

Capturing arguments with ArgumentCaptor



Understanding ArgumentCaptor

Creating an ArgumentCaptor

```
ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
```

@Captor

```
ArgumentCaptor<String> captor;
```

Capture Argument

// Assuming mockedObject is a mock

```
ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);  
mockedObject.sendMessage("Expected Message");
```

```
verify(mockedObject).sendMessage(captor.capture());  
String capturedArgument = captor.getValue();
```

// Now you can assert something about the captured argument

```
assertEquals("Expected Message", capturedArgument);
```

Capturing Multiple Calls

```
@Captor
ArgumentCaptor<String> captor;

// Inside the test
mockedObject.sendMultipleMessages("Hello", "World", "!");
verify(mockedObject, times(3)).sendMessage(captor.capture());

List<String> allCaptured = captor.getAllValues();

assertEquals(Arrays.asList("Hello", "World", "!"), allCaptured);
```

Generics and ArgumentCaptor

@Captor

```
ArgumentCaptor<List<String>> listCaptor;
```

```
// Imagine a method that adds to a list and then processes it  
mockedObject.processList(Arrays.asList("One", "Two", "Three"));  
verify(mockedObject).processList(listCaptor.capture());
```

```
List<String> capturedList = listCaptor.getValue();  
assertThat(capturedList, hasItems("One", "Two", "Three"));
```

Real-world Use Cases for ArgumentCaptor

Verifying Email Content

@Captor

```
ArgumentCaptor<User> userCaptor;
```

```
verify(mockEmailSender).sendWelcomeEmail(userCaptor.capture());
```

```
User registeredUser = userCaptor.getValue();
```

```
assertEquals("John Doe", registeredUser.getName());
```

```
assertEquals("[email protected]", registeredUser.getEmail());
```


Capturing Purchase Details

@Captor

```
ArgumentCaptor<Collection<Item>> itemsCaptor;
```

```
verify(mockInventoryService).updateInventory(itemsCaptor.capture());
```

```
Collection<Item> purchasedItems = itemsCaptor.getValue();
```

```
assertTrue(purchasedItems.stream().allMatch(item -> item.getStockLevel() >= 0));
```

Capturing Events

@Captor

```
ArgumentCaptor<ArticlePostedEvent> eventCaptor;
```

```
verify(mockEventPublisher).publish(eventCaptor.capture());  
ArticlePostedEvent event = eventCaptor.getValue();
```

```
assertEquals("Unit Testing Mastery", event.getArticleTitle());
```