

1 Übungen zu TDD mit PHP und TYPO3 CMS 7.6/8.7

1.1 TDD mit Composer (ohne TYPO3)

1. Klont euch das TDD-Seed-Projekt¹.
2. Führt ein `composer install` aus.
3. Richtet PHPUnit in PhpStorm ein und lasst die Tests des Projektes in PhpStorm laufen.
4. Lasst die Tests über die Kommandozeile laufen.
5. Wechselt vom `master`-Branch auf den Branch `feature/coffee`. Führt alle Tests aus.
6. Schreibt in `Test/Unit/MathTest.php` einen Test, der überprüft, dass $1 + 1 = 2$ ist.² Führt die Tests aus.
7. Nehmt euch ein Blatt Papier und schreibt *Testliste* darauf. Das ist eure Testliste, auf die ihr notiert, welche Tests ihr schreiben möchtet.
8. Legt ein `SizeOption`-Model an. Testet, dass es instanziiert werden kann.
9. Fügt (testgetrieben) ein Feld Integer-Feld `SizeOption.milliliters` hinzu. Werft eine `UnexpectedValueException`, falls beim Setter 0 oder ein negativer Wert übergeben wird.
10. Fügt testgetrieben ein Boolean-Feld `SizeOption.isIncludedInPrice` mit Getter und Setter hinzu.
11. Legt (testgetrieben) eine 1:n-Assoziation `CoffeeBeverage.sizeOptions` an (inklusive aller nötigen Methoden).
12. Schreibt (testgetrieben) eine Methode `CoffeeBeverage.getPriceTag`, die als einen String die verfügbaren Größen zurückgibt inklusive der Information, ob die Größe im Preis enthalten ist oder extra bezahlt werden muss.

¹<https://github.com/oliverklee/tdd-seed>, alternativ per Composer als `oliverklee/tdd-seed`

²Es ist gut, das ab und an zu überprüfen, weil wir ein echtes Problem bekommen, falls das irgendwann nicht mehr stimmen sollte.

1.2 TDD mit TYPO3 CMS

1. Klont euch die tea-Extension³.
2. Richtet PHPUnit in PhpStorm ein und lasst die Unit-Tests der Extension in PhpStorm laufen.
3. Richtet die Datenbank-Zugangsdaten für die Tests in PhpStorm ein und lasst die funktionalen Tests laufen.
4. Legt euch eine neue Testliste an.
5. Schreibt testgetrieben (mit Unit-Tests) beim Testimonial-Controller eine `showAction`.
6. Schreibt testgetrieben (mit Functional-Tests) `TestimonialRepository.findOneLatest`, die das zuletzt hinzugefügte Testimonial findet.
7. Schreibt testgetrieben (mit Unit-Tests) beim Testimonial-Controller eine `showLatestAction`.
8. Schreibt einen `Service/TeaBeverageImportService.php`, der TeaBeverage-Texte aus einer Textdatei liest (eins pro Zeile) und diese im Repository neu anlegt. Der Service braucht nicht zu überprüfen, ob es schon Models mit demselben Titel gibt.

³<https://github.com/oliverklee/tea>, alternativ per Composer als `oliverklee/tea`