

それでは始めましょう！  
ブラウザから次のURLにアクセスしてください。

Playground  
Online

<https://hyperledger.github.io/composer/>

Playground  
Tutorial

<https://hyperledger.github.io/composer/tutorials/playground-guide.html>

Hyperledger Composer

https://composer-playground.mybluemix.net/editor

Search

Web empty-business-network

DefineTest

admin

FILES

About  
README.md, package.json

Model File  
models/model.cto

Script File  
lib/logic.js

Access Control  
permissions.acl

Add a file...Export

UPDATE NETWORK

From: 0.0.2-deploy.2

To: 0.0.2-deploy.3

Deploy changes

About File README.md

This is the readme file for the Business Network Definition created in Playground

LegalGitHub

Playground v0.19.12TutorialDocsCommunity

# Model File全体

```
/**
 * My commodity trading network
 */
namespace org.example.mynetwork
asset Commodity identified by tradingSymbol {
    o String tradingSymbol
    o String description
    o String mainExchange
    o Double quantity
    --> Trader owner
}
participant Trader identified by tradeId {
    o String tradeId
    o String firstName
    o String lastName
}
transaction Trade {
    --> Commodity commodity
    --> Trader newOwner
}
```

# Assetの定義

```
asset Commodity identified by tradingSymbol {  
    o String tradingSymbol  
    o String description  
    o String mainExchange  
    o Double quantity  
    --> Trader owner  
}
```

- Assetクラス名 : Commodity (商品)
- tradingSymbol の値で識別する
- tradingSymbol、詳細、主要取引所、数量、所有者の値を定義する
  - 所有者の値は、Traderクラスを参照



# Participantの定義

```
participant Trader identified by tradeId {  
    o String tradeId  
    o String firstName  
    o String lastName  
}
```

- Participantクラス名 : Trader (商人)
- tradeId の値で識別する
- tradeId、商人の姓、名のプロパティを定義する

# Transactionの定義

```
transaction Trade {  
    --> Commodity commodity  
    --> Trader newOwner  
}
```

- Transactionクラス名 : Trade (取引)
- 商品、新所有者のプロパティを定義する
  - 商品の値は、Commodityクラスを参照
  - 新所有者の値は、Traderのクラスを参照

# Script File : Transaction Processor Functionの定義

```
/**
 * Track the trade of a commodity from one trader to
 another
 * @param {org.example.mynetwork.Trade} trade - the
 trade to be processed
 * @transaction
 */
async function tradeCommodity(trade) {
  trade.commodity.owner = trade.newOwner;
  let assetRegistry = await
getAssetRegistry('org.example.mynetwork.Commodity');
  await assetRegistry.update(trade.commodity);
}
```

トランザクション・インスタンスが引数に渡される

assetRegistryを介してAsset “commodity”の  
更新トランザクションを実行する

# Access Control全体 : Access Control ruleの定義

```
/**
 * Access control rules for tutorial-network
 */
rule Default {
  description: "Allow all participants access to all
resources"
  participant: "ANY"
  operation: ALL
  resource: "org.example.mynetwork.*"
  action: ALLOW
}

rule SystemACL {
  description: "System ACL to permit all access"
  participant: "ANY"
  operation: ALL
  resource: "org.hyperledger.composer.system.**"
  action: ALLOW
}
```

全てのParticipantに全てのビジネスネットワークへのアクセス許可

全てのParticipantに全てのシステム・リソースへのアクセス権限

Hyperledger Composer

https://composer-playground.mybluemix.net/test

Search

Web empty-business-network

DefineTest

admin

PARTICIPANTS

Trader

ASSETS

Commodity

TRANSACTIONS

All Transactions

Submit Transaction

Participant registry for org.example.mynetwork.Trader

+ Create New Participant

| ID                                                                                  | Data |
|-------------------------------------------------------------------------------------|------|
|  |      |

This registry is empty!

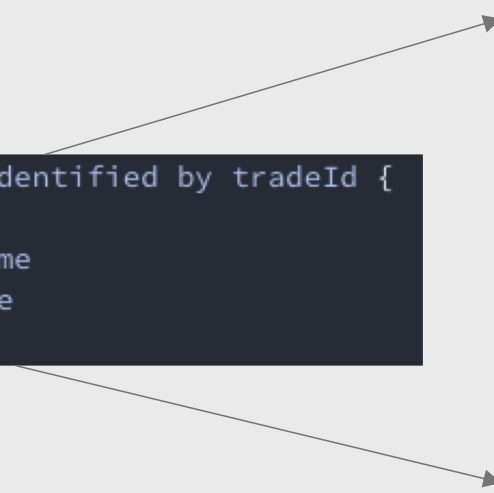
To create resources in this registry click create new at the top of this page

LegalGitHub

Playground v0.19.12TutorialDocsCommunity

# Participant - Traderクラスのインスタンス作成

```
participant Trader identified by tradeId {  
  o String tradeId  
  o String firstName  
  o String lastName  
}
```



The diagram shows two arrows originating from the participant definition box. One arrow points to the first JSON instance (Trader 1) and the other points to the second JSON instance (Trader 2).

```
1  {  
2    "$class": "org.example.mynetwork.Trader",  
3    "tradeId": "TRADER1",  
4    "firstName": "Donald",  
5    "lastName": "Trump"  
6  }
```

```
1  {  
2    "$class": "org.example.mynetwork.Trader",  
3    "tradeId": "TRADER2",  
4    "firstName": "Vladimir",  
5    "lastName": "Putin"  
6  }
```

# Asset – Commodity インスタンスの作成

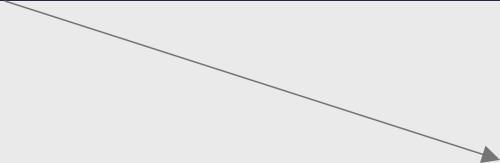
```
asset Commodity identified by tradingSymbol {  
  o String tradingSymbol  
  o String description  
  o String mainExchange  
  o Double quantity  
  --> Trader owner  
}
```



```
1  {  
2    "$class": "org.example.mynetwork.Commodity",  
3    "tradingSymbol": "ABC",  
4    "description": "Test commodity",  
5    "mainExchange": "Euronext",  
6    "quantity": 72.297,  
7    "owner": "resource:org.example.mynetwork.Trader#TRADER1"  
8  }
```

# Transactionの実施

```
transaction Trade {  
  --> Commodity commodity  
  --> Trader newOwner  
}
```



```
1  {  
2    "$class": "org.example.mynetwork.Trade",  
3    "commodity": "resource:org.example.mynetwork.Commodity#ABC",  
4    "newOwner": "resource:org.example.mynetwork.Trader#TRADER2"  
5  }
```



# Asset - Commodityの情報確認

## PARTICIPANTS

Trader

## ASSETS

Commodity

## TRANSACTIONS

All Transactions

Asset registry for org.example.mynetwork.Commodity

ID

Data

ABC

```
{  
  "$class": "org.example.mynetwork.Commodity",  
  "tradingSymbol": "ABC",  
  "description": "Test commodity",  
  "mainExchange": "Euronext",  
  "quantity": 72.297,  
  "owner": "resource:org.example.mynetwork.Trader#TRADER2"  
}
```

Collapse

# Transactionの履歴確認

## Historian Record

Transaction      Events (0)

```
1  |{
2    "$class": "org.example.mynetwork.Trade",
3    "commodity": "resource:org.example.mynetwork.Commodity#ABC",
4    "newOwner": "resource:org.example.mynetwork.Trader#TRADER2",
5    "transactionId": "3acfa124-1d6a-4688-802b-a4d003d4077e",
6    "timestamp": "2018-07-20T04:15:40.254Z"
7  }
```

TransactionのIDとタイムスタンプが追記



ワークショップ、セッション、および資料は、IBMまたはセッション発表者によって準備され、それぞれ独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる参加者に対しても法律的またはその他の指導や助言を意図したものではなく、またそのような結果を生むものでもありません。本講演資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本講演資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本講演資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引きだすことを意図したものでも、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでもなく、またそのような結果を生むものでもありません。

本講演資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本講演資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本講演資料に含まれている内容は、参加者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したものでも、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、[以下当該情報に関連し商標リスト中に掲載されたIBMブランドやIBMの製品名称があれば追加する]は、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。現時点でのIBMの商標リストについては、[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) をご覧ください。

Adobe、Adobeロゴ、PostScript、PostScriptロゴは、Adobe Systems Incorporated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。

IT Infrastructure LibraryはAXELOS Limitedの登録商標です。

インテル、Intel、Intelロゴ、Intel Inside、Intel Insideロゴ、Centrino、Intel Centrinoロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、およびPentium は Intel Corporationまたは子会社の米国およびその他の国における商標または登録商標です。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。  
PowerLinux is a trademark of International Business Machines Corp. The registered trademark Linux is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Microsoft、Windows、Windows NT および Windowsロゴは Microsoft Corporationの米国およびその他の国における商標です。

ITILはAXELOS Limitedの登録商標です。

UNIXはThe Open Groupの米国およびその他の国における登録商標です。

Cell Broadband Engineは、Sony Computer Entertainment, Inc.の米国およびその他の国における商標であり、同社の許諾を受けて使用しています。

JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の米国およびその他の国における商標または登録商標です。

Linear Tape-Open、LTO、LTOロゴ、UltriumおよびUltriumロゴは、HP、IBM Corp.およびQuantumの米国およびその他の国における商標です。

VMware、the VMware logo、VMware Cloud Foundation、VMware Cloud Foundation Service、VMware vCenter ServerおよびVMware vSphereは、VMware、Inc.またはその子会社の米国およびその他の地域における登録商標または商標です。