

## Популярные сценарии работы с Mockito

JSP & Servlets  
4 уровень, 5 лекция

ОТКРЫТА

### 6.1 Мокирование статического метода mockStatic()

И еще один важный момент – мокирование и верификация статических методов. “А что в этом такого?”, – спросишь ты. Да, статические, но ведь методы же. И будете неправы.

Помни, с чего мы начали изучение мок-объектов? С того, что эти объекты искусственно создаются через класс `DynamicProxy`. А статические методы ни к каким объектам не привязаны и перехватить вызовы к ним через `DynamicProxy` нельзя. Вот и все.

Но создатели Mockito и тут смогли извернуться – написали свой загрузчик классов и с его помощью смогли подменять классы на лету. Большая и сложная работа, но они все-таки смогли это сделать.

Понадобится добавить дополнительную библиотеку в `pom.xml`:

```
<dependency>
  <groupId>org.mockito</groupId>
  <artifactId>mockito-inline</artifactId>
  <version>4.2.0</version>
  <scope>test</scope>
</dependency>
```

Вот как нужно работать, если вам нужно мокировать статический метод.

#### 1 Создаем специальный мок-объект класса:

```
MockedStatic<ИмяКласса>управляющийОбъект = Mockito.mockStatic(ИмяКласса.class);
```

#### 2 Добавляем к этому объекту правила работы:

К этому объекту правила нужно цеплять другими способами.

```
управляющийОбъект.when(ИмяКласса::имяМетода).thenReturn(результат);
```

#### 3 Обязательно заворачиваем использование этого объекта в `try-with-resources`, чтобы объект сразу удалился и Mockito могло очистить связанные с ним правила.

Пример:

```
1  @Test
2  void givenStaticMethodWithNoArgs () {
3      try (MockedStatic< StaticUtils> utilities = Mockito.mockStatic( StaticUtils.class)) {
4          //добавляем правило
5          utilities.when(StaticUtils::name).thenReturn("Привет");
6      }
```

```
7 //проверяем, что правило работает
8 assertEquals("Привет", StaticUtils.name());
9     }
10 }
```

Не так красиво, как с аннотациями `@Mock` и `@Spy`, зато очень практично. Было очень тяжело писать тесты, когда нельзя было замочать простой статический метод, который использовался внутри тестируемых методов.

-

+21

+

Комментарии (2)

популярные

новые

старые

JavaCoder

Введите текст комментария



Stas S

Уровень 76, Гродно, Беларусь

8 сентября, 13:49

...

Может кому-нибудь поможет



Ответить

-

+4

+

Никита

Уровень 77, Киев, Украина

27 июля, 20:34

...

Я потратил уже 4 часа в попытках разобраться с тестированием статического метода в проекте в конце. Ставлю 1 звезду, т.к на основе этой лекции разобраться невозможно.

Ответить

-

+13

+

[Помощь по задачам](#)

[Форум](#)

[Отзывы](#)

[Подписки](#)

[Чат](#)

[FAQ](#)

[Задачи-игры](#)

[Истории успеха](#)

[Поддержка](#)

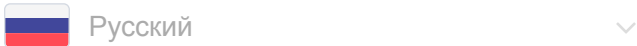
[Активности](#)



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА



СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ



"Программистами не рождаются" © 2022 JavaRush