



CHAPTER 12

ポリシー管理関数

この章では、次のポリシー管理関数に関する情報を示します。

- 「[asyncExecutePolicy](#)」 (P.12-1)
- 「[createPolicy](#)」 (P.12-3)
- 「[deletePolicy](#)」 (P.12-4)
- 「[enumerateDeviceFilterAttribute](#)」 (P.12-4)
- 「[enumerateSKUFilterAttribute](#)」 (P.12-5)
- 「[listAllPolicies](#)」 (P.12-6)
- 「[listFilteredDevices](#)」 (P.12-6)
- 「[listDeviceIdsByFilter](#)」 (P.12-7)
- 「[listFilteredSKUs](#)」 (P.12-7)
- 「[listApplicableDevicesBySKUs](#)」 (P.12-8)
- 「[readPolicy](#)」 (P.12-9)
- 「[writePolicy](#)」 (P.12-9)

asyncExecutePolicy

構文

```
String asyncExecutePolicy(UserToken token, String jobGroup, String policyID,  
IDStatusListener listener) throws RemoteException;
```

説明

この関数は、指定したポリシーを実行します。指定したポリシーに含まれるデバイスおよび SKU を選択するフィルタを使用し、ライセンスを取得して、デバイスのライセンスを配備します。

この関数は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。操作が完了すると、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。
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。
policyID	String、必須	—	Policy オブジェクトの ID。
listener	IDStatusListener、必須	—	リスナー 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。操作が完了すると、onStatus() メソッドの入力パラメータとしてステータスが提供されます。

リスナーは、IDStatusItem のリストを含む IDStatus を受信します。各 IDStatusItem には、PAK ID + "-::-" + SKU 名 + "-::-" + UDI、この操作のエラー 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

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。Status オブジェクト内の各 ID は、ポリシーの実行に関与するデバイス ID を表します。

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(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();
    }
}
```

createPolicy

構文

```
PolicyStatus createPolicy(UserToken token, String policy_name, SKUFilter sku_filter,
DeviceFilter dev_filter);
```

説明

この関数は、インベントリから SKU とデバイスを選択する規則のセットを表す Policy オブジェクトを作成します。

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 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。
policy_name	String、必須	x21 ~ x7A の範囲の ASCII 文字を含む最大 64 文字の文字列	Policy オブジェクトの名前。 許可されるポリシー名の最大長は、ポリシーを作成したユーザのユーザ名の長さによって異なります。ユーザ名とポリシー名の長さの合計が 63 文字を超えないようにする必要があります。
sku_filter	SKUFilter、 省略可能	—	入力 SKUFilter オブジェクトは、条件に一致するデバイスを検索するための条件を指定します。このフィルタを null に設定した場合、フィルタリングは行われません。
dev_filter	DeviceFilter、 省略可能	—	入力 DeviceFilter オブジェクトは、条件に一致するデバイスを検索するための条件を指定します。このフィルタを null に設定した場合、フィルタリングは行われません。

戻り値

この関数は PolicyStatus オブジェクトを返します。PolicyStatus には、Status と Policy オブジェクトが含まれます。PolicyStatus クラスと Status クラスの定義は次のとおりです。

```
public class Status implements Serializable {
    private int m_error_code;
    private String m_error_message;
}

public class PolicyStatusItem extends Status implements Serializable {
    private Policy m_policy;

    public Policy getPolicy() {
        return m_policy;
    }

    public void setPolicy(Policy policy) {
        this.m_policy = policy;
    }
}
```

エラーと例外

エラーが発生した場合は、PolicyStatus オブジェクトにエラー コードが含まれます。

deletePolicy

構文

```
Status deletePolicy(UserToken token, String policy_name);
```

説明

この関数は、指定した名前によって識別される Policy オブジェクトをシステムから削除します。

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 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。
policy_name	String、必須	—	Policy オブジェクトの名前。

戻り値

この関数は Status オブジェクトを返します。

エラーと例外

エラーが発生した場合は、エラーの情報を含む Status オブジェクトが返されます。

enumerateDeviceFilterAttribute

構文

```
String[] enumerateDeviceFilterAttribute(UserToken token, Policy.DeviceAttribute attrib);
```

説明

この関数は、DeviceFilter 内の指定した属性に対して有効なすべての値を含むリストを返します。

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。
attrib	Policy.DeviceAttribute、必須	—	属性名には次のいずれかを指定できます。 <ul style="list-style-type: none"> Policy.DeviceAttribute.MODEL Policy.DeviceAttribute.GROUP

戻り値

この関数は、指定した属性に対して有効なすべての値を表す文字列の配列を返します。

エラーと例外

エラーが発生した場合、この関数は `null` を返します。

enumerateSKUFilterAttribute

構文

```
String[] enumerateSKUFilterAttribute(UserToken token, Policy.SKUAttribute attrib);
```

説明

この関数は、SKUFilter 内の指定した属性に対して有効なすべての値を含むリストを返します。

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。
attrib	Policy.SKUAttribute、必須	—	属性名には <code>Policy.SKUAttribute.FEATURE_NAME</code> を指定できます。

戻り値

この関数は、指定した属性に対して有効なすべての値を表す文字列の配列を返します。

エラーと例外

エラーが発生した場合、この関数は `null` を返します。

listAllPolicies

構文

```
String[] listAllPolicies(UserToken token);
```

説明

この関数は、ユーザがアクセスできるすべてのポリシー（このユーザが所有しているプライベート ポリシーとすべてのパブリック ポリシーの両方を含む）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 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。

戻り値

この関数は、アクセス可能なポリシーの ID を表す文字列の配列を返します。

エラーと例外

エラーが発生した場合、この関数は `null` を返します。

listFilteredDevices

構文

```
String[] listFilteredDevices(UserToken token, DeviceFilter dev_filter);
```

説明

この関数は、指定したデバイス フィルタを実行することによって取得されるデバイスのリストを生成します。

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 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。
dev_filter	DeviceFilter 、必須	—	入力 DeviceFilter オブジェクトは、デバイスのリストを検索するための条件を指定します。

戻り値

この関数は、デバイスの ID を表す文字列の配列を返します。

エラーと例外

エラーが発生した場合、この関数は `null` を返します。

listDeviceIdsByFilter

構文

```
listDeviceIdsByFilter(UserToken token, String deviceType, String deviceModel, String[]
features) throws RemoteException;
```

説明

この関数は、デバイス タイプ、デバイス モデル、およびライセンス フィーチャを含むデバイス ID のリストを返します。フィルタリングを行わないフィルタには `null` を設定できます。すべてのフィルタを `null` に設定すると、エラーが発生して `null` が返されます。

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。
deviceType		—	デバイス タイプ。
deviceModel		—	デバイス モデル。
features		—	ライセンス フィーチャの名前を表す文字列の配列。

戻り値

この関数は、フィルタに一致するデバイス ID の配列を返します。

エラーと例外

エラーが発生した場合、この関数は `null` を返します。

listFilteredSKUs

構文

```
SKUIdentifier[] listFilteredSKUs(UserToken token, SKUFilter sku_filter);
```

説明

この関数は、ポリシー内の SKU フィルタを実行することによって取得される SKU 識別情報のリストを返します。この関数で使用する `SKUIdentifier` クラスは、次のように定義されます。

```
class SKUIdentifier {
    String pak;                // PAK of the selected SKU
    String sku_name;          // name the selected SKU
}
```

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。
sku_filter	SKUFilter、 必須	—	入力 SKUFilter オブジェクトは、デバイスのリストを検索するための条件を指定します。

戻り値

この関数は、フィルタに一致する SKUIDentifier オブジェクトの配列を返します。

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。

listApplicableDevicesBySKUs

構文

```
listApplicableDevicesBySKUs (UserToken token, SKU[] skus, String group);
```

説明

この関数は、指定したグループに含まれるデバイス ID のうち、指定した SKU 内のすべてのフィーチャと一致するライセンス可能なフィーチャを持つデバイス ID の配列を返します。グループが null の場合は、すべてのグループから一致するデバイス ID 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 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。
group	String、必須	—	グループの名前。
skus	—	—	照合に必要な SKU の配列。

戻り値

この関数は、指定したグループに含まれるデバイス ID のうち、指定した SKU 内のすべてのフィーチャと一致するライセンス可能なフィーチャを持つデバイス ID の配列を返します。入力 SKU 配列が null または空の場合、またはどの入力 SKU にもフィーチャが含まれていない場合は、すべてのデバイスが返されます。一致するデバイスが見つからない場合は、長さが 0 の配列が返されます。

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。

readPolicy

構文

```
PolicyStatus readPolicy(UserToken token, String policy_name);
```

説明

この関数は、指定した名前を使用して、データ ストレージから Policy オブジェクトを取得します。

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 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。
policy_name	String、必須	—	Policy オブジェクトの名前。

戻り値

この関数は PolicyStatus オブジェクトを返します。PolicyStatus には、Status と Policy オブジェクトが含まれます。

エラーと例外

エラーが発生した場合は、PolicyStatus オブジェクトにエラー コードが含まれます。

writePolicy

構文

```
Status writePolicy(UserToken token, Policy policy);
```

説明

この関数は、指定した Policy オブジェクトをデータ ストレージに書き込みます。入力 Policy オブジェクトには、createPolicy() 関数によって返された新しい Policy のインスタンス、または readPolicy() 関数によってデータ ストレージ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。
policy	Policy、必須	—	Policy オブジェクト。Policy クラスの新しいインスタンス、または readPolicy() 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。

戻り値

この関数は Status オブジェクトを返します。

エラーと例外

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。