

# JUnit + Mockito in Spring Boot

# Introduction

JUnitとmockitoを使用することにより、モックを使用した単体テストが可能になる。

モックとは、テストの際のみに使用される擬似的なオブジェクトである。



# 目次

1, メリット

2, JUnit

3, Mockito

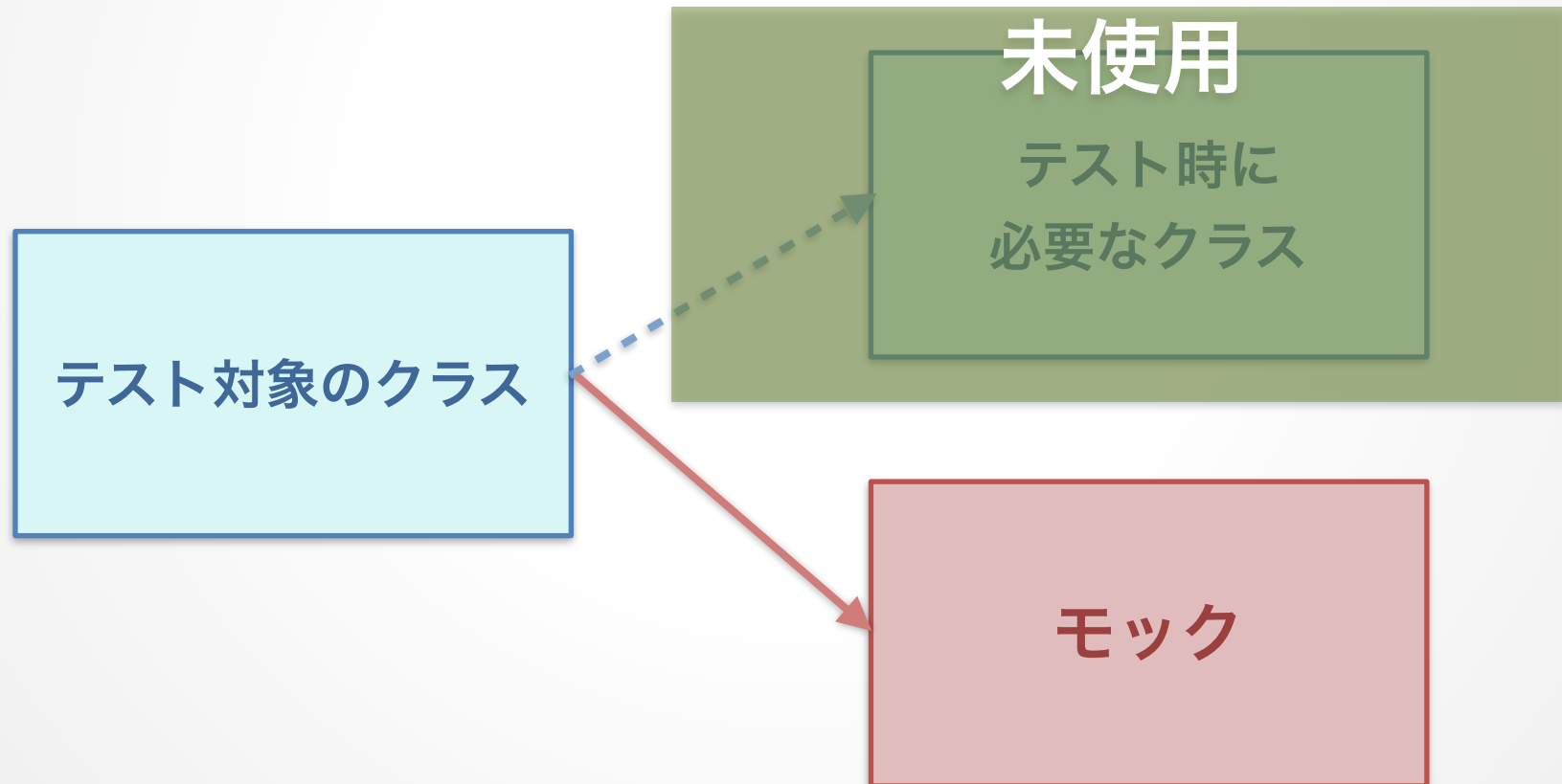
4, サンプル

4-1, プロジェクト作成

4-2, テストの実行

# 1, メリット

テストの際に、テスト対象が別クラスの処理に依存していた場合でも、その部分をモック化することにより、テスト対象のみをテストすることが出来る。



## 2, JUnit

テストコードから、テスト対象のメソッドを呼び出して挙動をテスト出来る。

テストコードは、他のテストに対しても流用可能なため、テスト時の工数を削減することが出来る。

The logo for JUnit, featuring the word "JUnit" in a serif font. The "J" is green, the "U" is red, and "nit" is red.

# 3, Mockito

モックを作成するためのモックライブラリである。

テスト対象の依存先を、モックオブジェクトに置き換えることにより、効率的なテストを実施することが出来る。



# 3, Mockito

## **@Mock**

モック化するインスタンスに付けるアノテーション。

## **@InjectMocks**

テスト対象のクラスに、モック化されたインスタンスを、  
注入するためのアノテーション。



## 4, サンプル

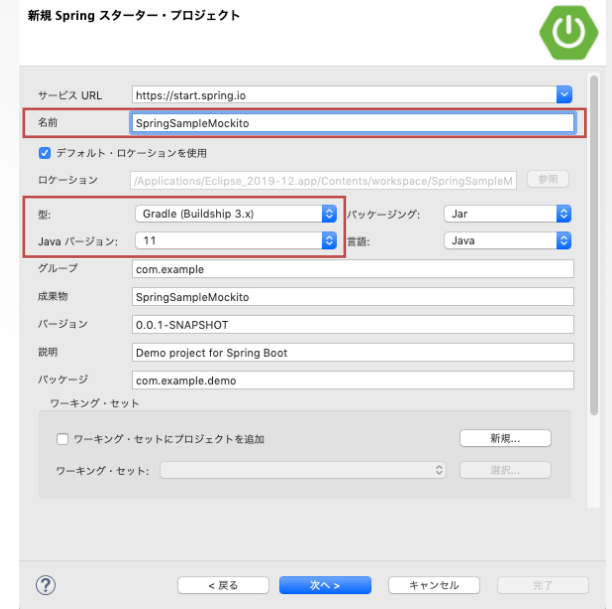
## 4-1, プロジェクト作成

## 1, Spring スターター・プロジェクト作成

名前：SpringSampleMockito

型 : Gradle (Buildship 3.x)

# Java バージョン : 11

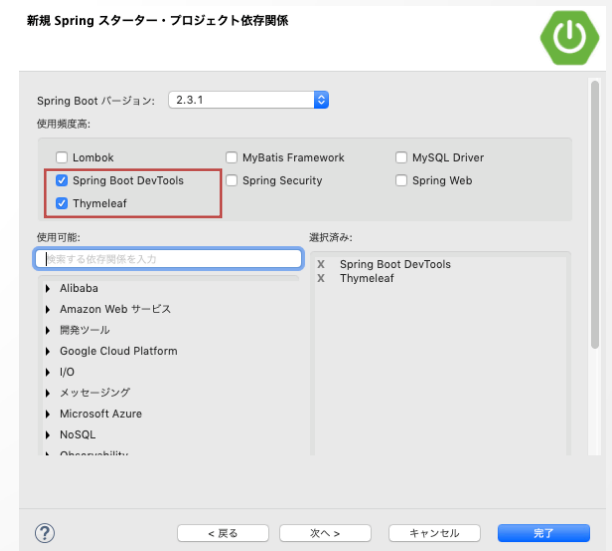


## 2, 新規 Spring スターター・プロジェクト依存関係

下記の項目をチェック。

- Spring Web
- Spring Boot DevTools

※画面上部に表示されていない場合は、検索欄を使用。





# 4, サンプル

## 4-1, プロジェクト作成

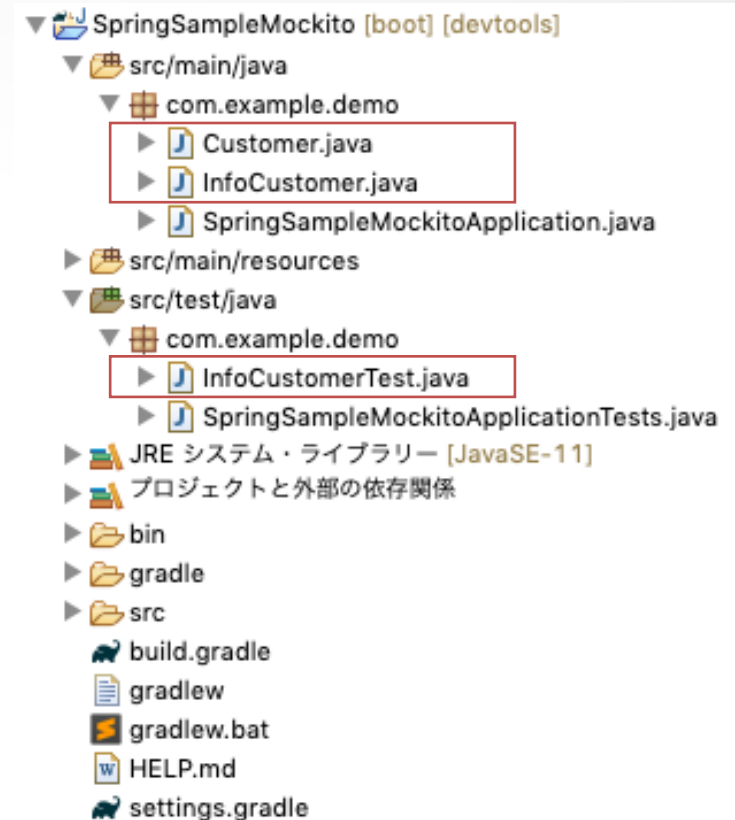
### 3, 追加ファイル

Javaクラスファイル

Customer.java

InfoCustomer.java

InfoCustomerTest.java



※ファイル内容は、サンプルファイルよりコピー&ペースト。

# 4, サンプル

## 4-1, プロジェクト作成

## 4, JUnitライブラリの追加

SpringSampleMockitoを右クリック

プロパティーを選択

Javaのビルド・パスを選択

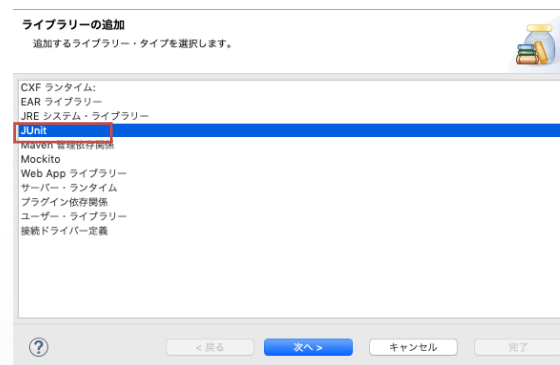
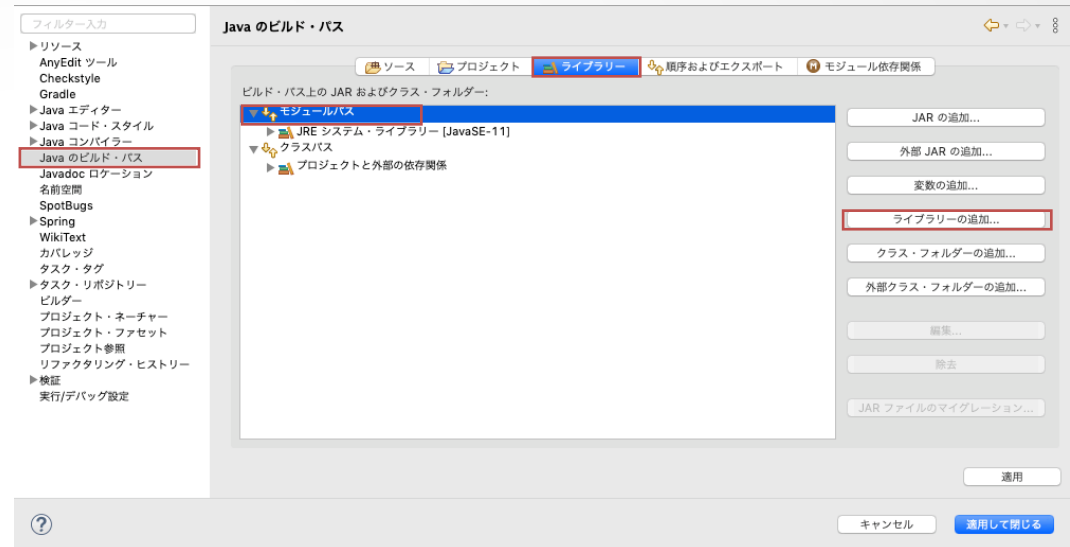
ライブラリー

→モジュールパス

→ライブラリーの追加

JUnitを選択

JUnit4を選択



# 4, サンプル

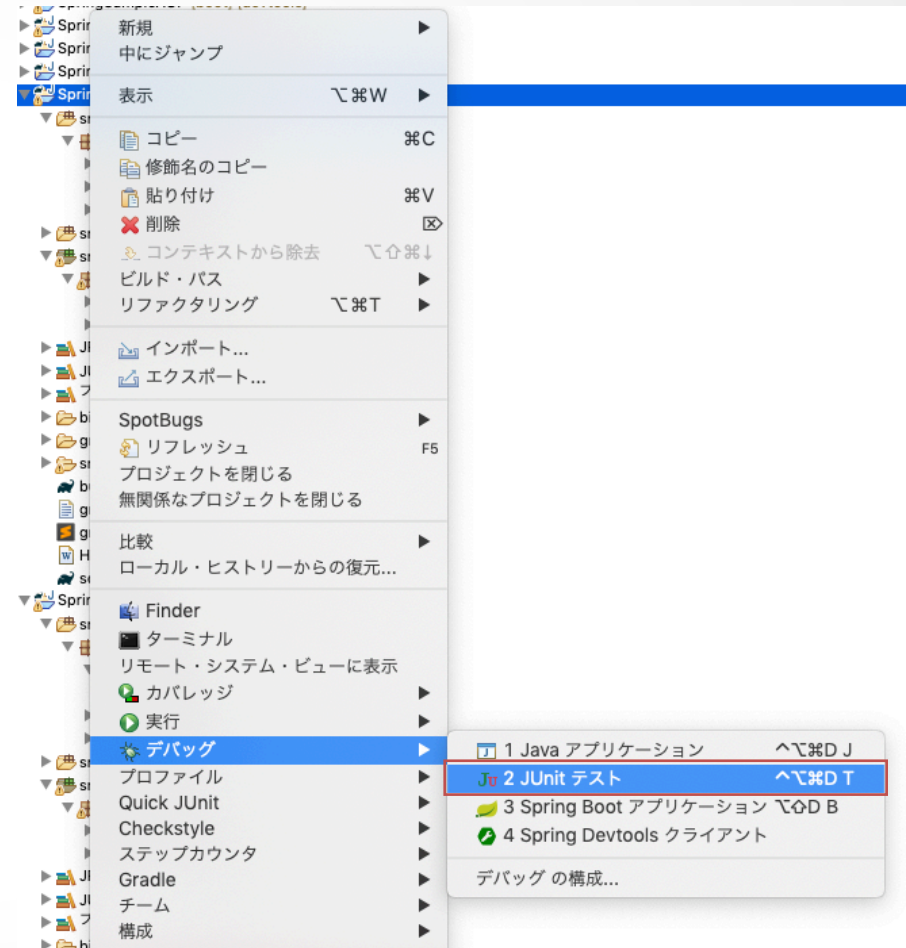
## 4-2, テストの実行

### 1, JUnitテスト

SpringSampleMockitoを右クリック

デバッグ

→JUnitテスト をクリック



# 4, サンプル

## 4-2, テストの実行

### 2, 実行結果の確認

JUnitのウィンドウが自動的に開く。

緑色のバーが表示される。

→テストが正常実行された。



# JUnit + Mockito in Spring Boot

Fin.