```
* assertClientCookieValueEquals()
* assertClientHasCookie()
* assertClientNotHasCookie()
* assertClientRawCookieValueEquals()
* assertHttpCodeEquals()
* assertInputValueEquals()
* assertInputValueNotEquals()
* assertPageTitleContains()
* assertPageTitleEquals()
* assertRequestAttributeValueEquals()
* assertResponseCookieValueEquals()
* assertResponseCookieValueNotEquals()
* assertResponseHasCookie()
* assertResponseHasHeader()
* assertResponseHeaderEquals()
* assertResponseHeaderNotEquals()
* assertResponseIsSuccessful()
* assertResponseNotHasCookie()
* assertResponseNotHasHeader()
* assertResponseRedirects()
* assertRouteEquals()
* assertSelectorContainsText()
* assertSelectorExists()
* assertSelectorNotContainsText()
* assertSelectorNotExists()
```

Check reussite réponse:

```
// Before
$this->assertSame(200, $client->getResponse()->getStatusCode());
// After
$this->assertResponseIsSuccessful();
```

Check redirection vers URL:

```
$this->assertSame(301, $client->getResponse()->getStatusCode());
$this->assertSame('https://example.com', $client->getResponse()->headers->get('Location'));
// After
$this->assertResponseRedirects('https://example.com', 301);
```

Check element contien du texte:

```
// Before
$this->assertContains('Hello World', $crawler->filter('h1')->text());
// After
$this->assertSelectorTextContains('h1', 'Hello World');
```

Assertions de réponse:

```
assertResponseIsSuccessful(string $message = '')
Affirme que la réponse a réussi (l'état HTTP est 2xx)

assertResponseStatusCodeSame(int $expectedCode, string $message = '')
Affirme un code d'état HTTP spécifique.

assertResponseRedirects(string $expectedLocation = null, int $expectedCode = null,
string $message = '')
Affirme que la réponse est une réponse de redirection (vous pouvez éventuellement
vérifier l'emplacement cible et le code d'état).

assertResponseHasHeader(string $headerName, string $message =
'')/assertResponseNotHasHeader(string $headerName, string $message = '')
Affirme que l'en-tête donné est (non) disponible sur la réponse.

assertResponseHeaderSame(string $headerName, string $expectedValue, string
$message = '')/assertResponseHeaderNotSame(string $headerName, string
$expectedValue, string $message = '')
Affirme que l'en-tête donné ne contient (pas) la valeur attendue sur la réponse.

assertResponseHasCookie(string $name, string $path = '/', string $domain = null,
string $message = '')/assertResponseNotHasCookie(string $name, string $path = '/',
string $domain = null, string $message = '')
Affirme que le cookie donné est présent dans la réponse (en vérifiant
éventuellement un chemin ou un domaine de cookie spécifique).

assertResponseCookieValueSame(string $name, string $expectedValue, string $path =
'/', string $domain = null, string $message = '')
Affirme que le cookie donné est présent et défini sur la valeur attendue.

assertResponseFormatSame(?string $expectedFormat, string $message = '')
Affirme que le format de réponse renvoyé par la méthode
:method:`Symfony\\Component\\HttpFoundation\\Response::getFormat` est le même que
la valeur attendue.


.. version ajoutée :: 5.3

    La méthode ``assertResponseFormatSame()`` a été introduite dans Symfony 5.3.
```