

Unit 18.2 Highlights

(Source: https://www.w3schools.com/jquery/event_trigger.asp)

.trigger() - method triggers the specified event and the default behavior of an event (like form submission) for the selected elements

```
$(selector).trigger(event,eventObj,param1,param2,...)
```

(Source: <https://mochajs.org/>)

beforeEach() - hook which runs before each test

afterEach() - hook which runs after each test

done() - callback parameter used when tests are asynchronous

Ex.1

```
//===== test.js=====//
```

```
const expect = require('chai').expect; //load the assertion library of choice; should, expect, assert
```

```
describe('an example of a group of tests', function() {
```

```
  beforeEach(function(done) {
    console.log(`this method will run before EVERY test and should be used for test setup`);
  });
```

```
  afterEach(function(done) {
    console.log(`this method will run after EVERY test and should be used for tear down`);
  });
```

```
  it('should run without error, 3 seconds later', function(done) {
    setTimeout(function(){
      alert("Hello");
      done();
    }, 3000);
  });
});
```

(Source: <https://sinonjs.org/>)

sinonjs - JavaScript framework for creating stand alone test spies, stubs and mocks

sinon.fakeServer.create() - creates a new, fake server

.restore() - restores/resets the native XHR constructor

.respond() - simulates server response for all queued asynchronous requests

.respondWith() - causes the server to respond to any request not matched by another response with the provided data

Ex.2

```
//===== test.js=====//
```

```
var sinon = require('sinon');
var mocha = require('mocha');
const expect = require('chai').expect; //load the assertion library of choice
```

```
//group the tests
```

```
describe('server test group', function() {
  var server;
```

```
  beforeEach(function(done) {
    console.log(`this method will run before EVERY test and should be used for test setup`);
    server = sinon.fakeServer.create();
  });
```

```
  afterEach(function(done) {
    console.log(`this method will run after EVERY test and should be used for tear down`);
    server.restore();
  });
```

```
  it('testing button clicks with fake server', function() {
    server.respondWith('GET', '/users', [
      200, { 'Content-Type': 'application/json' }, JSON.stringify({name: 'Joey', location: 'Atlanta'})
    ]);
    server.respondWith('GET', '/users', [
      200, { 'Content-Type': 'application/json' }, JSON.stringify({name: 'Benjamin', location: 'Atlanta'})
    ]);

    $('#exampleBtn1').trigger('click');
    $('#exampleBtn2').trigger('click');

    server.respond();
    expect($('#result1').text()).to.equal('Joey');
    expect($('#result2').text()).to.equal('Benjamin');
  });
});
```