

Protractors:

Helps us to run test against your browser. no longer wait for wait, sleep methods. Here once test runs complete. Next one will automatically start.

Jasmin:

Behavior driven test environment. Jasmin will allow us to structure your test code properly.

Karma:

Runs test in multiple browsers sometime or in same time.

test runner

//to run testcases run command called "ng test".

//hello.spec.ts

```
describe("check working", () => {
  let expected = '';
  let notexpected = '';
  let expectMatch = null;

  //beforeEach will run before each test .i.e it function
  beforeEach(()=>{
    expected="hello";
    notexpected="hello1";
    expectMatch = new RegExp(/^hello/);
  });

  //afterEach will run after each test .i.e it function
  afterEach(()=>{
    expected='';
    notexpected='';
  })

  it("hello returns hello", () =>
    expect("hello").toBe(expected)
  );

  it("hellotest is not hellotest", () =>
    expect("hello").not.toBe(notexpected)
  );

  it("hellotest expectMatch is hello",()=>{
    expect("hello").toMatch(expectMatch)
  });
})
```



If we don't want to run a test case then just mention x in front of it.

```
//if we dont want to run the test just mention x in front of it
xdescribe("check working", () => {
  let expected = '';
  let notexpected = '';
  let expectMatch = null;

  //beforeEach will run before each test .i.e it function
  beforeEach(()=>{
    expected="hello";
    notexpected="hello1";
    expectMatch = new RegExp(/^hello/);
  });

  //afterEach will run after each test .i.e it function
  afterEach(()=>{
    expected='';
    notexpected='';
  })

  it("hello returns hello", () =>
    expect("hello").toBe(expected)
  );

  it("hellotest is not hellotest", () =>
    expect("hello").not.toBe(notexpected)
  );

  it("hellotest expectMatch is hello",()=>{
    expect("hello").toMatch(expectMatch)
  });
});
```

//if you want to run only one test .i.e describe the remaining ignore then just put f in front of describe like below. The remaining describes are ignore.

```
//if we dont want to run only one test then just put f in front of describe
fdescribe("check working", () => {
  let expected = '';
  let notexpected = '';
```



```

let expectMatch = null;

//beforeEach will run before each test .i.e it function
beforeEach(()=>{
  expected="hello";
  notexpected="hello1";
  expectMatch = new RegExp(/^hello/);
});

//afterEach will run after each test .i.e it function
afterEach(()=>{
  expected='';
  notexpected='';
})

it("hello returns hello", () =>
  expect("hello").toBe(expected)
);

it("hellotest is not hellotest", () =>
  expect("hello").not.toBe(notexpected)
);

it("hellotest expectMatch is hello",()=>{
  expect("hello").toMatch(expectMatch)
});

});

//the remaining below will ignore
describe("second run",()=>{
  it("hello returns hi", () =>
    expect("hi").toBe("hi")
  );
})

```

## What is TestBed ?

**Angular TestBed** link. The **TestBed** is the most important of the **Angular** testing utilities. The **TestBed** creates a dynamically-constructed Angular test module that emulates an Angular @NgModule. The TestBed.configureTestingModule() method takes a metadata object that can have most of the properties of an @NgModule.



Create a service and test it:

//test.service.ts

```
import { Injectable } from "@angular/core";
@Injectable()
export class TestService {
  constructor() { }
  add(a, b) {
    return a + b;
  }
}
```

Create a spec file now

//test.service.spec.ts

```
import { TestService } from './test.service';
import { TestBed, inject } from '@angular/core/testing';

describe("test service", () => {
  beforeEach(() => {
    TestBed.configureTestingModule({
      providers: [TestService]
    });
  });

  it("should be created", inject([TestService], (service: TestService) => {
    expect(service).toBeTruthy();//truthy means its available or not here service
  }));

  it("add function should be there",inject([TestService],(service:TestService)=>{
    expect(service.add).toBeTruthy();//add method is really exists or not
  }));

  it("should add correctly",inject([TestService],(service:TestService)=>{
    expect(service.add(1,2)).toEqual(3);//sending parameters to service add method
  })
  });
});
```

//so finally if we want to run testcases in multiple browsers then we need to update karma configuration file.



```
//open karma.conf.js
```

```
// Karma configuration file, see link for more information
// https://karma-runner.github.io/1.0/config/configuration-file.html

module.exports = function (config) {
  config.set({
    basePath: '',
    frameworks: ['jasmine', '@angular-devkit/build-angular'],
    plugins: [
      require('karma-jasmine'),
      require('karma-chrome-launcher'),
      require('karma-jasmine-html-reporter'),
      require('karma-coverage-istanbul-reporter'),
      require('@angular-devkit/build-angular/plugins/karma')
    ],
    client: {
      clearContext: false // leave Jasmine Spec Runner output visible in browser
    },
    coverageIstanbulReporter: {
      dir: require('path').join(__dirname, '../coverage'),
      reports: ['html', 'lcovonly'],
      fixWebpackSourcePaths: true
    },
    reporters: ['progress', 'kjhtml'],
    port: 9876,
    colors: true,
    logLevel: config.LOG_INFO,
    autoWatch: true,
    browsers: ['Chrome'],
    singleRun: false
  });
};
```

//in above by karma default pulgins are there, we can see chrome also.

If we want to run our test in firefox then we need to install karma-firefox-launcher,so install it

npm install karma-firefox-launcher --save-dev

//now add firefox configuration in karma file.

```
// Karma configuration file, see link for more information
// https://karma-runner.github.io/1.0/config/configuration-file.html

module.exports = function (config) {
  config.set({
    basePath: '',
```



```

frameworks: ['jasmine', '@angular-devkit/build-angular'],
plugins: [
  require('karma-jasmine'),
  require('karma-chrome-launcher'),
  require('karma-firefox-launcher'), //firefox installer
  require('karma-jasmine-html-reporter'),
  require('karma-coverage-istanbul-reporter'),
  require('@angular-devkit/build-angular/plugins/karma')
],
client: {
  clearContext: false // leave Jasmine Spec Runner output visible in browser
},
coverageIstanbulReporter: {
  dir: require('path').join(__dirname, '../coverage'),
  reports: ['html', 'lcovonly'],
  fixWebpackSourcePaths: true
},
reporters: ['progress', 'kjhtml'],
port: 9876,
colors: true,
logLevel: config.LOG_INFO,
autoWatch: true,
browsers: ['Chrome', 'Firefox'], //firefox browser
singleRun: false
});
};

```