**Md MochaChaiDemo**

**Cd MochaChaiDemo**

**Npm init**

**// initialize application > creates package.json file**

**Npm install mocha chai –save**

Add package.json file

npm install mocha chai –save-dev

s

We have 2 files

Hello.js

module.export = function hello()

{

  return "hello";

}

create a folder test

helloTest.js file

const assert = require('chai').assert;

const app = require('../hello');

describe('this is a test ', () =>

{

  it('app shud return hello',()=>

{

   assert.equal('hello','hello');

})

 })

hello.js file

module.export = function hello()

{

return "hello";

}

module.exports = function addNumbers(a, b){

return a+b;

}

const assert = require('chai').assert;

const app = require('../hello');

describe('this is a test ', () =>

{

  it('app shud return hello',()=>

{

   assert.equal('hello','hello');

})

 })

 describe("Add numbers test", function() {

  it("checks valid result", function() {

      var a = 2;

      var b = 1;

      var result = app(a, b);

      assert.equal(result,13);

  });

});

const assert = require('chai').assert;

const app = require('../hello');

describe('this is a test ', () =>

{

  it('app shud return hello',()=>

{

   assert.equal('hello','hello');

})

it("checks valid result", function() {

  var a = 2;

  var b = 1;

  var result = app(a, b);

  assert.equal(result,3);

});

 })

Hello2.js file

module.exports = function hello()

{

  return 'hello';

}

Add Test folder

Hellotest.js file

var assert = require('chai').assert;

var add = require('../hello');

var hello = require('../hello2');

describe ('AAA', function()

{

it('first shud return hello', function()

{

  var res = hello();

  console.log(res);

  assert.equal(res,'hello');

});

it(‘test type', function()

{

  var res = hello();

  assert.typeOf(res,'string');

});

});

Assert.isAbove(result,4);

describe("Add numbers test", function() {

  it("checks valid result", function() {

      var a = 2;

      var b = 1;

      var result = add(a, b);

      assert.equal(result,3);

  });

});

// var assert = require('chai').assert;

// var first = require('../first');

// describe('First', function()

// {

// it('first should return hello', function(){

// assert.equal(first(), 'hello');

// });

// });

var assert = require('chai').assert;

describe('mocha testing with chai assert ', function() {

    it('test case for foo', function() {

        var foo = 'bar';

        assert.typeOf(foo, 'string'); // without optional message

        assert.typeOf(foo, 'string', 'foo is a string'); // with optional message

        assert.equal(foo, 'bar', 'foo equal `bar`');

        assert.lengthOf(foo, 3, 'foo`s value has a length of 3');

    });

    it('check beverages has 3 types of tea', function() {

        var beverages = { tea: [ 'chai', 'matcha', 'oolong' ] };

        assert.lengthOf(beverages.tea, 3, 'beverages has 3 types of tea');

    });

});

Add multiple functions in js file

module.exports =

{

  sayhello: function()

  {

    return 'hello';

  },

  addNumbers :function(x,y)

  {

    return x+y;

  }

}

var assert = require('chai').assert;

var add = require('../hello');

var hello = require('../hello2');

var sayHello = require('../hello2').sayhello;

var add1 = require('../hello2').addNumbers;

describe('Testing Methods of hello file', function()

{

it('shud give hello', function()

{

assert.equal(sayHello(), 'hello');

});

it('Is it string', function()

{

assert.typeOf(sayHello(), 'string');

});

it('add', function()

{

  var r = add1(10,20);

assert.equal(r, 30);

});

});

it('add numbers should be above 5', function()

{

  var r = add1(10,20);

assert.isAbove(r, 5);

});

describe ('AAA', function()

{

it('first shud return hello', function()

{

  var res = hello();

  console.log(res);

  assert.equal(res,'hello');

});

});

describe("Add numbers test", function() {

  it("checks valid result", function() {

      var a = 2;

      var b = 1;

      var result = add(a, b);

      assert.equal(result,3);

  });

});

var app = require('../hello2');

var addNumberResult = app.addNumbers(10,20);

var sayHello = app.sayhello();

describe('Testing Methods of hello file', function()

{

it('shud give hello', function()

{

assert.equal(sayHello, 'hello');

});