



CHAPTER 11

PAK 管理関数

この章では、次の Product Authorization Key (PAK; 製品認証キー) 管理関数に関する情報を示します。

- 「[asyncDownloadPAKInfo](#)」 (P.11-1)
- 「[createPAKs](#)」 (P.11-3)
- 「[deletePAKs](#)」 (P.11-4)
- 「[listAllFoldersByPAK](#)」 (P.11-5)
- 「[listAllPAKsInFolder](#)」 (P.11-6)
- 「[listPAKContainFeatures](#)」 (P.11-7)
- 「[readPAKs](#)」 (P.11-7)
- 「[writePAKs](#)」 (P.11-9)

asyncDownloadPAKInfo

構文

```
String asyncDownloadPAKInfo(UserToken token, String jobGroup, String[] pakIDs, IDStatusListener listener) throws RemoteException;
```

説明

この関数は、特定の PAK ID に関連付けられた情報を Cisco Product License Registration Portal からダウンロードし、その情報をインベントリに格納します。

この関数はノンブロッキングであり、呼び出し元に即座に要求 ID を返します。この関数を呼び出しているとき、クライアントプログラムは `StatusListener` インターフェイスを実装するリスナー オブジェクトを提供します。関数が完了すると、リスナー オブジェクトの `onStatus()` メソッドが呼び出されません。

入力パラメータ

パラメータ	タイプ	値	説明
token	UserToken、必須	—	ユーザの認証パスを表すトークン。これは、ユーザが login 関数を呼び出して、バックエンドサーバがそのユーザを認証した後に取得されます。
jobGroup	String、必須	—	この非同期ジョブが属するグループ。null に設定した場合は、返されたジョブ ID がジョブグループ名として使用されます。
pakIDs	String の配列、必須	—	使用する PAK を識別する PAK ID の配列。PAK ID を生成するには、createPAK 関数を使用し、パラメータとして PAK 名を渡して PAK ID を取得します。
listener	IDStatusListener オブジェクト、必須	—	IDStatusListener インターフェイスを実装するオブジェクト。

戻り値

この関数は呼び出し元に要求 ID を返します。操作が完了したら、onStatus() メソッドの IDStatus 入力パラメータを使用して操作ステータスをチェックできます。IDStatus には IDStatusItem のリストが含まれます。各 IDStatusItem には、PAK ID、この PAK ID に対する操作のエラーコード、およびエラーメッセージが含まれます。

エラーと例外

システムエラーによって操作が完了しなかった場合は、RemoteException がスローされます。

IDStatusListener の onStatus() メソッドの入力パラメータには、エラーコードとエラーメッセージが含まれます。IDStatus オブジェクトには複数のエラーコードとエラーメッセージが含まれる場合があります。次の例では、onStatus() メソッドでエラーコードとエラーメッセージを取得しています。

```
public void onStatus(String request_id, IDStatus status) {

    // The general error code of the operation.
    int err_code = status.getErrorCode();

    // The general error message of the operation.
    String err_msg = status.getErrorMessage();

    // A list of status for each individual element in the
    // bulk operation.
    IDStatusItem[] items = status.getIDStatusItems()

    // Iterate through the list to get individual status.
    for (int i = 0; i < items.length(); i++) {

        // Get the individual object ID returned by the operation.
        String id = items[i].getID();

        // Get the individual error code corresponding to
        // the object ID.
        int item_err_code = items[i].getErrorCode();

        // Get the individual error message corresponding to
```

```

        // the object ID.
        String item_err_msg = items[i].getErrorMessage();
    }
}

```

createPAKs

構文

```

PAKStatus createPAKs(UserToken token,String[] pakNames, String folder)
PAKStatus createPAKs(UserToken token,PAK[] paks, String folder)

```

説明

この関数は、インベントリ内の指定したフォルダに PAK オブジェクトを作成します。フォルダが null の場合、新しく作成された PAK はデフォルト フォルダに追加されます。

入力パラメータ

パラメータ	タイプ	値	説明
token	UserToken、必須	—	ユーザの認証パスを表すトークン。これは、ユーザが login 関数を呼び出して、バックエンドサーバがそのユーザを認証した後に取得されます。
pak_ids	String の配列、必須	x21 ~ x7A の範囲の ASCII 文字を含む最大 64 文字の文字列	PAK オブジェクトの各インスタンスの名前として使用される PAK ID の配列。
folder	String、必須	—	フォルダの名前。 (注) この関数にはフォルダ名を必要としない形式もあります。
paks	PAK の配列、必須	PAK オブジェクト	PAK オブジェクトの配列。
pakNames		x21 ~ x7A の範囲の ASCII 文字を含む最大 64 文字の文字列	PAK 名の配列。

戻り値

この関数は PAKStatus オブジェクトを返します。次の例では、ステータスに含まれるエラー コード、エラー メッセージ、および戻りオブジェクトを取得しています。

```

PAKStatus status = createPAKs(..., ...);

// The general error code of the operation.
int err_code = status.getErrorCode();

// The general error message of the operation.
String err_msg = status.getErrorMessage();

// A list of status for each individual element in the

```

```

// bulk operation.
PAKStatusItem[] items = status.getPAKStatusItems()

// Iterate through the list to get individual status.
for (int i = 0; i < items.length(); i++) {

// Get the individual object returned by the operation.
PAK pak = items[i].getPAK();

// Get the individual error code corresponding to
// the object.
int item_err_code = items[i].getErrorCode();

// Get the individual error message corresponding to
// the object.
String item_err_msg = items[i].getErrorMessage();
}

```

エラーと例外

システム エラーによって操作が完了しなかった場合は、`RemoteException` がスローされます。

入力配列の要素でエラーが発生した場合は、エラーに関する情報を含む `Status` オブジェクトが返されます。

deletePAKs

構文

```
IDStatus deletePAKs(UserToken token, String[] pak_ids) throws RemoteException;
```

説明

この関数は、指定した PAK ID を使用して、インベントリから PAK オブジェクトを削除します。PAK ID は、`createPAKs` 関数または `readPAKs` 関数から取得されます。この関数はさらに、PAK が見つかったフォルダから PAK を削除します。

入力パラメータ

パラメータ	タイプ	値	説明
token	UserToken、必須	—	ユーザの認証パスを表すトークン。これは、ユーザが <code>login</code> 関数を呼び出して、バックエンド サーバがそのユーザを認証した後に取得されます。
pak_ids	String の配列、必須	—	PAK ID の配列。

戻り値

この関数は `IDStatus` オブジェクトを返します。次の例では、ステータスに含まれるエラー コード、エラー メッセージ、および戻りオブジェクトを取得しています。

```

IDStatus status = deletePAKs(..., ...);

// The general error code of the operation.
int err_code = status.getErrorCode();

```

```

// The general error message of the operation.
String err_msg = status.getErrorMessage();

// A list of status for each individual element in the
// bulk operation.
IDStatusItem[] items = status.getIDStatusItems()

// Iterate through the list to get individual status.
for (int i = 0; i < items.length(); i++) {

// Get the individual ID returned by the operation.
String id = items[i].getID();

// Get the individual error code corresponding to
// the ID.
int item_err_code = items[i].getErrorCode();

// Get the individual error message corresponding to
// the ID.
String item_err_msg = items[i].getErrorMessage();
}

```

エラーと例外

システム エラーによって操作が完了しなかった場合は、`RemoteException` がスローされます。

入力配列の要素でエラーが発生した場合は、エラーに関する情報を含む `Status` オブジェクトが返されます。

listAllFoldersByPAK

構文

```
String[] listAllFoldersByPAK(UserToken token, String[] pak_id) throws RemoteException;
```

説明

この関数は、PAK が属するすべてのフォルダのリストを返します。

入力パラメータ

パラメータ	タイプ	値	説明
token	UserToken、必須	—	ユーザの認証パスを表すトークン。これは、ユーザが <code>login</code> 関数を呼び出して、バックエンドサーバがそのユーザを認証した後に取得されます。
pak_id	String の配列、必須	—	PAK オブジェクトの各インスタンスの ID として使用される PAK ID の配列。

戻り値

この関数は、指定した PAK が属するフォルダの配列を返します。指定した PAK が存在しない場合は、サイズが 0 の非 `null` 配列が返されます。

エラーと例外

システム エラーによって操作が完了しなかった場合は、`RemoteException` がスローされます。

エラーが発生した場合、この関数は `null` を返します。

listAllPAKsInFolder

構文

```
String[] listAllPAKsInFolder(UserToken token, String folder, PageInfo pageinfo) throws
RemoteException;
```

説明

この関数は、ステータス、総数、および指定したフォルダに属する PAK ID の配列を含む `IDPagingInfo` オブジェクトを返します。フォルダが `null` の場合は、デフォルト フォルダに含まれる PAK ID が返されます。

有効なオフセットと最大数を指定してページ分割オプションを設定できます。次の例は、オフセットと最大数を指定してページ分割を設定する方法を示します。

1 ページ目を取得するには、次のようにします。

```
PageInfo p = new PageInfo(0, 10)
```

5 ページ目を取得するには、次のようにします。

```
PageInfo p = new PageInfo(50, 10)
```

ページ分割せずにすべてのレコードを取得するには、次のようにします。

```
PageInfo p = new PageInfo(0, -1)
```

入力パラメータ

パラメータ	タイプ	値	説明
token	UserToken、必須	—	ユーザの認証パスを表すトークン。これは、ユーザが <code>login</code> 関数を呼び出して、バックエンドサーバがそのユーザを認証した後に取得されます。
folder	String、必須	—	PAK フォルダの名前。グループ化されていない PAK を検索する場合は <code>null</code> 。
pageinfo	DataObject、必須	—	<code>offset</code> と <code>max</code> を含むページ分割オブジェクト。 <code>offset</code> は、この関数によって取得する、最初のレコードセットからの相対的なオフセットを指定します。 <code>max</code> は、取得するレコードの最大数を指定します。 <code>max</code> を <code>-1</code> に設定すると、すべてのレコードが返されます。

戻り値

この関数は、PAK ID の配列を含む `IDPagingInfo` オブジェクトを返します。`offset` がレコード数よりも大きい場合は、サイズが 0 の配列が返されます。操作エラーが発生した場合、`Status` オブジェクトにはエラー コードとエラー メッセージが含まれます。

エラーと例外

システム エラーによって操作が完了しなかった場合は、`RemoteException` がスローされます。
エラーが発生した場合、この関数は `null` を返します。

listPAKContainFeatures

構文

```
String[] listPAKContainFeatures(UserToken token, String[] features) throws
RemoteException;
```

説明

この関数は、指定したフィーチャ リスト内のすべてのライセンス フィーチャを含む PAK のリストを返します。

入力パラメータ

パラメータ	タイプ	値	説明
token	UserToken、必須	—	ユーザの認証パスを表すトークン。これは、ユーザが <code>login</code> 関数を呼び出して、バックエンド サーバがそのユーザを認証した後に取得されます。
features	String の配列、必須	—	ライセンス フィーチャの名前を表す文字列の配列。この API は、このフィーチャ配列に格納されたすべてのフィーチャを含む (AND の関係) PAK のリストを返します。

戻り値

この関数は文字列の配列を返します。各文字列は 1 つの PAK ID を表します。

エラーと例外

システム エラーによって操作が完了しなかった場合は、`RemoteException` がスローされます。
エラーが発生した場合、この関数は `null` を返します。長さが 0 の配列は、PAK が見つからなかったことを意味します。

readPAKs

構文

```
PAKStatus readPAKs(UserToken token, String[] pak_ids) throws RemoteException;
```

説明

この関数は、指定した PAK ID を使用して、インベントリから PAK オブジェクトの配列を取得します。PAK ID は `createPAKs` 関数から取得されます。

入力パラメータ

パラメータ	タイプ	値	説明
token	UserToken、必須	—	ユーザの認証パスを表すトークン。これは、ユーザが <code>login</code> 関数を呼び出して、バックエンドサーバがそのユーザを認証した後に取得されます。
pak_ids	String の配列、必須	—	PAK ID の配列。

戻り値

この関数は `PAKStatus` オブジェクトを返します。次の例では、ステータスに含まれるエラー コード、エラー メッセージ、および戻りオブジェクトを取得しています。

```
PAKStatus status = readPAKs(..., ...);

// The general error code of the operation.
int err_code = status.getErrorCode();

// The general error message of the operation.
String err_msg = status.getErrorMessage();

// A list of status for each individual element in the
// bulk operation.
PAKStatusItem[] items = status.getPAKStatusItems()

// Iterate through the list to get individual status.
for (int i = 0; i < items.length(); i++) {

// Get the individual object returned by the operation.
PAK pak = items[i].getPAK();

// Get the individual error code corresponding to
// the object.
int item_err_code = items[i].getErrorCode();

// Get the individual error message corresponding to
// the object.
String item_err_msg = items[i].getErrorMessage();
}
```

エラーと例外

システム エラーによって操作が完了しなかった場合は、`RemoteException` がスローされます。

入力配列の要素でエラーが発生した場合は、エラーに関する情報を含む `Status` オブジェクトが返されます。

writePAKs

構文

```
IDStatus writePAKs(UserToken token, PAK[] pak) throws RemoteException;
```

説明

この関数は、指定した PAK オブジェクトをインベントリに書き込みます。入力 PAK オブジェクトには、createPAKs() 関数によって返された新しい PAK のインスタンス、または readPAKs() 関数によってインベントリから取得された既存のインスタンスを指定できます。

入力パラメータ

パラメータ	タイプ	値	説明
token	UserToken、必須	—	ユーザの認証パスを表すトークン。これは、ユーザが login 関数を呼び出して、バックエンドサーバがそのユーザを認証した後に取得されます。
paks	PAK の配列、必須	PAK オブジェクト	PAK オブジェクトの配列。

戻り値

この関数は IDStatus オブジェクトを返します。次の例では、ステータスに含まれるエラー コード、エラー メッセージ、および戻りオブジェクトを取得しています。

```
IDStatus status = writePAK(..., ...);

// The general error code of the operation.
int err_code = status.getErrorCode();

// The general error message of the operation.
String err_msg = status.getErrorMessage();

// A list of status for each individual element in the
// bulk operation.
IDStatusItem[] items = status.getIDStatusItems()

// Iterate through the list to get individual status.
for (int i = 0; i < items.length(); i++) {

// Get the individual ID returned by the operation.
String id = items[i].getID();

// Get the individual error code corresponding to
// the ID.
int item_err_code = items[i].getErrorCode();

// Get the individual error message corresponding to
// the ID.
String item_err_msg = items[i].getErrorMessage();
}
```

エラーと例外

システム エラーによって操作が完了しなかった場合は、`RemoteException` がスローされます。

入力配列の要素でエラーが発生した場合は、エラーに関する情報を含む `Status` オブジェクトが返されます。