

## DEMANDER DES ASSERTIONS:

```
assertRequestAttributeValueSame(string $name, string $expectedValue, string $message = '') Affirme que l'attribut :ref:`request <component-foundation-attributes>` est défini sur la valeur attendue.
```

```
assertRouteSame($expectedRoute, array $parameters = [], string $message = '') Affirme que la demande correspond à la route donnée et éventuellement aux paramètres de route.
```

## ASSERTIONS DU NAVIGATEUR:

```
assertBrowserHasCookie(string $name, string $path = '/', string $domain = null, string $message = '')/assertBrowserNotHasCookie(string $name, string $path = '/', string $domain = null, string $message = '')
```

Affirme que le client de test n'a (pas) le cookie défini (ce qui signifie que le cookie a été défini par n'importe quelle réponse dans le test).

```
assertBrowserCookieValueSame(string $name, string $expectedValue, string $path = '/', string $domain = null, string $message = '')
```

Affirme que le cookie donné dans le test Client est défini sur la valeur attendue.

```
assertThatForClient(Constraint $constraint, string $message = '')
```

Affirme la contrainte donnée dans le client. Utile pour utiliser vos assertions personnalisées de la même manière que les assertions intégrées (c'est-à-dire sans passer le client en argument) :

```
ajoute cette méthode dans une classe personnalisée importée dans vos tests  
fonction statique protégée assertMyOwnCustomAssert() : void  
{  
    self::assertThatForClient(new SomeCustomConstraint());  
}
```