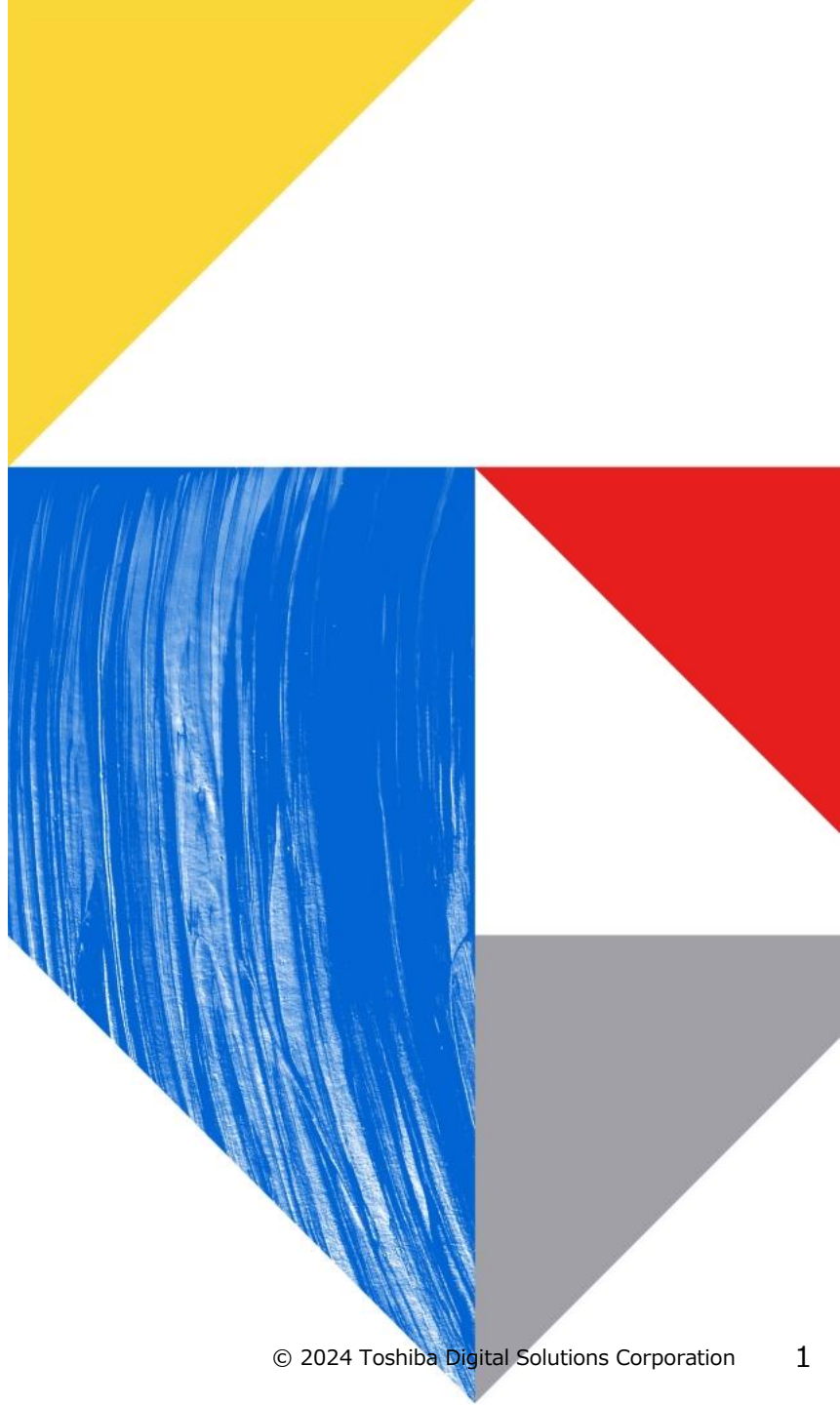
サンプルコードについて



サンプルコードの種類

サンプルコードは、DNCWARE Blockchain+™のアプリケーションを作成する際に参考になるプログラム(コード)を例示するものです。サンプルコードの種類として、以下の３種類があります。

basicサンプル（基礎編）

基礎的な使い方を例示して説明するものです。
ウォレット作成、コントラクトのデプロイ、コントラクトAPI、グループ管理、アクセス権限、
キーバリューコントラクト、UnifiedName、サブコントラクト、など

advancedサンプル（応用編）

より高度な使い方を例示して説明するものです。
TX要求のオフライン作成、マルチシグ、コントラクトのライブラリ化、プロキシ環境対応、など

practicalサンプル（実用編）

実用的なアプリケーションを想定したサンプルコードです。

サンプルコードの見かた

HTML形式のファイルで提供され、ブラウザで開いて見ることができます。

右のようなノートブックとなっており、説明つきでサンプルコードとその実行結果が示されています。

実行は、上から下へと進みます。

コードセルの中にコードそのものが示されています。コードセルの前にその説明、後ろにその実行結果が示されています。

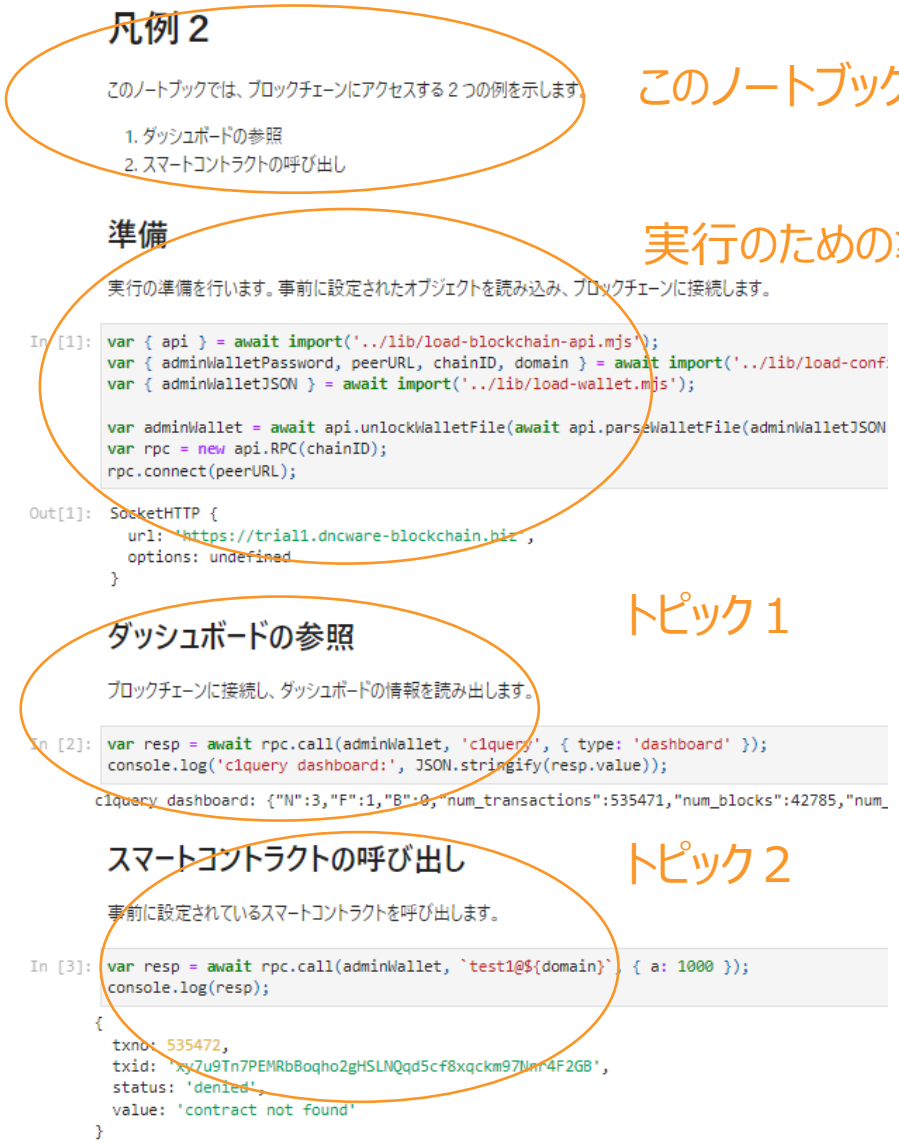
サンプルコードの構成

構成はおおよそ下記の通りです。

・冒頭で概要を説明します。
・実行のための準備をします。
・一番目のトピックを説明します。
・二番目のトピックを説明します。
あとはこの繰り返しです。

準備の部分は、実行のために必要なセットアップです。概略を把握したいだけのときは、ここは読み飛ばして構いません。

その下に続くトピックがもともと説明したい内容となります。複数のトピックがある場合には、見出しを大きくして、トピックの境界が分かるようにしています。



サンプルコードの中で利用しているユーティリティ関数について

サンプルコードが煩雑になるのを避けるため、本題に関係のない処理はユーティリティ関数を利用して記述する場合があります。ここでは、代表的なユーティリティ関数を説明します。
(なお、ユーティリティ関数はサンプルコード専用であり、一般に利用することを想定したものではありません)

```
var { api } = await import('../lib/load-blockchain-api.mjs');
```
クライアントAPIのライブラリを読み込みます。

```
var { adminWalletPassword, peerURL, chainID, domain } = await import('../lib/load-config.mjs');
```
設定ファイル(samples/etc/config.json)から設定値を読み込みます。

```
var { loadWallet, adminWalletJSON } = await import('../lib/load-wallet.mjs');
```
設定フォルダ(samples/etc)に置かれているウォレットのJSONを読み込みます。

```
var { adminWallet, rpc, deploySmartContract } = await import('../lib/notebook-util.mjs');
```
基礎編1の準備と同様の処理を行います。

```
var cid = deploySmartContract([argtypes,] func, options);
```
スマートコントラクトをブロックチェーンにデプロイします。

TOSHIBA