

Q

All 6

Passed 6

Failed 0

Flaky 0

Skipped 0

Card Game

exercise.2.spec.js:3

4.8s

webkit

Run

Test Steps

✓ ✓ Before Hooks	1.6s
✓ ✓ fixture: browser	1.3s
✓ browserType.launch	1.3s
✓ ✓ fixture: context	22ms
✓ browser.newContext	17ms
✓ ✓ fixture: page	235ms
✓ browserContext.newPage	234ms
✓ ✓ page.goto(https://deckofcardsapi.com/) — exercise.2.spec.js:5	2.8s
<pre>4 // Navigate to the Card Game API 5 await page.goto('https://deckofcardsapi.com/'); 6 </pre>	
✓ ✓ page.title — exercise.2.spec.js:8	5ms
<pre>7 // Confirm that the site is up 8 const siteIsUp = await page.title(); 9 expect(siteIsUp).toBe('Deck of Cards API');</pre>	
✓ ✓ expect.toBe — exercise.2.spec.js:9	0
<pre>8 const siteIsUp = await page.title(); 9 expect(siteIsUp).toBe('Deck of Cards API');</pre>	
✓ ✓ page.evaluate — exercise.2.spec.js:12	411ms
<pre>11 // Get a new deck of cards 12 const response = await page.evaluate(async () => { 13 const response = await fetch('https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1');</pre>	
✓ ✓ page.evaluate — exercise.2.spec.js:20	413ms
<pre>19 // Shuffle the deck 20 await page.evaluate(async (deckId) => { 21 await fetch('https://deckofcardsapi.com/api/deck/\${deckId}/shuffle/', {</pre>	
✓ ✓ page.evaluate — exercise.2.spec.js:27	400ms
<pre>26 // Deal three cards to each of two players 27 const player1Cards = await page.evaluate(async (deckId) => { 28 const response = await fetch('https://deckofcardsapi.com/api/deck/\${deckId}/draw/?count=3');</pre>	
✓ ✓ page.evaluate — exercise.2.spec.js:33	400ms
<pre>32 33 const player2Cards = await page.evaluate(async (deckId) => { 34 const response = await fetch('https://deckofcardsapi.com/api/deck/\${deckId}/draw/?count=3');</pre>	
✓ ✓ After Hooks	14ms
✓ fixture: page	1ms
✓ fixture: context	0

Attachments

stdout

Player 1 has blackjack!