

REFACTORING WITH INTELLIJE

Extract Method Objekt

- Markiere Methoden Body
- Extract => Method Object
 - Create inner class // aktiviert
 - Class name: SetupTeardownSurrounder
 - Alles anderes beibehalten
- OK
- Run MakeHTMLTest

Extract Instance Variable from a Local Variable

- In invoke(): lokale Variable "wikiPage" markieren
- Markiere "wikiPage" => Rechte Maustaste => Refactor => Extract => Field..
- Choose Class to Introduce Field = Inner Class SetupTeardownSurrounder
- Namensvorschlag "wikiPage" annehmen, anklicken
- Run MakeHTMLTest
- In invoke(): lokale Variable "buffer" markieren.
- Markiere "buffer" => Rechte Maustaste => Refactor => Extract => Field..
- Choose Class to Introduce Field = Inner Class SetupTeardownSurrounder
- Namensvorschlag "buffer" annehmen, anklicken
- Run MakeHTMLTest

Initialize the extracted Instance Variables (from local) in the Constructor

- In invoke(): lokale Variable "wikiPage" markieren.
- ALT, SHIFT, Up (Pfeil nach oben)
- wikiPage = pageData.getWikiPage() bis zum Constructor schieben
- Markiere "buffer" => ALT + Shift + Up bis zum Constructor schieben
- Run MakeHTMLTest

Eliminate Duplicates

```
WikiPagePath pagePath = wikiPage.getPageCrawler().getFullPath(suiteSetup);
String pagePathName = PathParser.render(pagePath);
buffer.append("!include -setup .").append(pagePathName).append("\n");
```

```
WikiPagePath setupPath = wikiPage.getPageCrawler().getFullPath(setup);
String setupPathName = PathParser.render(setupPath);
buffer.append("!include -setup .").append(setupPathName).append("\n");
```

```
WikiPagePath tearDownPath = wikiPage.getPageCrawler().getFullPath(tearDown);
String tearDownPathName = PathParser.render(tearDownPath);
buffer.append("!include -tearDown .").append(tearDownPathName).append("\n");
```

```
WikiPagePath pagePath = wikiPage.getPageCrawler().getFullPath(suiteTearDown);
String pagePathName = PathParser.render(pagePath);
buffer.append("!include -tearDown .").append(pagePathName).append("\n");
```

Extract a Field from a Method Call

- Aus `wikiPage.getPageCrawler()` extrahieren wir `private Crawler crawler`
- Markiere `wikiPage.getPageCrawler()`. => Refactor => Extract => Field
- Choose Class to Introduce Field = Inner Class `SetupTearDownSurrouder`
- Namensvorschlag: "crawler" übernehmen
- Replace All 4 Occurrence aktivieren, Doppelt klick auf "crawler"
- Run MakeHTMLTest
- Move `crawler = wikiPage.getPageCrawler();` from invoke to the Constructor
- ALT + Shift + Up
- Run MakeHTMLTest

PARAMETERIZE A CONSTANT

```
buffer.append("!include -setup .").append(pagePathName).append("\n");
buffer.append("!include -setup .").append(setupPathName).append("\n");
buffer.append("!include -tearDown .").append(tearDownPathName).append("\n");
buffer.append("!include -tearDown .").append(pagePathName).append("\n");
```

- Im ersten Segment "setup" markieren => Refactor => Extrakt => Variable
- Name = mode => OK
- Run MakeHTMLTest
- Wiederholen für die zweite "setup", Erste "tearDown" und zweite "tearDown"
- Name = mode => OK
- Run MakeHTMLTest
- Die Anweisung `String mode` nach oben Verschieben vor `WikiPagePath ...`
- ALT + Shift + Up

Extract a Method from a Duplicate Code

```
String mode = "setup";
WikiPagePath pagePath = crawler.getFullPath(suiteSetup);
String pagePathName = PathParser.render(pagePath);
buffer.append("!include -" + mode + " . ").append(pagePathName).append("\n");
```

```
String mode = "setup";
WikiPagePath setupPath = crawler.getFullPath(setup);
String setupPathName = PathParser.render(setupPath);
buffer.append("!include -" + mode + " . ").append(setupPathName).append("\n");
```

```
String mode = "teardown";
WikiPagePath teardownPath = crawler.getFullPath(teardown);
String teardownPathName = PathParser.render(teardownPath);
buffer.append("!include -" + mode + " . ").append(teardownPathName).append("\n");
```

```
String mode = "teardown";
WikiPagePath pagePath = crawler.getPageCrawler().getFullPath(suiteTeardown);
String pagePathName = PathParser.render(pagePath);
buffer.append("!include -" + mode + " . ").append(pagePathName).append("\n");
```

- Markiere die ersten 3 Zeilen von dieser repeating groups

```
WikiPagePath pagePath = crawler.getFullPath(suiteSetup);
String pagePathName = PathParser.render(pagePath);
buffer.append("!include -" + mode + " . ").append(pagePathName).append("\n");
```

- Rechte Maustaste => Refactor => Extrakt Method
- Name = includePage
- Parameters = String mode
- Parameters = WikiPage suitePage. => zu page ändern => Refactor
- Process Duplicate 1 of 3 > Do you want to replace this occurrence?
- Replace bestätigen : 3 mal bestätigen
- Run MakeHTMLTest

```
String mode = "setup";
includePage(mode, suiteSetup);
```

```
String mode = "setup";
includePage(mode, setup);
```

```
String mode = "teardown";
includePage(mode, teardown);
```

```
String mode = "teardown";
includePage(mode, suiteTeardown);
```

CLEAN UP

```
String mode = "setup";
includePage(mode, suiteSetup);

String mode = "setup";
includePage(mode, setup);

String mode = "teardown";
includePage(mode, teardown);

String mode = "teardown";
includePage(mode, suiteTeardown);
```

Inline Pattern

- In includePage Aufrufe : markiere den Parameter "mode"
- Rechte Maustaste => Refactor => Inline ...
- Inline Alle References and remove the variable (1 occurrence)
- Refactor => Bestätigen
- Parameters = String mode
- Parameters = WikiPage suitePage. => zu page ändern => Refactor
- Process Duplicate 1 of 3 > Do you want to replace this occurrence?
- Replace bestätigen : 3 mal bestätigen
- Run MakeHTMLTest

CLEAN UP

```
if(suiteSetup != null)
{
    includePage("setup", suiteSetup);
}

if(setup != null){
    includePage("setup", setup);
}

if(teardown != null){
    includePage("teardown", teardown);
}

if(suiteTeardown != null)
{
    includePage("teardown", suiteTeardown);
}
```

- Geschweifte Klammer {} entfernen!
- Run MakeHTMLTest

CLEAN UP : String.format Methode benutzen

- In private includePage: Ersetze

```
buffer.append("!include -" + mode + " . ").append(pagePathName).append("\n");
```

- durch: String.format

```
buffer.append(String.format("!include -%s . %s\n", mode, pagePathName));
```

- Run MakeHTMLTest