



## アカウントの管理

---

この章は、次の項で構成されています。

- [アカウント \(1 ページ\)](#)
- [カスタム アカウント タイプの追加 \(1 ページ\)](#)

## アカウント

Open Automation を使用して、新しいカスタム アカウント タイプを Cisco UCS Director に追加することができます。

カスタム アカウント タイプは、アカウントを新しい方法で操作可能にする新しい Cisco UCS Director データインフラストラクチャを提供します。たとえば、新しいアカウントタイプを使用すると、新しい方法で Cisco UCS Director の管理対象要素を管理してそれに関するレポートを作成することができます。

## カスタム アカウント タイプの追加

アカウントタイプを追加するには、次の作業を実行します。

- `com.cloupia.lib.connector.account.AbstractInfraAccount` クラスを拡張します。
- `AccountType` と `AccountLabel` の値を定義します。
- `AccountTypeEntry` の必須パラメータを割り当てます。これには、`setPodTypes`、`setAccountClass`、`setAccountType`、および `setAccountLabel` が含まれます。
- 接続テストの実装を完了します。

次のコード スニペットに、新しいアカウントの詳細の指定方法を示します。

```
//This class is used to register a new connector into the UCSD.  
AccountTypeEntry entry=new AccountTypeEntry();  
  
//Set implementation class for Account type  
entry.setCredentialClass(FooAccount.class);
```

```
//Account Type
entry.setAccountType(FooConstants.INFRA_ACCOUNT_TYPE);

//Account Label which will be shown in UI
entry.setAccountLabel(FooConstants.INFRA_ACCOUNT_LABEL);

//Account Category like Compute, Storage, Network or Multi-Domain.
entry.setCategory(InfraAccountTypes.CAT_NETWORK);

//This is mandatory , report generation on context Level for the new account type
entry.setContextType(ReportContextRegistry.getInstance().
getContextByName(FooConstants.INFRA_ACCOUNT_TYPE).getType());

//Account class like Physical-1, Virtual-2, Network-5, Multi-domain-3
//or Other-4
entry.setAccountClass(AccountTypeEntry.PHYSICAL_ACCOUNT);

// This will be used along with the account name to show
// in the System Tasks Report.
entry.setInventoryTaskPrefix("Open Automation Inventory Task");

//Set time frequency to collect account Inventory
entry.setInventoryFrequencyInMins(15);

//Supported POD types for this connector.
entry.setPodTypes(new String[]{"FooStack"});

//To add this account type entry
PhysicalAccountTypeManager.getInstance().addNewAccountType(entry);

// This is mandatory, to test the connectivity of the new account. The
// Handler should be of type PhysicalConnectivityTestHandler.
entry.setTestConnectionHandler(new FooTestConnectionHandler());
// This is mandatory, we can implement inventory listener according to
// the account Type , collect the inventory details.
entry.setInventoryListener(new FooInventoryListener());

//This is mandatory , to show in the converged stack view
entry.setConvergedStackComponentBuilder(new DummyConvergedStackBuilder());

//This is required to show up the details of the stack view in the GUI
entry.setStackViewItemProvider(new DummyStackViewItemProvider());

// This is required credential.If the Credential Policy support is
// required for this Account type then this is mandatory, can implement
// credential check against the policyname.
entry.setCredentialParser(new FooAccountCredentialParser());
```



---

**重要** SDK サンプルを参照してください。実装については、`com.cloupia.feature.foo.accounts.FooAccount` と `com.cloupia.feature.foo.FooModule` を参照してください。

---