization:2281AAE154CBA1781ADE90D481AB6CB0
{
  "with": "scopes",
  "products": [
    {
      "id": "SysMgrG",
      "functions": [
        {
          "id": "status",
          "scopes": [
            "msc_status_0000000000"
          ]
        }
      ]
    }
  ]
}
```

```
        "recursive": true
      }
    ]
  }
]
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "loginuser": {
    "tenantId": "string",
    "userId": "string",
    "userName": "string",
    "role": "string",
    "approval": true,
    "email": "string",
    "remarks": "string",
    "products": [
      {
        "id": "SysMgrG",
        "functions": [
          {
            "id": "status",
            "scopes": [
              {
                "id": "msc_status_0000000000",
                "name": "groupA",
                "viewType": "SysMgrG_STS_GROUP",
                "parent": 0,
                "allows": [
                  "MSC_GENERAL_REFERENCE"
                ]
              }
            ],
            "allows": [
              "MSC_GENERAL_REFERENCE"
            ]
          }
        ]
      }
    ]
  }
}
```

## 2.8 ユーザ管理基盤 API

ユーザ管理基盤 API について説明します。

### 【権限】

ユーザ管理基盤に必要とする権限はありません。

## 2.8.1 ユーザ認証

ユーザ認証を行い、認証トークンを払い出します。

### URL

```
POST /session/new
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8

### リクエストパラメータ

なし

### リクエストボディ

キー	型	省略可否	説明
UserID	string	不可	認証を行うユーザのユーザ ID
Password	string	不可	認証を行うユーザのパスワード

### レスポンスコード

コード	意味	説明
200	OK	ユーザ認証成功
400	Bad Request	リクエストボディのパラメータに不備がある

### レスポンスヘッダ

レスポンスヘッダ名	説明
X-Authorization	token \${認証トークン}
Set-Cookie	JSESSIONID=\${認証トークン}

### レスポンスボディ

JSON キー	説明
user.id	ログインユーザの ID

JSON キー	説明
user.displayName	ログインユーザの表示名
user.mailAddress	ログインユーザのメールアドレス
user.tenant.id	ログインユーザの所属するテナントのテナント ID
user.tenant.displayName	ログインユーザの所属するテナントの表示名

## 実行例

### 【リクエスト】

```
POST /session/new HTTP/1.1
Content-Type: application/json; charset=utf-8
{
  "UserID": "string",
  "Password": "string"
}
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONID=98376C0C259B6FB6C109EE1B8D564524
X-Authorization: token 98376C0C259B6FB6C109EE1B8D564524
{
  "user": {
    "id": "string",
    "displayName": "string",
    "mailAddress": "string",
    "tenant": {
      "id": "string",
      "displayName": "string"
    }
  }
}
```

## 2.8.2 認証トークン検証

認証トークンの有効性を判定する。

### URL

```
GET /session/validate
```

## リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
X-Authorization	不可	認証トークン

## リクエストパラメータ

なし

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	認証トークンは有効である
401	Unauthorized	認証トークンは有効ではない

## レスポンスボディ

JSON キー	説明
user.id	ログインユーザの ID
user.displayName	ログインユーザの表示名
user.mailAddress	ログインユーザのメールアドレス
user.tenant.id	ログインユーザの所属するテナントのテナント ID
user.tenant.displayName	ログインユーザの所属するテナントの表示名

## 実行例

### 【リクエスト】

```
GET /session/validate HTTP/1.1
Content-Type: application/json; charset=utf-8
X-Authorization: token ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
```

### 【レスポンス】

```
HTTP/1.1 200 OK
{
```

```

"user": {
  "id": "string",
  "displayName": "string",
  "mailAddress": "string",
  "tenant": {
    "id": "string",
    "displayName": "string"
  }
}
}
}

```

### 2.8.3 認証トークン破棄

認証トークンを破棄する。

#### URL

```
DELETE /session/invalidate
```

#### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
X-Authorization	不可	認証トークン

#### リクエストパラメータ

なし

#### リクエストボディ

なし

#### レスポンスコード

コード	意味	説明
200	OK	認証トークン破棄に成功

#### レスポンスボディ

なし

## 実行例

### 【リクエスト】

```
DELETE /session/invalidate HTTP/1.1
X-Authorization: token ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
```

### 【レスポンス】

```
HTTP/1.1 200 OK
```

## 2.9 プラグインインスタンス API

プラグインインスタンス API について説明します。

### 【権限】

プラグインインスタンスに必要とする権限は以下です。

API 機能	権限	説明
プラグインインスタンス一覧表示	プラグイン定義ファイル(config.json)に記載された権限※	プラグインインスタンス一覧情報の取得と検索
プラグインインスタンス操作	プラグイン定義ファイル(config.json)に記載された権限※	プラグインインスタンスの操作

### ヒント

※プラグイン定義ファイルに記載された権限の詳細は、vDC Automation Standard Edition ServiceLinker オプションプラグイン開発・利用マニュアルの「プラグイン定義ディレクトリ・ファイルを作成する」を参照してください。

proxy 環境下でプラグインを利用する場合、以下、設定してください。

表 2-5 設定ファイル(webframework.properties)

キー	型	必須/任意	説明
product.cloudportal.provider.http.proxy.host	string	任意	HTTP のプロキシサーバ名または IP アドレス ワイルドカード指定不可
product.cloudportal.provider.http.proxy.port	string	任意	HTTP のプロキシサーバのポート ワイルドカード指定不可
product.cloudportal.provider.https.proxy.host	string	任意	HTTPS のプロキシサーバ名または IP アドレス ワイルドカード指定不可
product.cloudportal.provider.https.proxy.port	string	任意	HTTPS のプロキシサーバのポート ワイルドカード指定不可



キー	型	必須/任意	説明
product.cloudportal.provider.http.no.proxy	string	任意	指定されたアドレスは HTTP プロキシサーバを通さない 複数アドレスを設定する場合、「,」で区切る 例：192.168.1.1,172.28.1.1 ワイルドカード指定不可
product.cloudportal.provider.https.no.proxy	string	任意	指定されたアドレスは HTTPS プロキシサーバを通さない 複数アドレスを設定する場合、「,」で区切る 例：192.168.1.1,172.28.1.1 ワイルドカード指定不可

## 2.9.1 プラグインインスタンス一覧取得

プラグインインスタンス一覧を取得します。

### URL

```
GET /cloudportal/api/{version}/generalServices/{serviceName}
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者はプラグイン定義ファイル(config.json)に記載された権限を持つこと

### リクエストパラメータ

リクエストヘッダ名	省略可否	説明
version	不可	プラグインインスタンス一覧取得 API の API バージョン。「v1.0」を扱う
serviceName	不可	プラグイン名 (config.json のプロパティ「name」の値)

### リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功
400	Bad Request	<ul style="list-style-type: none"> <li>指定されたプラグイン種別の誤り</li> <li>指定されたプラグインのライセンスが無い</li> </ul>
404	Not Found	指定されたプラグインが見つからない

## レスポンスボディ

JSON キー	説明
tenantId	テナント ID
tenantName	テナント名
{インスタンスのプロパティ 1~n}	プラグインインスタンスのプロパティ値

## 実行例

### 【リクエスト】

```
GET /cloudportal/api/v1.0/generalServices/plugin1 HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
{
  "plugin1Instances" :
  [
    {
      "tenantId" : "tenA",
      "tenantName": "TenantA",
      "instanceId": "ocdb1",
      "displayName": "DBsystem1",
      "hostName": "host1",
      "shape": "VM.Standard2.4"
    },
    {
      "tenantId" : "tenA",
      "tenantName": "TenantA",
      "instanceId": "ocdb2",
      "displayName": "DBsystem2",
      "hostName": "host2",
```

```

    "shape": "BM.DenseIO1.36"
  },
  {
    "tenantId": "tenB",
    "tenantName": "TenantB",
    "instanceId": "ocdb3",
    "displayName": "DBsystem3",
    "hostName": "host3",
    "shape": "BM.DenseIO1.36"
  }
]
}

```

## 2.9.2 プラグインインスタンス操作

リクエストボディで指定された内容でプラグインインスタンスを操作します。

### URL

```
POST /cloudportal/api/{version}/generalServices/{serviceName}/{instanceId}
/{action}
```

### リクエストヘッダ

リクエストヘッダ名	省略可否	説明
Content-Type	不可	HTTP ヘッダの Content-Type:application/json;charset=utf-8
ApiKey	不可	API 利用者の API キー(自動生成された 32 文字の文字列) ※API 利用者はプラグイン定義ファイル(config.json)に記載された権限を持つこと

### リクエストパラメータ

リクエストヘッダ名	省略可否	説明
version	不可	プラグインインスタンス操作 API の API バージョン。「v1.0」を扱う
serviceName	不可	プラグインインスタンスのサービス名 (config.json のプロパティ「name」の値)
instanceId	不可	対象インスタンスのインスタンス ID
action	不可	操作種別 (config.json のリスト「operations」のプロパティ「name」の値)

## リクエストボディ

なし

## レスポンスコード

コード	意味	説明
200	OK	成功
400	Bad Request	<ul style="list-style-type: none"> <li>指定されたプラグイン種別の誤り</li> <li>指定されたプラグインインのライセンスが無い</li> <li>指定されたプラグインインスタンスの操作が見つからない</li> </ul>
404	Not Found	<ul style="list-style-type: none"> <li>指定されたプラグインインが見つからない</li> <li>指定されたプラグインインスタンスが見つからない</li> </ul>

## レスポンスボディ

JSON キー	説明
tenantId	テナント ID
tenantName	テナント名
{インスタンスのプロパティ 1~n}	プラグインインスタンスのプロパティ値

## 実行例

### 【リクエスト】

```
GET /cloudportal/api/v1.0/generalServices/plugin1 HTTP/1.0
Content-Type: application/json; charset=utf-8
ApiKey: 8A71C752DD816B0707D9DFD903495216
```

### 【レスポンス】

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "plugin1Instances" :
  [
    {
      "tenantId" : "tenA",
      "tenantName": "TenantA",
      "instanceId": "ocdb1",
```

```
"displayName": "DBsystem1",
"hostName": "host1",
"shape": "VM.Standard2.4"
},
{
  "tenantId" : "tenA",
  "tenantName": "TenantA",
  "instanceId": "ocdb2",
  "displayName": "DBsystem2",
  "hostName": "host2",
  "shape": "BM.DenseIO1.36"
},
{
  "tenantId" : "tenB",
  "tenantName": "TenantB",
  "instanceId": "ocdb3",
  "displayName": "DBsystem3",
  "hostName": "host3",
  "shape": "BM.DenseIO1.36"
}
]
}
```