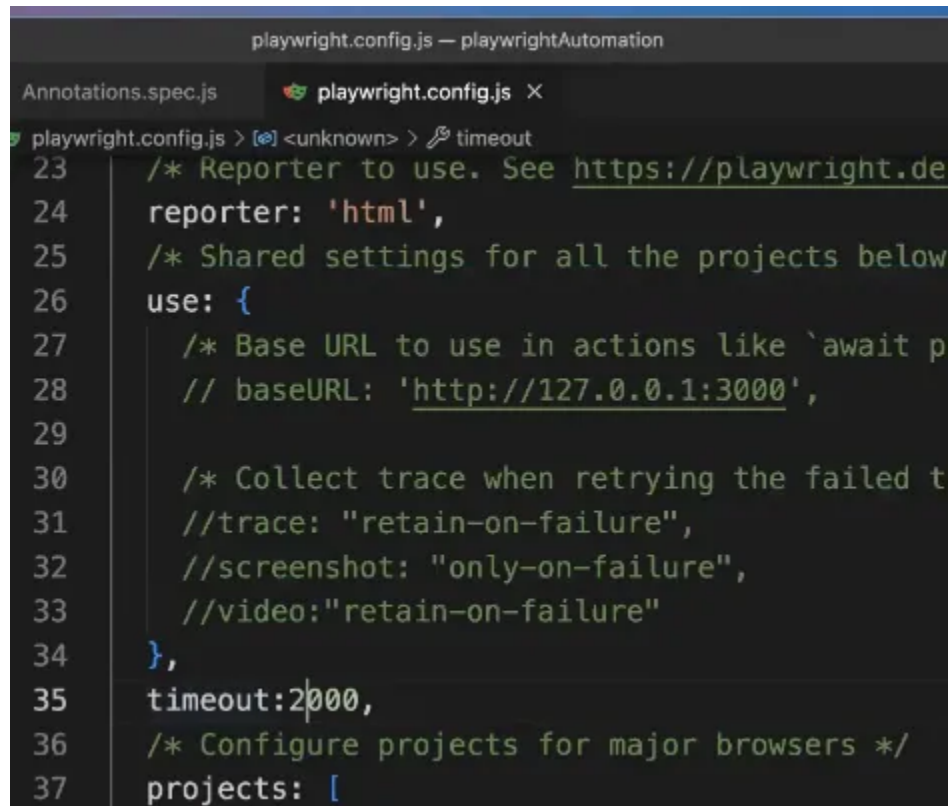


pavanpw-33

Annotations | Only, Skip, Fail, Fixme & Slow

where to give time outs-



```
playwright.config.js — playwrightAutomation
Annotations.spec.js playwright.config.js X
playwright.config.js > [?] <unknown> > ⚙ timeout
23  /* Reporter to use. See https://playwright.dev/docs/reporter.
24  reporter: 'html',
25  /* Shared settings for all the projects below
26  use: {
27    /* Base URL to use in actions like `await page.goto(...)`.
28    // baseUrl: 'http://127.0.0.1:3000',
29
30    /* Collect trace when retrying the failed test.
31    //trace: "retain-on-failure",
32    //screenshot: "only-on-failure",
33    //video: "retain-on-failure"
34  },
35  timeout: 2000,
36  /* Configure projects for major browsers */
37  projects: [
```

```
1  //
2  const {test, expect}=require('@playwright/test')
3
4  // only
5  /*test.only('test1', async({page})⇒{
6      console.log('this is test1')
7  })
8
9  //one test can have multiple only and all only will run.
10 // only
11 /*test.only('test1', async({page})⇒{
12     console.log('this is test1')
13 })
14
15 */
16 //skip
17 /*test.skip('test2', async({page})⇒{
18     console.log('this is test2')
19 })
```

```
1 //skip the test based on some conditions.
2 test('test3', async({page, browserName})=>{
3     console.log('this is test3')
4     if(browserName==='chromium')
5     {
6         test.skip()
7     }
8 })
9 */
10
11 //Fixme
12 //when you know some test has bugs, then you dont want to run it
13 //same as skip
14 /*test('test4', async({page})=>{
15     test.fixme()
16     console.log('this is test 4..')
17
18 })
19 */
20
21 // Fail
22 //when test fails and assertion also fails then test is passed.
23 /*test('test5', async({page})=>{
24     test.fail() //exp
25     console.log('this is test5...')
26     expect(1).toBe(2); //If both exp & Actual is failed then test pass
27 })
28
29 // Fail
30 //the entire test fails because we expected test to fail but expect makes it pass.
31 //output will have "this is test5..." printed.
32 /*test('test5', async({page})=>{
33     test.fail() //exp
34     console.log('this is test5...')
35     expect(1).toBe(1); //If both exp & Actual is failed then test pass
36 })
```

```
1 //fail based on condition.
2 test('test6', async({page, browserName})=>{
3   console.log('this is test6...')
4   if(browserName==='firefox')
5   {
6     test.fail() //exp
7   }
8 })
9 */
10
11 // slow()
12
13 test('test7', async({page})=>{
14   test.slow(); //this will multiply the timeout specified in pw config by three.
15   //or it will multiply default timeout by three.
16
17   //to set the timeout for specific test use the below method.
18   // test.setTimeout(5000)
19   await page.goto('https://www.demoblaze.com/index.html')
20   console.log('this is test 7..')
21 })
```