

# S2JUnit4 を使ったテスト

[http://yakunikunotare.boo.jp/orebase2/sastruts/service\\_test\\_by\\_s2junit4](http://yakunikunotare.boo.jp/orebase2/sastruts/service_test_by_s2junit4)

[http://daipresents.com/2009/s2junit4\\_service\\_unit\\_testing/](http://daipresents.com/2009/s2junit4_service_unit_testing/)

## 準備

<http://eclipse.seasar.org/updates/3.3/>

から

S2JUnit4

をインストール

## テストクラスの作成

1. テストクラスを作成するファイルを右クリック
2. S2JUnit4 -> テスティングペアを開く
3. ペアがない場合は、「作成しますか?」と聞かれるので、「はい」
4. テストの内容を書く

## サンプル

### テストの実行

ファイルを選択して

ctrl+0

メニューから選んでもいいけど、バグがあるらしく「テストケースを選択してください」と表示されて進まないことがある。

普通にJUnit テストを選んでもいけると思う。

## テストデータについて

### 命名規則

s2junit4.dicon

にテストデータの命名規則が書いてある。基本的には

初期データは、テストクラス名\_テストメソッド名.xls  
実行後(期待される)データは、テストクラス名\_テストメソッド名\_Expected.xls

また、初期データは

テストクラス名.xls

としてまとめても良い。

```
accessor.readXlsAllReplaceDb("DeptServiceTest_testFindById.xls");
```

で、テストの前にメソッド名やクラス名と関係のないファイルを読み込むこともできる。

### データの内容のルール

以下の様な決まりがあるので注意。

- ・ sheet 名に対象のテーブル
- ・ 1 行目にカラム名
- ・ 2 行目移行にデータ

### データファイルのパス

基本的にはテストクラスと同じ場所に xls ファイルを置く。

環境によっては、テストクラスと同じ場所に置いても読み込んでくれないことがある。

1 階層戻って、target ディレクトリ以下に置けば読みこんでくれるはず。

例えば、テスト対象とテストクラスが

```
struts_sample02%src%main%java%sample02%service%DeptService.java
struts_sample02%src%test%java%sample02%service%DeptServiceTest.java
```

の場合

```
struts_sample02%target%test-classes%sample02%service
```

にテストのデータ (xls) ファイルを置いてしまう。

テストを実際に行なっているパスを知る

```
System.out.println(getClass().getResource("").getPath());
```

をテストクラスに書くと実際にテストを行なっているパスがわかる。

テストの前にテーブルの中をテストデータに置き換える

こうすることで、すべて置き換わる。

テスト後はロールバックするので元に戻る。