



CHAPTER 6

REST API

- 「概要」 (P.6-1)
- 「REST API の呼び出し」 (P.6-9)
- 「詳細な API リファレンス」 (P.6-15)
- 「エラー メッセージ」 (P.6-49)
- 「クイック リファレンス」 (P.6-50)

概要

シスコは、Cisco Service Portal (Service Portal) に定義されているエンティティにアクセスするための、標準的な REST (Representational State Transfer) の API および Java スタブを一式提供しています。これらは、まとめて nsAPI と呼ばれます。

nsAPI では、組み込みのセッション サポートを使用して認証が行われます。エンティティに対するアクセス権限は、Service Portal アプリケーションでユーザに対して定義されたロールベース アクセス コントロール (RBAC) のオブジェクト レベル権限によって制御されます。

nsAPI を使用すると、外部アプリケーションおよび Portal Manager ソリューションとの統合が可能です。ポータル機能は、Java、JavaScript、HTML や Ext JS を使用して作成されたポートレットの設計およびレンダリング、つまり、ポータルの UI フレームワークをサポートしています。このようなポートレットの中で nsAPI を呼び出すことによって、必要なエンティティの情報を取得し、特定のタイプのエンティティのデータを更新可能にすることができます。ポータル モジュールの詳細については、『Cisco Service Portal Designer Guide』に説明があります。

サポートされるエンティティ

nsAPI でサポートされるエンティティは、次のカテゴリに分けられます。

エンティティ グループ	エンティティ タイプ
定義データ	カテゴリ
	サービス
	提供サービス
	エージェント

ディレクトリ データ	組織単位 個人 グループ アカウント
トランザクション データ	契約 要求 要求エントリ 承認 タスク
Lifecycle Center データ	サービス項目の詳細 すべてのサービス項目 標準
Portal Designer データ	コンテンツのカスタマイズ テーブル

操作

次のタイプの操作が、nsAPI でサポートされています。

- Service Portal アプリケーションに事前定義されているコア エンティティに対する HTTP GET 操作
- Lifecycle Center および Portal Designer のユーザ定義エンティティに対する HTTP GET 操作
- 次の特定のエンティティに対する HTTP POST 操作
 - 個人：個人の作成、個人の詳細情報の更新
 - タスク：タスクのアクション（完了/許可/拒否/確認）の実行

GET 操作とは、指定されたエンティティのデータを取得する操作であり、PUT 操作では、エンティティの詳細情報またはステータスに変更を加えることができます。操作を呼び出したユーザは、操作対象のエンティティ インスタンスに対して必要な読み取り権限または書き込み権限を持っている必要があります。

規約と構文

- REST の URL は次の規則に従います。

```
http(s)://<serverURL>/RequestCenter/nsapi/<entityGroup>/<entityType>/<filters>?sortBy=
<columnName>&sortDir=<sortOrder>?startRow=<x>&recordSize=<y>
```

山カッコ (<>) で囲まれている要素は、適切なパラメータまたはパラメータ値に置換する必要があります。

- フィルタ、ソート、およびページングのコントロールとアクションは、オプション パラメータとして REST URL に渡されます。フィルタには、「[フィルタ](#)」の項で説明するように、1 つまたは複数の式を含めることができます。フィルタを指定しなかった場合は、エンティティに関して見つかったすべてのインスタンスが返されます。

- 複数のフィルタの組み合わせが可能なこともあります。その場合、2 番め以降のパラメータの指定はオプションです。このようなパラメータについては、構文内でそのパラメータ構文を角カッコ ([]) で囲むことによって示しています。オプションのパラメータは、区切り文字 (|) で囲む必要があります。
- 特記のない限り、すべての URL で大文字と小文字が区別されます。
- nsAPI にはバージョン機能が組み込まれています。ベースバージョン 1.0 の REST URL にはバージョン番号を表すパラメータがありません。しかし、これは将来のリリースで変更される可能性があります。
- 要求された操作が成功したのか、それとも失敗したのかを示すために、HTTP のステータスコードおよびエラーメッセージが REST 応答で返されます。
- nsAPI Java バインディング パッケージの名前は com.newscale.nsapi.* です。

フィルタ

nsAPI は、次のように GET 操作においてエンティティのタイプに基づき多様なフィルタをサポートしています。

エンティティタイプ	使用可能なフィルタ/構文	REST URL の例
すべてのエンティティ	Id /id/<value>	http://serverURL/RequestCenter/nsapi/definition/servicedefs/id/16
	名前：完全一致 /name/<value>	http://serverURL/RequestCenter/nsapi/definition/servicedefs/name/Create%20Custom%20VM
	名前：ワイルドカード検索 ?name=<value>	http://serverURL/RequestCenter/nsapi/definition/servicedefs?name=Create%20Custom*
標準、サービス項目、カスタムコンテンツ	<p>すべてのテーブルのカラム。次の要素で構成されたフィルタ式を最大 3 つ使用可能</p> <ul style="list-style-type: none"> • 比較演算子：=、>、<、>=、<= • 関係演算子：AND、OR（大文字小文字の区別あり） • 区切り文字： <p>/<columnName1><operator1><value1>[<AND OR>]<columnName2><operator2><value2>][<AND OR> <columnName3><operator3><value3>]</p> <p>日付フィールドの値は mm-dd-yyyy 形式にする必要があります。</p>	<p>http://serverURL/RequestCenter/nsapi/standard/StOperatingSystem/Custom1=Linux AND Custom2=64</p> <p>これは、次のように記述することもできます。</p> <p>http://serverURL/RequestCenter/nsapi/standard/StOperatingSystem/Custom1=Linux Custom2=64</p> <p>http://serverURL/RequestCenter/nsapi/standard/StOperatingSystem/Custom1=Linux OR Custom2=64</p> <p>http://serverURL/RequestCenter/nsapi/serviceitems/serviceitemsubscription/SubmitDate=>03-01-2011</p> <p>http://serverURL/RequestCenter/nsapi/content/UcAnnouncementObj/Category=Corporate</p>

サービス項目	ビュー名 ?View Name=<value> 指定可能なビュー名は次のとおりです。 <i>My ServiceItems</i> (<i>My Services</i> に表示されるもの) <i>Manage ServiceItems</i> (<i>Service Item Manager</i> に表示されるもの)	http://serverURL/RequestCenter/nsapi/serviceitem/SiDesktop?View%20Name=My%20ServiceItems http://serverURL/RequestCenter/nsapi/serviceitem/SiDesktop?View%20Name=Manage%20ServiceItems
組織単位	OU のタイプ ?type=<all/businessUnit/serviceTeam>	http://serverURL/RequestCenter/nsapi/directory/organizationalunits?type=serviceTeam
個人	ログイン名 ?loginname/<value>	http://serverURL/RequestCenter/nsapi/directory/people?loginname/dsmith
	OU 名 ?ouname=<value>	http://serverURL/RequestCenter/nsapi/directory/people?ouname=Operations
	グループ名 ?groupname=<value>	http://serverURL/RequestCenter/nsapi/directory/people?groupname=Approvers
	ロール名 ?rolename=<value> 継承されたロールをフィルタに使用することはできません。	http://serverURL/RequestCenter/nsapi/directory/people?rolename=Service%20Performer
カテゴリ	カテゴリ タイプ ?catalogType=<serviceCatalog/offeringCatalog>	http://serverURL/RequestCenter/nsapi/definition/categories?catalogType=serviceCatalog
サービス	カテゴリ名 ?categoryName=<value>	http://serverURL/RequestCenter/nsapi/definition/servicedefs?categoryName=Manage%20Physical%20Servers
	キーワード ?keywordName=<value>	http://serverURL/RequestCenter/nsapi/definition/servicedefs?keywordName=server
要求および承認	ビュー名、ステータス /ViewName=<value1>[Status=<value2>] フィルタに使用可能なビュー名は、 <i>My Services</i> の <i>[Requisitions]</i> タブ および <i>[Authorizations]</i> タブにあるビューのリストに対応しています。 ステータス フィルタは、ビュー名 フィルタと組み合わせて使用する必要があります。ステータス フィルタ単独で使用することはできません。	http://serverURL/RequestCenter/nsapi/transaction/requisitions/ViewName=Ordered%20for%20Self http://serverURL/RequestCenter/nsapi/transaction/authorizations/ViewName=Authorizations%20for%20Self http://serverURL/RequestCenter/nsapi/transaction/authorizations/ViewName=Authorizations%20for%20Self Status=Approved

タスク	ビュー名 ?viewName=<value> フィルタに使用可能なビュー名は、 <i>Service Manager</i> のシステム定義ビューのリストに対応しています。ユーザ定義ビューをフィルタに使用することはできません。	http://serverURL/RequestCenter/nsapi/transaction/tasks?viewName=AvailableWork
タスク (特定の要求エントリに関するもの)	タスク タイプ ?taskType=<value> 指定可能なタスクのタイプ: <i>all</i> , <i>delivery</i> , <i>authorization</i> タスク ステータス (Skipped) ?showSkippedTasks=<false true>	http://<ServerURL>/RequestCenter/nsapi/transaction/tasks/RequisitionEntryNumber=1234?taskType=delivery&showSkippedTasks=true

POST 操作では、次のフィルタがサポートされています。

エンティティ タイプ	使用可能なフィルタ	REST URL の例
個人	ログイン名または Id フィルタは操作の中で暗黙的に行われ、個人の識別情報が要求 XML から取得されます。	http://serverURL/RequestCenter/nsapi/directory/people/update 作成アクションおよび更新アクションの両方で、この同じ URL が使用されます。個人の属性を URL に渡す必要はありません。
タスク	Id /<value>/<done approve reject review> 「 」は、指定可能なオプションの 1 つだけを選択する必要があることを示しています。	http://serverURL/RequestCenter/nsapi/transaction/tasks/215/approve

名前検索について

エンティティの名前フィルタでワイルドカード検索を行う場合 (例: ?name=<value>)、検索文字列の先頭のワイルドカード («*」や「%」) は無視されます。これらの文字が文字列の中または末尾にある場合は、ワイルドカード照合で適用されます。

名前検索において先頭のワイルドカード文字を有効にするには、`newscale.properties` ファイル (`RequestCenter.ear\config` にあります) 内のプロパティ `ContainsQueryInFnS` を見つけ、次のように値を `true` に設定してください。

```
ContainsQueryInFnS=true
```

このプロパティは、*Service Manager* および *Service Link* モジュールにおけるワイルドカード検索サポートもすべて制御します。このような検索操作は、システムのパフォーマンスに悪影響を及ぼす可能性があるため、一般的には実稼働環境での使用は推奨されません。

関連付けられているエンティティとは、プライマリ エンティティと一緒に取得される、ネストされたエンティティのことです。たとえば、個人がメンバとして属している OU などがあります。関連付けられているエンティティの名前フィルタでは、ワイルドカード検索はサポートされません。このような

フィルタによる検索では、完全一致の結果のみが返されます。また、大文字小文字が区別されます。つまり、検索文字列内のワイルドカード文字は、照合の際にリテラルとして扱われます。次に例を示します。

```
/nsapi/directory/people?ouname=star*OU
```

文字どおり「star*OU」という名前の組織単位に含まれている人が返されます。

```
/nsapi/directory/people?groupname=starGroup*
```

文字どおり「starGroup*」という名前のグループに含まれている人が返されます。

ソート

複数のデータ行を返す操作では、ソートコントロールを使用できます。フィルタと同様、ソートコントロールも次のように REST URL 内にパラメータとして指定します。

```
?sortBy=<columnName>&sortDir=<sortOrder>
```

- **sortBy** : ソート基準として使用するフィールドの名前。
- **sortDir** : ソートの方向。指定可能な値は **asc** (昇順) と **desc** (降順) です。

ソートパラメータが REST URL に渡されなかった場合は、次の表に示すように、デフォルトのソートフィールドとソート順序が使用されます。

エンティティ タイプ	デフォルトのソート フィールドとソート順序
カテゴリ	Name (昇順)
サービス	Name (昇順)
提供	Name (昇順)
エージェント	Name (昇順)
人	First Name (昇順)
組織単位	Name (昇順)
グループ	Name (昇順)
アカウント	Name (昇順)
すべてのサービス項目	Row Id* (昇順)
サービス項目の詳細	Row Id* (昇順)
標準	Row Id* (昇順)
カスタム コンテンツ	Row Id* (昇順)
要求	Submit Date (昇順)、Requisition Id (昇順)
要求エントリ	Requisition Entry ID (昇順)
タスク	Task Name (昇順)、Task Id (昇順)
承認	Requisition Id (昇順)、Task Id (昇順)
契約	Agreement Name (昇順)

Row Id とは、サービス項目、標準、およびカスタム コンテンツ テーブルにレコードが挿入された物理的な順序を表します。これは、それらの各テーブルの PrimaryID というカラムに対応しています。

サービス項目、標準、およびカスタム コンテンツの各エンティティは、すべてのカラムでソートできます。他のエンティティは、特定のフィールドによるソートだけがサポートされています。詳細については、「[詳細な API リファレンス](#)」(P.6-15) を参照してください。

例

```
/nsapi/directory/people?sortBy=lastName&sortDir=asc
```

個人レコードが、姓の昇順にリストされて返されます。

```
/nsapi/directory/people?sortBy=login&sortDir=desc
```

個人レコードが、ログイン名の降順にリストされて返されます。

ページング

複数のデータ行を返す操作では、ページコントロールを使用できます。これらも次のように REST URL 内にパラメータとして指定します。

```
?startRow=<x>&recordSize=<y>
```

- **startRow** : レコードの取得を開始する行。デフォルト値は 1 です。
- **recordSize** : 一度に取得するレコード数。デフォルト値は Portal Designer の共通設定で設定されます。許容最大レコード数は 50 です。

上記の各パラメータおよび返されるレコードの総数が、REST XML 応答のルート タグ内に属性として示されます。次に例を示します。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<categories totalCount="11" recordSize="10" startRow="1">
  <category>
    <categoryId>1</categoryId>
    <categoryName><b>Consumer Services</b></categoryName>
    <description />
    .. .
```

取得されるレコード数よりも多い数を **startRow** に指定した場合、HTTP 500 エラーが返されます。取得されるレコード数よりも多い数を **recordSize** に指定した場合、すべての行が返されます。

例

Service Designer に 60 個のサービスが定義されていて、Portal Designer の nsAPI 設定に指定されているレコードの最大数が 30 であるとします。

```
/nsapi/definition/servicedefs
```

最初の 30 個のサービスが返されます。

```
<services totalCount="60" recordSize="30" startRow="1">
  ...
</services>
```

```
/nsapi/definition/servicedefs?startRow=4
```

4 番目のサービスから 30 個のサービスが返されます。

```
<services totalCount="60" recordSize="30" startRow="4">
  .. .
</services>
```

```
/nsapi/definition/servicedefs?startRow=4&recordSize=5
```

4 番目のサービスから 5 個のサービスが返されます。

```
<services totalCount="60" recordSize="5" startRow="4">
```

```

    ..
  </services>

/nsapi/definition/servicedefs?startRow=4&recordSize=35

4 番目のサービスから 30 個のサービスが返されます。

<services totalCount="60" recordSize="30" startRow="4">
  ..
</services>

```

ネストされたエンティティ

一部のエンティティには、ネスト構造が含まれています。nsAPI は、第 1 レベルの子または関連付けられているエンティティの取得だけをサポートしています。親エンティティとその子エンティティについて、次の表にまとめます。

親エンティティ	子エンティティ
カテゴリ	カテゴリに関するサブカテゴリ、組み込みサービス、および提供
サービス	サービスに関連付けられているカテゴリ、キーワード、およびバンドル化サービス
提供サービス	提供サービスに関連付けられているカテゴリ、キーワード、目的、コストドライバ、およびコンポーネント サービス
個人	個人に関連付けられている OU、グループ、ロール、住所、連絡先、設定、および代理人

結果セット内に複数のデータ行を返すカテゴリ検索、サービス検索、または個人検索の場合は、パフォーマンス上の理由で、関連するエンティティは取得されません。個々のエンティティに対する Id または Name 動作による GET でのみ、関連するエンティティを取得可能です。

例

```
/nsapi/directory/people/loginname/<value>
```

個人がメンバとして属しているグループが返されます。

```

<person>
  ..
<associatedGroups>
  <associatedGroup>
    <id>2</id>
    <name>group2</name>
  </associatedGroup>
  <associatedGroup>
    <id>5</id>
    <name>group5</name>
  </associatedGroup>
</associatedGroups>
</person>

```

```
/nsapi/directory/people/id/<value>
```

個人が属しているグループが返されます。

```
/nsapi/directory/people
```


個人が属しているグループは返されません。

```
/nsapi/directory/people?ouname=<value>
```

個人が属しているグループは返されません。

REST API の呼び出し

HTTP クライアントでの nsAPI の使用

ブラウザまたはその他の HTTP クライアントから nsAPI を呼び出すには、アプリケーション認証が必要です。

Service Portal に正常にログインしたら、有効な nsAPI REST URL をブラウザのアドレス バーに入力します。次に例を示します。

```
http://<serverURL>/RequestCenter/nsapi/definition/categories/id/3
```

次のような応答 XML がブラウザに表示されます。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<category>
  <categoryId>3</categoryId>
  <categoryName>Workplace Services</categoryName>
  <description>Services for voice and data communications, desktop, mobile devices, and
application access.</description>
  <topDescriptionEnabled>>false</topDescriptionEnabled>
  <topDescription />
  <middleDescriptionEnabled>>false</middleDescriptionEnabled>
  <middleDescription />
  <bottomDescriptionEnabled>>false</bottomDescriptionEnabled>
  <bottomDescription />
  <catalogTypeId>1</catalogTypeId>
  <catalogType>Consumer Services Catalog</catalogType>
  <isRoot>>false</isRoot>
  <associatedServices>
    <associatedService>
      <description>Order a new or refurbished laptop.Manager approval
required.</description>
      <id>20</id>
      <name>New Laptop</name>
      <status>Active</status>
    </associatedService>
    <associatedService>
      <description>Order a new iPhone or Blackberry, configured and maintained under
corporate policy.</description>
      <id>22</id>
      <name>New Mobile Device</name>
      <status>Active</status>
    </associatedService>
  </associatedServices>
  <includedCategories>
    <includedCategory>
      <id>8</id>
      <name>Email</name>
    </includedCategory>
    <includedCategory>
      <id>9</id>
      <name>Laptops</name>
    </includedCategory>
  </includedCategories>
</category>
```

```

    </includedCategory>
  </includedCategories>
  <categoryURLSc>
    <a
href='/RequestCenter/myservices/navigate.do?categoryid=3&query=catalog&layout=popu
p_p' onclick="return GB_showFullScreen('Category', this.href)">Workplace Services</a>
    </categoryURLSc>
    <categoryURLOnlySc>/RequestCenter/myservices/navigate.do?categoryid
=3&query=catalog</categoryURLOnlySc>
  </category>

```

アプリケーションにログインする前に REST API 要求が実行された場合、URL は次のエラーを返します。

```
HTTP Error 401 Unauthorized
```

他のクライアントを介して行われた HTTP 要求の場合は、次のように nsAPI ログイン URL に HTTP ヘッダー パラメータとして認証用資格情報を渡すことができます。

```
http://<serverURL>/RequestCenter/nsapi/authentication/login
```

この HTTP ヘッダーには、次のパラメータを含める必要があります。

```
username=<username>
```

```
password=<password>
```

認証が成功したら、HTTP 応答で JSessionID クッキーが返されます。後続の nsAPI 呼び出しで認証を再び行うことなくセッションを維持するためには、同じ JSessionID クッキーを要求に含める必要があります。

JavaScript ポートレットでの nsAPI の使用

Portal Designer モジュールで作成した JavaScript ポートレット内で、nsAPI 呼び出しを実行することができます。JavaScript、REST URL、AJAX または Ext JS コンポーネントを使用して、必要なエンティティのデータを取得するポートレットを作成し、そのポートレットをグリッド形式で表現できます。

Ext JS グリッドでのデータの表現

Ext JS を使用してカテゴリのリストを表現する例を、次に示します。

```

/* create the Data Store */
var store = new Ext.data.Store(<
  /* load using HTTP */
  URL :
'http://<serverURL>/RequestCenter/nsapi/definition/servicecatalog/categories',

  /* the return is XML, so let's set up a reader */
  reader : new Ext.data.XmlReader(<
    root : "categorys",
    record : "category"
  >, [<
    name : "id"
  >, <
    name : "name"
  >, <
    name : "description"
  >])
  >);

```

```
/* create the grid */
var grid = new Ext.grid.GridPanel(<
    store : store,
    columns : [<
        header : "Category ID",
        dataIndex : 'id',
        sortable : true
    >, <
        header : "Category Name",
        dataIndex : "name",
        sortable : true
    >, <
        header : "Description",
        dataIndex : "description",
        sortable : true
    >],
    renderTo : '#divName#',
    width : "100%",
    autoHeight : true,
    layout : 'fit',
    viewConfig : <
        forceFit : true
    >,
    bbar : new Ext.PagingToolbar(<
        pageSize : 25,
        store : store,
        displayInfo : true,
        displayMsg : 'Displaying topics <0> - <1> of <2>',
        emptyMsg : "No topics to display"
    >)
>);

store.load();
```

ログインユーザの取得

現在ログインしているユーザの名前空間変数を取得し、JavaScript ポートレット内で使用できます。

```
Ext.onReady(function() {
    /* Demonstrate JavaScript to get Logged-In user details */
    alert('\PersonId: ' + nsAPP_CurrentUserId);
    alert('\Login name: ' + nsAPP_CurrentUserLoginName);
    alert('\First name: ' + nsAPP_CurrentUserFirstName);
    alert('\Last name: ' + nsAPP_CurrentUserLastName);
    alert('\HomeOUIId: ' + nsAPP_CurrentUserHomeOUIId);
})
```

JSR ポートレットでの nsAPI の使用

認証

JSR ポートレットからアクセスする場合、nsAPI Java クライアントを使用してログイン操作およびログアウト操作を呼び出すことができます。

```
import com.newscale.nsapiclient.NSApiClientFactory;
import com.newscale.nsapiclient.NSApiClient;
.. ..
```

```

NSApiClient nsApiClient = NSApiClientFactory.getInstance();
nsApiClient.login("http://<serverURL>/RequestCenter", "username", "password" ); // Login
by username, password.
..
..
nsApiClient.logout(); // Logout

..
..
NSApiClient nsApiClient = NSApiClientFactory.getInstance();
nsApiClient.login("http://<serverURL>/RequestCenter",
sessionId); // Login using current session id.
..
..
nsApiClient.logout();

```

ログイン ユーザの取得

次に、ログインしているユーザの詳細情報を取得する、Spring ベースの JSR Portlet Controller の例を示します。

```

import javax.portlet.RenderRequest;
import javax.portlet.RenderResponse;
import org.springframework.ui.Model;
import org.springframework.stereotype.Controller;
import com.newscale.nsapi.directory.person.Person;
...
@Controller
public class MyJSRController {
private NSApiClient nsApiClient = getNSApiClient();
public NSApiClient getNsApiClient() {
return nsApiClient;
}
public void setNsApiClient(NSApiClient nsApiClient) {
this.nsApiClient = nsApiClient;
}
@RequestMapping("VIEW")
@RenderMapping("NORMAL")
public String viewNormal(RenderRequest request, RenderResponse response, Model model) {
nsApiClient.login("http://<AppServer host>:<port>/RequestCenter",
request.getPortletSession().getId());
// Get Currently Logged-in user from nsAPI client
Person persons = nsApiClient.getDirectory().getCurrentUser();
// Get user info
long personId = persons.getPersonId();
long homeOUId = persons.getHomeOrganizationalUnitId();
String firstName =persons.getFirstName();
String lastName =persons.getLastName();
String username =persons.getLogin();
}
}

```

Get 操作

ここでは、エンティティの取得方法を示すサンプル コードの一部をいくつか示します。これらの方法の詳細については、個々のエンティティ クラスに関する Javadoc を参照してください。

- Id による個人の取得

```

package com.newscale.nsapiclient.directory;
import com.newscale.nsapiclient.directory.person.Person;
import com.newscale.nsapi.NSApiConstants;

```

```

import com.newscale.nsapiclient.directory.Directory;
import com.newscale.nsapiclient.NSApiclientFactory;
import com.newscale.nsapiclient.NSApiclient;
.. .

NSApiclient nsApiClient = NSApiclientFactory.getInstance();
nsApiClient.login("http://<serverURL>/RequestCenter", "username", "password");

Person person = nsApiClient.getDirectory().getPersonById(123);

/* This is the equivalent of REST URL
http://<serverURL>/RequestCenter/nsapi/directory/people/id/123
*/
.. .
nsApiClient.logout();

```

- Name による個人の取得

```

package com.newscale.nsapiclient.directory;
import com.newscale.nsapiclient.directory.person.Person;
import com.newscale.nsapi.NSApiConstants;
import com.newscale.nsapiclient.directory.Directory;
import com.newscale.nsapiclient.NSApiclientFactory;
import com.newscale.nsapiclient.NSApiclient;
.. .
NSApiclient nsApiClient = NSApiclientFactory.getInstance();
nsApiClient.login("http://<serverURL>/RequestCenter", "username", "password");
Person person = nsApiClient.getDirectory().getPersonByLoginName("jsmith");
/*
This is the equivalent of REST URL
http://<serverURL>/RequestCenter/nsapi/directory/people/loginname/jsmith
*/
.. .
nsApiClient.logout();

```

- すべての人の取得

```

package com.newscale.nsapiclient.directory;
import com.newscale.nsapiclient.directory.person.Person;
import com.newscale.nsapi.NSApiConstants;
import com.newscale.nsapiclient.directory.Directory;
import com.newscale.nsapiclient.NSApiclientFactory;
import com.newscale.nsapiclient.NSApiclient;
import java.util.List;
import org.apache.commons.collections.map.MultiValueMap;
.. .
NSApiclient nsApiClient = NSApiclientFactory.getInstance();
nsApiClient.login("http://<serverURL>/RequestCenter", "username", "password");

PersonList persons = nsApiClient.getDirectory().getPeople(null);

/*
This is the equivalent of REST URL
http://<serverURL>/RequestCenter/nsapi/directory/people
*/
.. .
nsApiClient.logout();

```

- クエリーフィルタによる個人の取得

```

package com.newscale.nsapiclient.directory;
import com.newscale.nsapiclient.directory.person.Person;
import com.newscale.nsapi.NSApiConstants;
import com.newscale.nsapiclient.directory.Directory;

```

```

import com.newscale.nsapiclient.NSApiClientFactory;
import com.newscale.nsapiclient.NSApiClientConstants;
import com.newscale.nsapiclient.NSApiClient;
import org.apache.commons.collections.map.MultiValueMap;

import java.util.List;

NSApiClient nsApiClient = NSApiClientFactory.getInstance();
nsApiClient.login("http://<serverURL>/RequestCenter","username","password");

MultiValueMap filterMap = new MultiValueMap();//this can be used to specify multiple
filter criteria
filterMap.put(NSApiClientConstants.QUERYPARAM_NAME, "John");
PersonList persons = nsApiClient.getDirectory().getPeople(filterMap);

/*
  This is the equivalent of REST URL
  http://<serverURL>/RequestCenter/nsapi/directory/people?name=John
*/
..
nsApiClient.logout();

```

- 複数のクエリー フィルタによる個人の取得

```

package com.newscale.nsapiclient.directory;
import com.newscale.nsapiclient.directory.person.Person;
import com.newscale.nsapi.NSApiConstants;
import com.newscale.nsapiclient.directory.Directory;
import com.newscale.nsapiclient.NSApiClientFactory;
import com.newscale.nsapiclient.NSApiClientConstants;
import com.newscale.nsapiclient.NSApiClient;
import org.apache.commons.collections.map.MultiValueMap;

import java.util.List;
..
NSApiClient nsApiClient = NSApiClientFactory.getInstance();
nsApiClient.login("http://<serverURL>/RequestCenter","username","password");

MultiValueMap filterMap = new MultiValueMap();
filterMap.put(NSApiClientConstants.QUERYPARAM_NAME, "John");
filterMap.put(NSApiClientConstants.QUERYPARAM_SORTBY, "lastName");
filterMap.put(NSApiClientConstants.QUERYPARAM_SORTDIR, "asc");
PersonList persons = nsApiClient.getDirectory().getPeople(filterMap);

// This is the equivalent of REST URL
// http://<serverURL>/RequestCenter/nsapi/directory/people?name
// =John&sortBy=lastName&sortDir=asc
// search for people by the name John,
// sort the result set by Last Name in ascending order
..
nsApiClient.logout();

```

Post 操作

ここでは、個人の作成および更新方法、ならびにタスクに対するアクションの実行方法を示すサンプルコードの一部をいくつか示します。これらの方法の詳細については、個々のエンティティクラスに関する Javadoc を参照してください。

- 個人の更新

```

package com.newscale.nsapiclient.directory;
import com.newscale.nsapiclient.directory.person.Person;
import com.newscale.nsapi.NSApiConstants;

```

```

import com.newscale.nsapiclient.directory.Directory;
import com.newscale.nsapiclient.NSApiClientFactory;
import com.newscale.nsapiclient.NSApiClient;
.. .
NSApiClient nsApiClient = NSApiClientFactory.getInstance();
nsApiClient.login("http://<serverURL>/RequestCenter", "username", "password");
Person person = NSApiClient.getDirectory().getPersonById(123);
person.setLastName("Smith");
Person persons = NSApiClient.getDirectory().updatePerson(person);
/*
This is the equivalent of posting XML of person 123 with last name changed to "Smith" to
the REST URL
http://<serverURL>/RequestCenter/nsapi/directory/people/update
*/
.. .
nsApiClient.logout();

```

- 配信タスクの完了

```

package com.newscale.nsapiclient.transaction;
import com.newscale.nsapiclient.transaction.task.TaskAction;
import com.newscale.nsapi.NSApiConstants;
import com.newscale.nsapiclient.transaction.Transaction;
import com.newscale.nsapiclient.NSApiClientFactory;
import com.newscale.nsapiclient.NSApiClientConstants;
import com.newscale.nsapiclient.NSApiClient;
import org.apache.commons.collections.map.MultiValueMap;

import java.util.List;
.. .
NSApiClient nsApiClient = NSApiClientFactory.getInstance();
nsApiClient.login("http://<serverURL>/RequestCenter", "username", "password");

// This is the equivalent of REST URL
// http://<serverURL>/RequestCenter/nsapi/tasks/10/done

NSApiClient.getTransaction().completeTask(10);
.. .
nsApiClient.logout();

```

詳細な API リファレンス

nsAPI の javadoc は、製品インストーラの Image フォルダにあります。REST URL および java クライアントから nsAPI を呼び出す例を、サポートされているエンティティごとに次に示します。

定義データ

カテゴリ

エリア	例
コア API	<p>すべてのカテゴリの取得</p> <p>デフォルトでは、タイプが「Service Catalog」のカテゴリのみが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/categories</code></p> <p>Java の例 :</p> <pre>CategoryList categories = NSApiClient.getDefinition().getCategories(null);</pre>
	<p>すべてのコンシューマ サービス カテゴリの取得</p> <p>「Consumer Services」タイプのすべてのカテゴリが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/categories</code></p> <p>Java の例 :</p> <pre>CategoryList categories = NSApiClient.getDefinition().getCategories("serviceCatalog");</pre>
	<p>すべての提供サービスのカテゴリの取得</p> <p>「Service Offerings」タイプのすべてのカテゴリが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/categories?catalogType=offeringCatalog</code></p> <p>Java の例 :</p> <pre>CategoryList categories = NSApiClient.getDefinition().getCategories("offeringCatalog");</pre>
	<p>Name によるコンシューマ サービス カテゴリの取得</p> <p>指定した名前のサービス カタログ カテゴリが返されます。ネストされたエンティティ (サブカテゴリおよび組み込みサービス) が取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/categories/name/<categoryName></code></p> <p>Java の例 :</p> <pre>Category categories = NSApiClient.getDefinition().getCategoryByName("<categoryName>");</pre>

エリア	<p>例</p> <p>Id によるコンシューマ サービス カテゴリの取得</p> <p>指定した Id のサービス カタログ カテゴリが返されます。ネストされたエンティティ (サブカテゴリおよび組み込みサービス) が取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/categories/id/<categoryId></code></p> <p>Java の例 :</p> <pre>Category categories = NSApiClient.getDefinition().getCategoriesById(<categoryId>);</pre> <hr/> <p>Name による提供サービス カテゴリの取得</p> <p>ネストされたエンティティ (サブカテゴリおよび組み込み提供) が含まれます。指定した名前のサービス提供カテゴリが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/categories/name/<categoryName>?catalogType=offeringCatalog</code></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put(QUERYPARAM_CATALOG_TYPE, "offeringCatalog"); Category categories = NSApiClient.getDefinition().getCategoryByName("<categoryName>");</pre>
フィルタ	<p>カテゴリ名フィルタ</p> <p>検索では大文字と小文字が区別されます。</p> <p>StartsWith (newscale.properties の ContainsQueryInFnS=false) : 先頭のワイルドカードは無視されます。</p> <p>Contains (newscale.properties の ContainsQueryInFnS=true) : 先頭のワイルドカードは適用されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/categories?name=<wildcardValue>&catalogType=serviceCatalog</code></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put(QUERYPARAM_CATALOG_TYPE, "serviceCatalog"); paramsMap.put(QUERYPARAM_NAME, "<wildcardValue>"); CategoryList categories = NSApiClient.getDefinition().getCategories(paramsMap);</pre>

エリア	例
ソート カラム	Category Name
応答 XML	<pre><categories totalCount="x" recordSize="y" startRow="z"> <category> .. . <associatedServices> .. . </associatedServices> </category> .. . <associatedService> <description> </description> <id></id> <imageUrl></imageUrl> <name> </name> <status> </status> </associatedService> .. . </associatedServices> .. . </category> </categories></pre>

サービス

エリア	例
コア API	<p>すべてのサービスの取得</p> <p>すべてのサービスが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/servicedefs</code></p> <p>Java の例 :</p> <pre>ServiceList services = NSApiClient.getDefinition().getServices(null);</pre> <hr/> <p>サービス カタログのワイルドカード検索によるすべてのサービスの取得</p> <p>サービスの名前、サービスの説明、カテゴリ、またはキーワードが検索文字列と一致しているサービスがすべて返されます。先頭のワイルドカードはサポートされません。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/servicedefs?search={wildcardValue*}</code></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put("search", "<wildcardValue*>"); ServiceList services = NSApiClient.getDefinition().getServices(paramsMap);</pre>

エリア	例
	<p>Id によるサービスの取得</p> <p>ネストされたエンティティ（関連付けられているカテゴリおよびキーワード）は、<code>getById</code> および <code>getByName</code> の場合にのみ取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/servicedefs/id/<serviceId></code></p> <p>Java の例 :</p> <pre>Service services = NSApiClient.getDefinition().getServiceById(<serviceId>);</pre>
	<p>Name によるサービスの取得</p> <p>ネストされたエンティティ（関連付けられているカテゴリおよびキーワード）は、<code>getById</code> および <code>getByName</code> の場合にのみ取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/servicedefs/name/<serviceName></code></p> <p>Java の例 :</p> <pre>Service services = NSApiClient.getDefinition().getServiceByName("<serviceName>");</pre>
	<p>カテゴリに含まれるすべてのサービスの取得</p> <p>そのカテゴリに関連付けられているすべてのサービスが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/servicedefs?CategoryName=<categoryName></code></p> <p>Java の例 :</p> <pre>ServiceList services =NSApiClient.getDefinition().getServiceByCategoryName("<categoryName>");</pre>
	<p>キーワードによるすべてのサービスの取得</p> <p>そのキーワードに関連付けられているすべてのサービスが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/servicedefs?KeywordName=<keyword></code></p> <p>Java の例 :</p> <pre>ServiceList services = NSApiClient.getDefinition().getServiceByKeyword("<keyword>");</pre>

エリア	例
フィルタ	<p>サービス名フィルタ</p> <p>検索では大文字と小文字が区別されます。</p> <p>StartsWith (newscale.properties の ContainsQueryInFnS=false) : 先頭のワイルドカードは無視されます。</p> <p>Contains (newscale.properties の ContainsQueryInFnS=true) : 先頭のワイルドカードは、必要に応じて指定してください。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/definition/servicedefs?name=<wildcardValue></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put ("name", "<wildcardValue*>"); ServiceList servicesList = NSApiClient.getDefinition().getServices (paramsMap)</pre>
ソート カラム	Service Name
応答 XML	<pre><services totalCount="x" recordSize="y" startRow="z"> <service> .. . <includedServices> .. . <includedService> <id></id> <name> </name> </includedService> .. . </includedServices> .. . </service> </services></pre>

提供サービス

エリア	例
コア API	<p>すべての提供サービスの取得</p> <p>すべての提供サービスが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/serviceofferings</code></p> <p>Java の例 :</p> <pre>BusinessServiceList offerings = NSApiClient.getDefinition().getBusinessServices(null);</pre> <hr/> <p>Id による提供サービスの取得</p> <p>ネストされたエンティティ（関連付けられているカテゴリおよびキーワード）は、<code>getById</code> および <code>getByName</code> の場合にのみ取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/serviceofferings/id/<offeringId></code></p> <p>Java の例 :</p> <pre>BusinessService offerings = NSApiClient.getDefinition().getBusinessServiceById(<offeringId>);</pre> <hr/> <p>Name による提供サービスの取得</p> <p>ネストされたエンティティ（関連付けられているカテゴリおよびキーワード）は、<code>getById</code> および <code>getByName</code> の場合にのみ取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/serviceofferings/name/<offeringName></code></p> <p>Java の例 :</p> <pre>BusinessService of= NSApiClient.getDefinition().getBusinessServiceByName("<offeringName>");</pre>
フィルタ	<p>提供名フィルタ</p> <p>検索では大文字と小文字が区別されます。</p> <p>StartsWith (<code>newscale.properties</code> の <code>ContainsQueryInFnS=false</code>) : 先頭のワイルドカードは無視されます。</p> <p>Contains (<code>newscale.properties</code> の <code>ContainsQueryInFnS=true</code>) : 先頭のワイルドカードは、必要に応じて指定してください。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/serviceofferings?name=<wildcardValue></code></p> <p>Java の例 :</p> <pre>BusinessServiceList offer = NSApiClient.getDefinition().getBusinessServicesByFilter("<wildcardValue>");</pre>

エリア	例
ソートカラム	Offering Name
応答 XML	<pre><serviceOfferings totalCount="x" recordSize="y" startRow="z"> <serviceOffering> .. </serviceOffering> </serviceOfferings></pre>

エージェント

エリア	例
コア API	<p>すべてのエージェントの取得</p> <p>すべてのエージェントが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/agents</code></p> <p>Java の例 :</p> <pre>AgentList agents = NSApiClient.getDefinition().getAgents(null);</pre> <hr/> <p>Id によるエージェントの取得</p> <p>発信および着信のプロパティは、<code>getById</code> および <code>getByName</code> の場合にのみ取得されません。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/agents/id/<agentId></code></p> <p>Java の例 :</p> <pre>Agent agents = NSApiClient.getDefinition().getAgentById(<agentId>);</pre> <hr/> <p>Name によるエージェントの取得</p> <p>発信および着信のプロパティは、<code>getById</code> および <code>getByName</code> の場合にのみ取得されません。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/definition/agents/name/<agentName></code></p> <p>Java の例 :</p> <pre>Agent Agents = NSApiClient.getDefinition().getAgentByName("<agentName>");</pre>

エリア	例
フィルタ	<p>エージェント名フィルタ</p> <p>検索では大文字と小文字が区別されます。</p> <p>StartsWith (newscale.properties の ContainsQueryInFnS=false) : 先頭のワイルドカードは無視されます。</p> <p>Contains (newscale.properties の ContainsQueryInFnS=true) : 先頭のワイルドカードは、必要に応じて指定してください。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/definition/agents?name=<wild cardValue></p> <p>Java の例 :</p> <pre>AgentList agents = NSApiClient.getDefinition().getAgentsByFilter("<wildcardValue>");</pre>
ソートカラム	Agent Name
応答 XML	<pre><agents totalCount="x" recordSize="y" startRow="z"> <agent> .. </agent> </agents></pre>

契約

エリア	例
コア API	<p>すべての契約の取得</p> <p>すべての契約が返されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/definition/agreements</p> <p>Java の例 :</p> <pre>AgreementList agreements = NSApiClient.getDefinition().getAgreements(null);</pre>
	<p>Id による契約の取得</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/definition/agreements/id/<agreementId></p> <p>Java の例 :</p> <pre>Agreement agreements = NSApiClient.getDefinition().getAgreementById(<agreementId>);</pre>
	<p>Name による契約の取得</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/definition/agreements/name/<agreementName></p> <p>Java の例 :</p> <pre>Agreement agreements = nsclient.getDefinition().getAgreementByName("<agreementName>");</pre>

エリア	例
フィルタ	<p>契約名フィルタ</p> <p>検索では大文字と小文字が区別されます。</p> <p>StartsWith (newscale.properties の ContainsQueryInFnS=false) : 先頭のワイルドカードは無視されます。</p> <p>Contains (newscale.properties の ContainsQueryInFnS=true) : 先頭のワイルドカードは、必要に応じて指定してください。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/definition/agreements?name=<wildcardValue></p> <p>Java の例 :</p> <pre>AgreementList agreements = NSApiClient.getDefinition().getAgreementsByFilter("<wildcardValue>");</pre>
ソートカラム	<p>Agreement Name</p>
応答 XML	<pre><agreements totalCount="x" recordSize="y" startRow="z"> <agreement> . . </agreement> </agreements></pre>

ディレクトリ データ

個人

エリア	例
コア API	<p>すべての人の取得</p> <p>すべての人が返されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/directory/people</p> <p>Java の例 :</p> <pre>PersonList person = NSApiClient.getDirectory().getPeople(null);</pre>
	<p>Id による個人の取得</p> <p>指定した Person Id を持つ個人が返されます。</p> <p>ネストされたエンティティ (OU、グループ、ロール、住所、連絡先、および設定) が取得されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/directory/people/id/<personId></p> <p>Java の例 :</p> <pre>Person persons = NSApiClient.getDirectory().getPersonById(<personId>);</pre>

エリア	例
	<p>LoginName による個人の取得</p> <p>指定した LoginName を持つ個人が返されます。</p> <p>ネストされたエンティティ（OU、グループ、ロール、住所、連絡先、および設定）が取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/people/loginname/<loginName></code></p> <p>Java の例 :</p> <pre>Person persons = NSApiClient.getDirectory().getPersonByLoginName("<loginName>");</pre>
	<p>ログイン ユーザの取得</p> <p>現在ログインしている個人が返されます。</p> <p>REST URL :</p> <p><code>http://<serverURL>/RequestCenter/nsapi/directory/people/currentuser</code></p> <p>Java の例 :</p> <pre>Person person = NSApiClient.getDirectory().getCurrentUser();</pre>

エリア	例
特殊条件	<p>個人の作成/更新</p> <p>個人を取得 (GET) する前述の REST URL のいずれかの応答から、個人 XML を取得します。</p> <p>個人を更新する REST URL に対して個人 XML を POST することによって、個人の作成または変更を行います。</p> <p>(XML のログインまたは <code>personId</code> で識別される) 個人が存在する場合は更新操作が実行され、そうでない場合は新規個人が作成されます。</p> <p>作成操作には、次の 5 つの要素が必要です。</p> <ul style="list-style-type: none"> • <code>firstName</code> • <code>lastName</code> • <code>homeOrganizationalUnitName</code> • <code>email</code> • <code>login</code> <p>作成操作では、個人のパスワードにログイン名が設定されます。</p> <p>更新操作でホーム OU に対する変更を行うと、個人のホーム OU が置換されます。関連付けられている他の OU、グループ、およびロールは、作成操作でも更新操作でも無視されます。</p> <p>ホーム OU、タイムゾーン、ロケール、スーパーバイザ、代理承認者、ログイン モジュール、連絡先、および住所の各属性には、次のルールが適用されます。</p> <ul style="list-style-type: none"> • <code>id</code> 要素が XML で送信されたがデータベースには見つからなかった場合、例外がスローされ、該当するメッセージが示されます。 • <code>id</code> 要素が見つからなかった場合は、XML で送信された <code>Name</code> 要素が代わりに使用されます。その名前がデータベースに見つからなかった場合は、例外がスローされます。 • <code>id</code> または <code>name</code> 要素のいずれも XML 内に存在せず、属性がオプションである場合、その属性は作成/更新操作で無視されます。(HomeOU は必須です)。 • XML で送信されたデータに誤りや欠落があったために作成または更新操作が失敗した場合、HTTP ステータス コード「422 Unprocessable Entity」が返されます。 <p>POST REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/people/update</code></p> <p>Java の例 :</p> <pre>Person person = NSApiClient.getDirectory().getPersonById(<personId>); person.setLastName("<lastName>"); Person persons = NSApiClient.getDirectory().updatePerson(person);</pre>

エリア	例
フィルタ	<p>OU 名フィルタ</p> <p>検索では大文字と小文字が区別され、完全一致が使用されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/directory/people?ouname=<ouName></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put("ouname", "<ouName>"); PersonList person = NSApiClient.getDirectory().getPeople(paramsMap);</pre>
	<p>グループ名フィルタ</p> <p>検索では大文字と小文字が区別され、完全一致が使用されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/directory/people?groupname=<groupName></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put("groupname", "<groupName>"); PersonList person = NSApiClient.getDirectory().getPeople(paramsMap);</pre>
	<p>ロール名フィルタ</p> <p>検索では大文字と小文字が区別され、完全一致が使用されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/directory/people?rolename=<roleName></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put("rolename", "<roleName>"); PersonList person = NSApiClient.getDirectory().getPeople(paramsMap);</pre>
ソートカラム	First Name、Last Name、Login Name
XML 応答	<pre><people totalCount="x" recordSize="y" startRow="z"> <person> .. . <contacts> .. . <contact> <contactId></contactId> <contactType></contactType> <contactTypeId></contactTypeId> <value></value> </contact> .. . </contacts> .. . </person> </people></pre>

組織単位

エリア	例
コア API	<p>すべての組織単位の取得</p> <p>すべての組織単位が返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/organizationalunits</code></p> <p>Java の例 :</p> <pre>OrganizationalUnitList Ou = NSApiClient.getDirectory().getOrganizationalUnits(null);</pre> <hr/> <p>Id による組織単位の取得</p> <p>ネストされたエンティティ (サブ組織単位) は、<code>getById</code> および <code>getName</code> の場合にのみ取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/organizationalunits/id/<ouId></code></p> <p>Java の例 :</p> <pre>OrganizationalUnit Ou = NSApiClient.getDirectory().getOrganizationalUnitById(ouId);</pre> <hr/> <p>Name による組織単位の取得</p> <p>ネストされたエンティティ (サブ組織単位) は、<code>getById</code> および <code>getName</code> の場合にのみ取得されます。</p> <p>同じ名前タイプが異なる OU が 2 つ見つかった場合は、サービス チーム OU が返されます。</p> <p>オプションのパラメータ <code>?type=<businessUnit/serviceTeam></code> を指定することもできます。(この <code>type</code> パラメータ値に「all」を指定することはできません。)</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/organizationalunits/name/<ouName></code></p> <p>Java の例 :</p> <pre>OrganizationalUnit Ou = NSApiClient.getDirectory().getOrganizationalUnitByName("<ouName>");</pre> <hr/> <p>Type による OU の取得</p> <p>指定した OU タイプの組織単位がすべて返されます。</p> <p>指定可能な値: 「all」、「businessUnit」、「serviceTeam」。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/organizationalunits?type=<ouType></code></p> <p>Java の例 :</p> <pre>OrganizationalUnitList Ou = NSApiClient.getDirectory().getOrganizationalUnitByType("<ouType>");</pre>

エリア	例
フィルタ	<p>組織単位名フィルタ</p> <p>検索では大文字と小文字が区別されます。</p> <p>StartsWith (newscale.properties の ContainsQueryInFnS=false) : 先頭のワイルドカードは無視されます。</p> <p>Contains (newscale.properties の ContainsQueryInFnS=true) : 先頭のワイルドカードは、必要に応じて指定してください。</p> <p>ワイルドカードの名前検索では、オプションのパラメータ <code>?type=<businessUnit/serviceTeam></code> を指定することもできます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/organizationalunits?name=<wildcardValue></code></p> <p>Java の例 :</p> <pre>OrganizationalUnitList Ou = NSApiClient.getDirectory().getOrganizationalUnitsByFilter("<wildcardValue>");</pre>
ソートカラム	Organizational Unit Name
応答 XML	<pre><organizationalunits totalCount="x" recordSize="y" startRow="z"> <organizationalunit> . . . </organizationalunit> </organizationalunits></pre>

グループ

エリア	例
コア API	<p>すべてのグループの取得</p> <p>すべてのグループが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/groups</code></p> <p>Java の例 :</p> <pre>GroupList groups = NSApiClient.getDirectory().getGroups(null);</pre> <p>Id によるグループの取得</p> <p>ネストされたエンティティ (サブグループ) は、<code>getById</code> および <code>getByName</code> の場合にのみ取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/groups/id/<groupId></code></p> <p>Java の例 :</p> <pre>Group groups = NSApiClient.getDirectory().getGroupsById(<groupId>);</pre>

エリア	例
	<p>Name によるグループの取得</p> <p>ネストされたエンティティ（サブグループ）は、<code>getById</code> および <code>getByName</code> の場合にのみ取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/groups/name/<groupName></code></p> <p>Java の例 :</p> <pre>Group groups = NSApiClient.getDirectory().getGroupsByName("<groupName>");</pre>
フィルタ	<p>グループ名フィルタ</p> <p>検索では大文字と小文字が区別されます。</p> <p><code>StartsWith</code> (<code>newscale.properties</code> の <code>ContainsQueryInFnS=false</code>) : 先頭のワイルドカードは無視されます。</p> <p><code>Contains</code> (<code>newscale.properties</code> の <code>ContainsQueryInFnS=true</code>) : 先頭のワイルドカードは、必要に応じて指定してください。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/groups?name=<wild cardValue></code></p> <p>Java の例 :</p> <pre>GroupList groups = NSApiClient.getDirectory().getGroupsByFilter("<wildcardValue>");</pre>
ソートカラム	Group Name
応答 XML	<pre><groups totalCount="x" startRow="y" recordSize="z"> <group> .. </group> </groups></pre>

アカウント

エリア	例
コア API	<p>すべてのアカウントの取得</p> <p>すべてのアカウントが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/accounts</code></p> <p>Java の例 :</p> <pre>AccountList accounts = NSApiClient.getDirectory().getAccounts(null);</pre> <hr/> <p>Id によるアカウントの取得</p> <p>ネストされたエンティティ（関連付けられている組織単位）は、<code>getById</code> および <code>getByName</code> の場合にのみ取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/accounts/id/<accountId></code></p> <p>Java の例 :</p> <pre>Account accounts = NSApiClient.getDirectory().getAccountsById(<accountId>);</pre> <hr/> <p>Name によるアカウントの取得</p> <p>ネストされたエンティティ（関連付けられている組織単位）は、<code>getById</code> および <code>getByName</code> の場合にのみ取得されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/accounts/name/<accountName></code></p> <p>Java の例 :</p> <pre>Account accounts = NSApiClient.getDirectory().getAccountsByName("<accountName>");</pre>
フィルタ	<p>グループ名フィルタ</p> <p>検索では大文字と小文字が区別されます。</p> <p><code>StartsWith</code> (<code>newscale.properties</code> の <code>ContainsQueryInFnS=false</code>) : 先頭のワイルドカードは無視されます。</p> <p><code>Contains</code> (<code>newscale.properties</code> の <code>ContainsQueryInFnS=true</code>) : 先頭のワイルドカードは、必要に応じて指定してください。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/directory/accounts?name=<wild cardValue></code></p> <p>Java の例 :</p> <pre>AccountList accounts = NSApiClient.getDirectory().getAccountsByFilter("<wildcardName>");</pre>
ソートカラム	Account Name
応答 XML	<pre><accounts totalCount="x" startRow="y" recordSize="z"> <account> .. </account> </accounts></pre>

トランザクション データ

要求

エリア	例
コア API	<p>現行ユーザに対する要求の取得</p> <p>次に示すデフォルトのフィルタを使用して要求を取得します。</p> <p>ViewName = Ordered for Self</p> <p>Status = Ongoing</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/transaction/requisitions</p> <p>Java の例 :</p> <pre>RequisitionList requisitions = NSApiClient.getTransaction().getRequisitions(null);</pre>
	<p>Id による要求の取得</p> <p>ログインしているユーザに対して RBAC チェックが適用されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/transaction/requisitions/id/<requisitionId></p> <p>Java の例 :</p> <pre>Requisition requisitions = NSApiClient.getTransaction().getRequisitionsById(<requisitionId>);</pre>

エリア	例
フィルタ	<p>ビューおよびステータス フィルタ</p> <p>フィルタに使用可能なビューおよびステータスは、My Services の [Requisitions] タブにあるものに対応しています。</p> <p>Status が指定されなかった場合は、「Ongoing」が使用されます。</p> <p>ViewName が指定されなかった場合は、「Ordered for Self」が使用されます。</p> <p>ViewName に指定可能な値：</p> <ul style="list-style-type: none"> • Ordered for Self：現行ユーザに対する要求。 • Ordered for Others：現行ユーザが（「Order on Behalf」を使用して）他人の代わりに送信した要求。 • Ordered for my unit：現行ユーザが属している OU 内の人に関する要求（ユーザが「See Requisitions for My Business Units」機能を持っている場合に限り、このビューは他の人に関するデータを返します）。 <p>Status に指定可能な値：Ongoing、Preparation、Ordered、Closed、Cancelled、Rejected、All。</p> <p>誤った値が指定された場合、デフォルトの値（「Ongoing」および「Ordered for Self」）が使用されます。</p> <p>REST URL：</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/transaction/requisitions/ViewName=<viewName>[AND Status=<status>]</code></p> <p>Java の例：</p> <pre>String filterString = "ViewName=<viewName> AND Status=<status>"; RequisitionList requisitions = NSApiClient.getTransaction().getRequisitionsByFilter(filterString);</pre>
ソートカラム	Customer Name、 Owner (Initiator) Name、 Requisition ID、 Service Name、 Started Date、 Status、 Submit Date
応答 XML	<pre><requisitions totalCount="x" recordSize="y" startRow="z"> <requisition> . . . </requisition> </requisitions></pre>

要求エントリ

エリア	例
コア API	<p>Id による要求エントリの取得</p> <p>現行ユーザに対して RBAC チェックが適用されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/transaction/requisitionentries/id/<reqEntryId></code></p> <p>Java の例 :</p> <pre>RequisitionEntry requisitonEntry = NSApiClient.getTransaction().getRequisitionEntryById(<reqEntryId>);</pre> <hr/> <p>Requisition Id による要求エントリの取得</p> <p>現行ユーザに対して RBAC チェックが適用されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/transaction/requisitionentries/RequisitionNumber=<reqId></code></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put("startRow", "1"); //optional paramsMap.put("recordSize","10"); //optional String filterString = "RequisitionNumber=" + <reqId>; RequisitionEntryList requisitonEntries = NSApiClient.getTransaction().getRequisitionEntries(paramsMap, filterString);</pre>
フィルタ	N/A
ソートカラム	Due On、 Requisition Entry ID
応答 XML	<pre><requisitionEntries totalCount="x" recordSize="y" startRow="z"> <requisitionEntry> ... </requisitionEntry> </requisitionEntries></pre>

承認

エリア	例
コア API	<p>現行ユーザに対する承認の取得</p> <p>次に示すデフォルトのフィルタを使用して承認を取得します。</p> <p>ViewName = Authorizations for Self</p> <p>Status = Ongoing</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/transaction/authorizations</p> <p>Java の例 :</p> <pre>AuthorizationList authorizations = NSApiClient.getTransaction().getAuthorizations(null);</pre> <hr/> <p>Id による承認の取得</p> <p>現行ユーザに対して RBAC チェックが適用されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/transaction/authorizations/id/<taskId></p> <p>Java の例 :</p> <pre>Authorization authorizations = NSApiClient.getTransaction().getAuthorizationsById(<taskId>);</pre> <hr/> <p>Id による関連承認タスク (承認チェーン) の取得</p> <p>現行ユーザに対して RBAC チェックが適用されます。</p> <p>渡された Id が、要求レベルの承認タスク (部門承認、部門確認、財務承認) の Id であった場合、その特定の要求に属するすべての要求レベルの承認タスクが返されます。</p> <p>渡された Id が、要求エン트리 レベルの承認タスク (サービス グループ承認、サービス グループ確認) の Id であった場合、その特定の要求エントリに属するすべての要求エントリ レベルの承認タスクが返されます。</p> <p>デフォルトのソートは、Due On の降順で行われます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/transaction/authorizations?taskId=<taskId></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put("startRow", "" + 1); //optional paramsMap.put("recordSize", "" +10); //optional paramsMap.put("sortBy", "dueOn"); //optional paramsMap.put("sortDir", "asc"); //optional paramsMap.put("taskId", "<taskId>"); Authorization authorizations = NSApiClient.getTransaction().getAuthorizations (paramsMap);</pre>

エリア	例
フィルタ	<p>ビューおよびステータス フィルタ</p> <p>フィルタに使用可能なビューおよびステータスは、My Services の [Authorizations] タブにあるものに対応しています。</p> <p>Status が指定されなかった場合は、「Ongoing」に設定されます。</p> <p>ViewName が指定されなかった場合は、「Authorizations for Self」に設定されます。</p> <p>ViewName に指定可能な値：</p> <ul style="list-style-type: none"> • Authorizations for Self：現行ユーザに割り当てられている承認 • Assigned and Unassigned Authorizations for Self：現行ユーザ、または現行ユーザがアクセスできるキューに割り当てられている承認 • Authorizations for Others：現行ユーザが属している OU 内の人に割り当てられている承認（ユーザが「See Authorizations for My Business Units」機能を持っている場合に限り、このビューはデータを返します） <p>Status に指定可能な値：Ongoing、Approved、Rejected、Canceled、Reviewed、All。</p> <p>誤った値が指定された場合、デフォルトの値（「Ongoing」および「Authorizations for Self」）が使用されます。</p> <p>REST URL：</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/transaction/authorizations/ViewName=<viewName>[AND Status=<status>]</code></p> <p>Java の例：</p> <pre>String filterString = "ViewName=<viewName> AND Status=<status>"; AuthorizationList authorizations = NSApiClient.getTransaction().getAuthorizationsWithFilters(filterString);</pre>
ソート カラム	Customer Name、Due On、Performer Name、Priority、Requisition ID
応答 XML	<pre><authorizations totalCount="x" recordSize="y" startRow="z" /> <authorization> .. </authorization> </authorizations></pre>

タスク

エリア	例
コア API	<p>現行ユーザに対するタスクの取得</p> <p>次に示すデフォルトのフィルタを使用してタスクを取得します。</p> <p>ViewName = AvailableWork</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/transaction/tasks</p> <p>Java の例 :</p> <pre>TaskList tasks = NSApiClient.getTransaction().getDeliveryTasks(null);</pre> <hr/> <p>Id によるタスクの取得</p> <p>現行ユーザに対して RBAC チェックが適用されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/transaction/tasks/id/<taskId></p> <p>Java の例 :</p> <pre>TaskFull tasks = NSApiClient.getTransaction().getDeliveryTaskById(<taskId>);</pre>
	<p>Requisition Entry Id によるタスクの取得</p> <p>現行ユーザに対して RBAC チェックが適用されます。</p> <p>デフォルトのソートは、Activity Id の昇順で行われます。</p> <p>パラメータおよび指定可能な値 :</p> <ul style="list-style-type: none"> • showSkippedTasks : false (デフォルト)、true • taskType : all (デフォルト)、delivery、authorization • showNestedTasks : false、true (デフォルト) <p>このパラメータが true に設定されている場合、親と子のネスト階層が XML 構造内に保持された状態で配信タスクが返されます。</p> <ul style="list-style-type: none"> • showChildDeliveryPlan : false (デフォルト)、true <p>これは、バンドル サービスにのみ適用されます。このパラメータが true に設定されている場合は、すべての組み込みサービスの配信タスクも返されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/transaction/tasks/RequisitionEntryNumber=<reqEntryId>?taskType=<taskType>&showSkippedTasks=<false true>&showNestedTasks=<false true>&showChildDeliveryPlan=<false true></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put("taskType", "delivery"); //optional paramsMap.put("showSkippedTasks", "true"); //optional paramsMap.put("showNestedTasks", "true"); //optional paramsMap.put("sortBy", "dueOn"); //optional paramsMap.put("sortDir", "desc"); //optional String filterString = "RequisitionEntryNumber=" + <reqEntryId>; TaskList tasks = NSApiClient.getTransaction().getAuthAndDeliveryTasks(paramsMap, filterString);</pre>

エリア	例
	<p>Requisition Id によるマイルストーンの取得</p> <p>現行ユーザに対して RBAC チェックが適用されます。</p> <p>配信プロセスのマイルストーン（確認、承認、配信プロジェクト）が、時系列順に返されます。ソートおよびページングはサポートされていません。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/transaction/tasks/RequisitionNumber=<reqId></code></p> <p>Java の例 :</p> <pre>String filterString = "RequisitionNumber=" + <reqId>; MilestoneList milestones = NSApiClient.getTransaction().getDeliveryProcessForMilestone (filterString);</pre>
特殊条件	<p>タスクの許可</p> <p>REST URL にアクションおよびタスク Id を指定して HTTP POST を実行します。</p> <p>POST REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/transaction/tasks /<taskId>/approve</code></p> <p>Java の例 :</p> <pre>TaskAction approve = NSApiClient.getTransaction().approveTask(<taskId>;</pre> <hr/> <p>タスクの拒否</p> <p>REST URL にアクションおよびタスク Id を指定して HTTP POST を実行します。</p> <p>POST REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/transaction/tasks/<taskId>/reject</code></p> <p>Java の例 :</p> <pre>TaskAction Reject = NSApiClient.getTransaction().rejectTask(<taskId>;</pre>
	<p>タスクの完了</p> <p>REST URL にアクションおよびタスク Id を指定して HTTP POST を実行します。</p> <p>POST REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/transaction/tasks/<taskId>/done</code></p> <p>Java の例 :</p> <pre>TaskAction Complete = NSApiClient.getTransaction().completeTask(<taskId>;</pre> <hr/> <p>タスクの確認</p> <p>REST URL にアクションおよびタスク Id を指定して HTTP POST を実行します。</p> <p>POST REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/transaction/tasks/<taskId>/review</code></p> <p>Java の例 :</p> <pre>TaskAction Review = NSApiClient.getTransaction().reviewTask(<taskId>;</pre>

エリア	例
フィルタ	<p>ビュー フィルタ</p> <p>フィルタに使用可能なビューは、Service Manager モジュールのビューに対応していません。ユーザ定義ビューは使用できません。</p> <p>ViewName フィルタが指定されなかった場合、「AvailableWork」に設定されます。</p> <p>ViewName に指定可能な値：AvailableWork、MyWork、MyLateWork、WorkForeCast</p> <p>誤った値が指定された場合は、デフォルト値が使用されます。</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/transaction/tasks?viewName=<viewName></p> <p>Java の例 :</p> <pre>MultiValueMap paramsMap = new MultiValueMap(); paramsMap.put("ViewName", "<viewName>"); TaskList tasks = NSApiClient.getTransaction().getDeliveryTasks(paramsMap);</pre>
ソート カラム	<p>Activity ID、Completed On、Customer Name、Customer OU Name、Due On、Effort、Initiator Name、Performer Name、Priority、Requisition ID、Scheduled Start Date、Service Name、Task Name、Task Type</p>
応答 XML	<pre><tasks totalCount="x" recordSize="y" startRow="z"> <task> .. </task> </tasks></pre>

LifecycleCenter データ

サービス項目の詳細

エリア	例
コア API	<p>Name による取得</p> <p>指定したサービス項目名に関して現行ユーザに割り当てられている、サービス項目インスタンスの詳細情報およびサブスクリプション データが返されます。</p> <p>また、ユーザが「View Service Items for My Business Units」機能を持っている場合は、そのユーザが属している全 OU に含まれている人に関するサービス項目が返されます。</p> <p>サービス項目名パラメータは、[Design Service Item] ページの [Name] フィールドに示されるサービス項目の内部テーブル名 (SiVirtualMachine など) を受け入れます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/serviceitem/<serviceName></code></p> <p>Java の例 :</p> <pre>ServiceItemDTO serviceItems = NSApiClient.getServiceItem().getServiceItemData("<serviceName>", null);</pre> <hr/> <p>ビュー フィルタ</p> <p>フィルタに使用可能なビューは、My Services および Service Item Manager モジュールの表示内容に対応しています。</p> <p>ViewName に指定可能な値 :</p> <ul style="list-style-type: none"> • My ServiceItems : 現行ユーザが所有しているサービス項目のすべてのインスタンス「View Service Items for My Business Units」機能を持っているユーザは、そのユーザが属している OU 内の他人が所有している項目も表示できます。 • Manage ServiceItems : サービス項目のすべてのインスタンス (ユーザが「Manage Service Item Instances」機能を持っている場合に限り、このビューはデータを返します)。 <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/serviceitem/<serviceName>/View%20Name=<viewName></code></p> <p>Java の例 :</p> <pre>ServiceItemDTO serviceItems = NSApiClient.getServiceItem().getServiceItemDataWithFilters("<serviceName>", "<viewName>");</pre>

エリア	例
フィルタ	<p>サービス項目属性フィルタ</p> <p>最大 3 つのフィルタがサポートされます。</p> <p>サービス項目およびサブスクリプションのすべてのカラムがサポートされます。</p> <p>サービス項目分類名（グループ）またはサービス項目タイプ（シスコが予約しているもの、またはユーザ定義のもの）によるフィルタはサポートされていません。</p> <p>REST URL :</p> <p>比較演算子 :</p> <p>数値 / 日付カラム : =、>、<、>=、<=</p> <p>文字列カラム : = (大文字小文字の区別あり。like、contains、および starts-with 演算子については後述)</p> <p>関係演算子 : AND、OR (大文字小文字の区別あり)</p> <p>区切り文字 = </p> <p>使用可能なカラム : サービス項目およびサブスクリプションの全カラム</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/serviceitem/<serviceName>/<columnName1><operator1><value1>[<AND OR> <columnName2><operator2><value2>][<AND OR> <columnName3><operator3><value3>]</code></p> <p>Java の例 :</p> <pre>String filter = "<columnName1><operator1><value1> <and or> <columnName2><operator2><value2> <and or> <columnName3><operator3><value3>"; ServiceItemDTO serviceItems = NSApiClient.getServiceItem().getServiceItemDataWithFilters("<serviceName>", filter);</pre>
	<p>文字列カラムのフィルタ</p> <p>% (starts with) 演算子用のフィルタ。</p> <p>* (contains) 演算子用のフィルタ。</p> <p>(ends with) 演算子はサポートされていません。</p> <p>例 :</p> <p>Name=service*</p> <p>Name=*g*</p> <p>Name=*g : 許可されません。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/serviceitem/<serviceName>/<columnName>=<wildcardValue></code></p> <p>Java の例 :</p> <pre>ServiceItemDTO serviceItems = NSApiClient.getServiceItem().getServiceItemDataWithFilters("<serviceName>",</pre>

エリア	例
	<p>日付カラムの属性</p> <p>日付フィールドの値は mm-dd-yyyy 形式にする必要があります。</p> <p>比較演算子 : =、>、<、<=、>=</p> <p>例 :</p> <p>SubmittedDate=12-10-2010</p> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/serviceitem/<serviceName>/<columnName><operator><mm-dd-yyyy></p> <p>Java の例 :</p> <pre>ServiceItemDTO serviceItems = NSApiClient.getServiceItem().getServiceItemDataWithFilters("<serviceName> ", "<columnName><operator><mm-dd-yyyy>");</pre>
ソートカラム	<p>サービス項目の属性 : すべてのテーブルカラム。</p> <p>サブスクリプション : Assigned Date、Display Name、ID (内部 ID)、Requisition ID、Submit Date。</p>
応答 XML	<pre><serviceitem totalCount="x" recordSize="3" startRow="1" id="62"> <logicName></logicName> <name></name> <subscription> .. . <assignedDate> </assignedDate> <assignedDateRaw></assignedDateRaw> <customerID></customerID> <customerName> </customerName> <displayName> </displayName> <id></id> <organizationalUnitID></organizationalUnitID> <organizationalUnitName> </organizationalUnitName> <requisitionEntryID></requisitionEntryID> <requisitionID></requisitionID> <serviceItemClassificationID></serviceItemClassificationID> <serviceItemTypeID></serviceItemTypeID> <serviceItemTypeName> </serviceItemTypeName> <submittedDate> </submittedDate> <submittedDateRaw></submittedDateRaw> .. . </subscription> </serviceitem></pre>

すべてのサービス項目

エリア	例
コア API	<p>すべての項目の取得</p> <p>すべてのサブスクリプションおよびサービス項目データを表示します。</p> <p>デフォルトでは、現行ユーザに割り当てられているサービス項目のみがフィルタされます。また、ユーザが「View Service Items for My Business Units」機能を持っている場合は、そのユーザが属している全 OU に含まれている他人のサービス項目も返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/serviceitems/serviceitemsubscription</code></p> <p>Java の例 :</p> <pre>ServiceItemSubscriptionList AllserviceItems = NSApiClient.getServiceItemSubscription().getServiceItemSubscriptionData (null) ;</pre>
フィルタ	<p>ビュー フィルタ</p> <p>フィルタに使用可能なビューは、My Services および Service Item Manager モジュールのそれぞれに表示される内容に対応しています。</p> <p>ViewName に指定可能な値 :</p> <ul style="list-style-type: none"> • My ServiceItems : 現行ユーザが所有しているすべてのサービス項目「View Service Items for My Business Units」を持っているユーザの場合は、そのユーザが属している OU 内の他人が所有している項目も取得します。 • Manage ServiceItems : すべてのサービス項目 (ユーザが「Manage Service Item Instances」機能を持っている場合に限り、このビューはデータを返します)。 <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/serviceitems/serviceitemsubscription/View%20Name=<viewName></code></p> <p>Java の例 :</p> <pre>ServiceItemSubscriptionList AllserviceItems = NSApiClient.getServiceItemSubscription().getServiceItemSubscriptionFilterData ("<viewName>");</pre>

エリア	例
	<p>サブスクリプション フィルタ</p> <p>Service Item Subscription テーブルの全カラム用のフィルタ。 最大 3 つのフィルタがサポートされます。</p> <p>REST URL :</p> <p>比較演算子 :</p> <ul style="list-style-type: none"> • 数値/日付カラム : =、>、<、>=、<= • 文字列カラム : = (大文字小文字の区別あり。like、contains、および starts-with 演算子については後述) <p>関係演算子 : AND、OR (大文字小文字の区別あり)</p> <p>フィルタ区切り文字 : </p> <p>サポートされるカラム : すべてのカラム</p> <p>http://<ServerURL>/RequestCenter/nsapi/serviceitems/serviceitemsubscription/<columnName1><operator1><value1>[<AND OR> <columnName2><operator2><value2>][<AND OR> <columnName3><operator3><value3>]</p> <p>Java の例 :</p> <pre>String filter = " <columnName1><operator1><value1> <and or> <columnName2><operator2><value2> <and or> <columnName3><operator3><value3>"; ServiceItemSubscriptionList AllserviceItems = NSApiClient.getServiceItemSubscription().getServiceItemSubscriptionFilterData (filter);</pre>
	<p>文字列カラムのフィルタ</p> <p>% (starts with) 演算子がサポートされています。</p> <p>* (contains) 演算子がサポートされています。</p> <p>(ends with) 演算子はサポートされていません。</p> <p>例 :</p> <pre>Name=service* Name=*g* Name=*g : 許可されません</pre> <p>REST URL :</p> <p>http://<ServerURL>/RequestCenter/nsapi/serviceitems/serviceitemsubscription/<columnName>=<wildcardValue></p> <p>Java の例 :</p> <pre>String filter = "<columnName>=<wildcardValue>"; ServiceItemSubscriptionList AllserviceItems = NSApiClient.getServiceItemSubscription().getServiceItemSubscriptionFilterData (filter);</pre>

エリア	例
	<p>日付カラムのフィルタ</p> <p>日付フィールドの値は mm-dd-yyyy 形式にする必要があります。</p> <p>比較演算子 : =、>、<、<=、>=</p> <p>例 :</p> <pre>SubmittedDate=12-10-2010</pre> <p>REST URL :</p> <pre>http://<ServerURL>/RequestCenter/nsapi/serviceitems/serviceitemsubscription /<columnName><operator><mm-dd-yyyy></pre> <p>Java の例 :</p> <pre>String filter = "<columnName><operator><mm-dd-yyyy>"; ServiceItemSubscriptionList AllserviceItems = NSApiClient.getServiceItemSubscription().getServiceItemSubscriptionFilterData (filter);</pre>
ソートカラム	<p>Assigned Date、Customer ID、Display Name、Organizational Unit ID、Requisition ID、Requisition Entry ID、Service Item Classification ID、Service Item ID、Service Item Type ID、Service Item Type Name、Submitted Date</p>
応答 XML	<pre><AllServiceItems totalCount="x" recordSize="y" startRow="z"> <serviceitemsubscription displayName=" " id=""> .. <serviceItemTypeName> </serviceItemTypeName> <organizationalUnitID></organizationalUnitID> <assignedDate> </assignedDate> <requisitionID></requisitionID> <submittedDate> </submittedDate> <submittedDateRaw></submittedDateRaw> <assignedDateRaw></assignedDateRaw> <customerID></customerID> <requisitionEntryID></requisitionEntryID> <serviceItemID></serviceItemID> <serviceItemTypeID></serviceItemTypeID> <organizationalUnitName> </organizationalUnitName> <customerName> </customerName> <serviceitem id=""> <logicName> </logicName> <name> </name> <serviceItemData rowId=""> <serviceItemAttribute name=" " > </serviceItemAttribute> .. </serviceitemsubscription> </AllServiceItems></pre>

標準

エリア	例
コア API	<p>Name による取得</p> <p>デフォルトでは、指定した標準の最初の 50 個のエントリが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/standard/<standardName></code></p> <p>Java の例 :</p> <pre>StandardDTO standards = NSApiClient.getStandard().getStandardData("<standardName>", null);</pre>
フィルタ	<p>最大 3 つのフィルタがサポートされます。</p> <p>REST URL :</p> <p>比較演算子 :</p> <ul style="list-style-type: none"> 数値/日付カラム : =、>、<、>=、<= (大文字小文字の区別あり) 文字列カラム : = (like、contains、および starts-with 演算子については後述) <p>関係演算子 : AND、OR (大文字小文字の区別あり)</p> <p>フィルタ区切り文字 : </p> <p>サポートされるカラム : すべてのカラム</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/standard/<standardName>/<columnName1><operator1><value1>[<AND OR> <columnName2><operator2><value2>][<AND OR> <columnName3><operator3><value3>]</code></p> <p>Java の例 :</p> <pre>String filter: "<columnName1><operator1><value1> <and or> <columnName2><operator2><value2> <and or> <columnName3><operator3><value3>"; StandardDTO standards = NSApiClient.getStandard().getStandardDataWithFilters("<standardName>", filter);</pre>

エリア	例
	<p>文字列カラムのフィルタ</p> <p>% (starts with) 演算子がサポートされています。</p> <p>* (contains) 演算子がサポートされています。</p> <p>(ends with) 演算子はサポートされていません。</p> <p>例 :</p> <pre>Name=service* Name=*g* Name=*g : 許可されません</pre> <p>REST URL :</p> <pre>http://<ServerURL>/RequestCenter/nsapi/standard/<standardName>/<columnName>=<wildcardName></pre> <p>Java の例 :</p> <pre>StandardDTO standards = NSApiClient.getStandard().getStandardDataWithFilters("<standardName>", "<wildcardName>");</pre>
	<p>日付カラムのフィルタ</p> <p>日付フィールドの値は mm-dd-yyyy 形式にする必要があります。</p> <p>比較演算子 : =、>、<、<=、>=</p> <p>例 :</p> <pre>SubmittedDate=12-10-2010</pre> <p>REST URL :</p> <pre>http://<ServerURL>/RequestCenter/nsapi/standard/<standardName>/<columnName><operator><mm-dd-yyyy></pre> <p>Java の例 :</p> <pre>StandardDTO standards = NSApiClient.getStandard().getStandardDataWithFilters("<standardName>", "<columnName><operator><mm-dd-yyyy>");</pre>
ソートカラム	すべてのテーブルカラム。
応答 XML	<pre><standard totalCount="x" startRow="y" recordSize="z" id="a"> <loginName></loginName> <name></name> <standardData rowId=""> .. . <standardAttribute name="id" /> .. . <standardURL><a ></standardURL> <standardURLOnly> </standardURLOnly> .. . </standardData> </standard></pre>

Service Portal データ

カスタム コンテンツ

エリア	例
コア API	<p>Name による取得</p> <p>デフォルトでは、指定したテーブルの最初の 50 個のエントリが返されます。</p> <p>REST URL :</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/customcontent/<customContentName></code></p> <p>Java の例 :</p> <pre>CustomContentDTO customs = NSApiClient.getCustomContent().getContentData("<customContentName>", null);</pre>
フィルタ	<p>すべてのカスタム コンテンツ カラムがサポートされています。</p> <p>% (starts with) 演算子がサポートされています。</p> <p>(ends with) 演算子はサポートされていません。</p> <p>例 :</p> <pre>Name=custom* Name=*g : 許可されません</pre> <p>REST URL :</p> <p>比較演算子 : =、>、<、>=、<= (大文字小文字の区別あり)</p> <p>関係演算子 : AND、OR (大文字小文字の区別あり)</p> <p>フィルタ区切り文字 : </p> <p>サポートされるカラム : すべてのカラム</p> <p><code>http://<ServerURL>/RequestCenter/nsapi/customcontent/<customContentName>/<columnName1><operator1><value1>[<AND OR> <columnName2><operator2><value2>][<AND OR> <columnName3><operator3><value3>]</code></p> <p><code>http://<ServerURL>/RequestCenter/nsapi/customcontent/<customContentName>/<columnName>=<wildcardName></code></p> <p><code>http://<ServerURL>/RequestCenter/nsapi/customcontent/<customContentName>/<columnName><operator><mm-dd-yyyy></code></p> <p>Java の例 :</p> <pre>String filter = "<columnName><operator><value>"; CustomContentDTO customs = NSApiClient.getCustomContent().getCustomContentDataWithFilters("CoPortalContent", filter);</pre>

エリア	例
ソートカラム	すべてのテーブルカラム
応答 XML	<pre><customContent totalCount="x" startRow="y" recordSize="z" id="a"> <logicName></logicName> <name></name> .. <customContentData rowId=""> <customContentAttribute name=" "></customContentAttribute> .. </customContentData> </customContent></pre>

全カラム名のリスト、および各エンティティの説明については、『Cisco Service Portal Designer Guide』の「Portal Manager」の章にある「Reference Data」の項を参照してください。

エラーメッセージ

nsAPI は、次のように、例外の性質に応じてさまざまな HTTP 応答コードを返します。

- HTTP ステータス コード **401** (Unauthorized) および XML エラー応答メッセージ「**User does not have proper authentication.**」: 認証パラメータが無効であるか、または指定されていません。
- HTTP ステータス コード **404** (Not Found) および XML エラー応答メッセージ「**Requested resource could not be found.**」: 指定されたパラメータ/URL 値のデータを取得できませんでした。

例:

```
nsapi/directory/people/id/-1
nsapi/directory/people/id/foo
nsapi/directory/people/id/1000 (id が 1000 の個人が存在しない場合)
nsapi/directory/people/name/<non-existent person>
nsapi/directory/people/idxyz/1
```

- HTTP ステータス コード **403** (Forbidden) および XML エラー応答メッセージ「**The user does not have sufficient permissions to perform the operation this object.**」: 指定された操作をオブジェクトに対して実行するために必要な権限をユーザが持っていないため、データを取得できませんでした。
- HTTP ステータス **500** (内部エラー) および XML エラー応答メッセージ「**Internal Error: Invalid parameter values specified or unexpected error.**」: パラメータの誤り。あるいは、nsAPI の内部で発生したその他の例外。または、その他の一般的なサーバエラー。

例:

```
nsapi/directory/people?startRow=5000 (5000 行存在していない場合)
nsapi/directory/people?sortBy=wrongColumn&sortDir=Asc (サポートされていないカラム)
nsapi/directory/people?recordSize=-1 (recordSize が負またはゼロの値)
```

- Http ステータス **422** (Unprocessable Entity)、および POST/UPDATE 操作に対する XML 妥当性検査エラー応答メッセージ: 要求データの必須フィールドに値がありません。または無効な値が入っています。

nsAPI は、指定されたフィルタに対する結果が存在しない場合には、エラーを返しません。次に例を示します。

```
nsapi/directory/people?name=<non-existent person>
```

nsAPI でのメソッドの実行中に発生したすべての例外について、nsAPI は Java から NSAPIException をスローします。

クイック リファレンス

次のチャートに、さまざまなエンティティタイプでサポートされる操作についての要約を示します。

エンティティ名	Id による取得	Name による取得	すべて取得	ワイルドカード名前検索	ソート	ページング	REST フィルタから Java クライアントメソッドへの変換	更新	フィルタ	ネストされたエンティティ
認証										
認証 : ログイン/ログアウト					○		○			
定義データ										
カテゴリ	○	○	○	○	○	○	○		カタログタイプ	サブカテゴリ、組み込みサービス、組み込み提供
サービス	○	○	○	○	○	○	○		カテゴリ ID、カテゴリ名、キーワード	カテゴリ、キーワード、組み込みサービス
提供サービス	○	○	○	○	○	○	○		名前	カテゴリ、キーワード、目的、コストドライバ、コンポーネントサービス
エージェント	○	○	○	○	○	○	○		名前	
トランザクション データ										
契約	○	○	○	○	○	○	○		名前	
要求	○		○		○	○	○		ビュー名、ステータス	
要求エントリ	○		○		○	○				
認可	○		○		○	○	○		ビュー名、ステータス	
作業	○		○		○	○	○		ビュー名	
タスク : 特定の要求エントリに関するタスク			○		○	○	○		タスクのタイプ、ステータス (Skipped)	組み込みサービスのタスク
タスク : 特定の要求に関するマイルストーン			○							

タスク：アクション								○		
ディレクトリ データ										
組織単位	○	○	○	○	○	○	○		OU のタイプ	
個人	○	○	○	○	○	○	○	○	OU 名、グループ名、ロール名	OU、グループ、ロール、住所、連絡先、設定、代理人
グループ	○	○	○	○	○	○	○		名前	
アカウント	○	○	○	○	○	○	○		名前	
Lifecycle Center データ										
すべてのサービス項目			○		○	○			任意のカラム	
サービス項目			○		○	○	○		任意のカラム	
標準			○		○	○	○		任意のカラム	
Service Portal データ										
カスタム コンテンツ		○	○	○	○	○	○		任意のカラム	

