



NativeScript-Testing

A little about me

- > 20 years doing programming and DevOps
 - > 20 Languages and many OS's
 - > 20 NativeScript Plugins
 - Wrote "THE" book on NativeScript
 - AppProtection.net Encryption Service
 - NativeScript.rocks Plugin, Masters and Community sites
 - Freelance/Contract Developer and Trainer in multiple computer fields and languages.
-
- Blog: <http://fluentReports.com/blog>
 - Twitter: @CongoCart
 - Email: nathan@master-technology.com



So why do testing?



So why do testing?



- Detect failures early when changing code
- Greater customer satisfaction & retention
- Easier to fix when bugs show up immediately
- Ability to verify the product is always in a working state.

Unit Testing

- Treats a piece of code like a black box
- Test both good and bad values to verify values returned are always correct.
- Ability to run tests after making changes to verify functionality is still consistent.





Built in NativeScript Unit Testing

- **tns test init**
 - Answer a simple question about which testing framework you want to use (Jasmine, Mocha or Qunit).
 - Uses Chia for Asserts
 - Everything is installed and ready to go
 - All tests are located in your /app/tests folder
- **tns test android** or **tns test ios**
 - Runs all the tests

Cool Card App

- Code & Tests repo is at <http://github.com/NathanaelA/carder>

Card Example

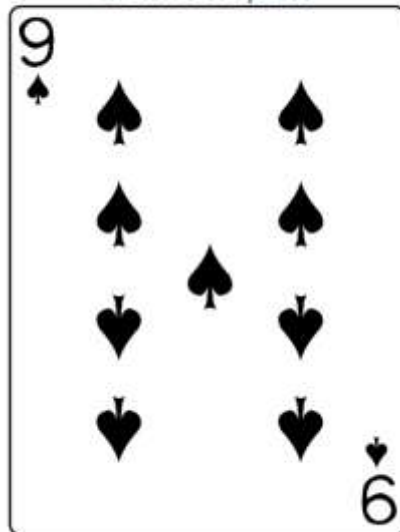
Back of Card



Prior Random Next

Card Example

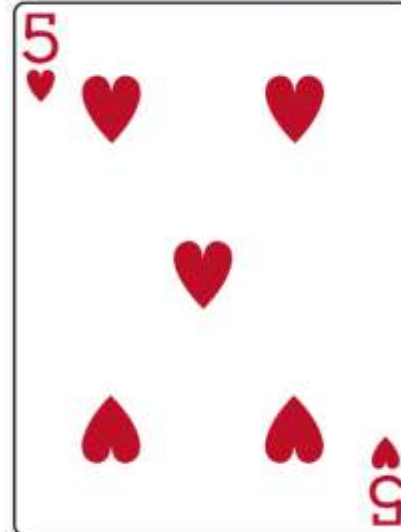
BLACK 9 of Spades



Prior Random Next

Card Example

RED 5 of Hearts



Prior Random Next

Card Example

RED Jack of Diamonds



Prior Random Next

Sample Mocha Test

```
var cards = require('../cards.js');

describe('Cards', function() {
  describe('current card', function() {
    it ('should return Ace of Spades', function() {
      cards.resetDeck();
      var result = cards.getCurrentCard();
      assert.equal(result.suit, 'Spade');
      assert.equal(result.value, 'Ace');
    });
  });
});

describe("next card", function() {
  it ('should return the 2 of spades', function() {
    cards.resetDeck();
    var result = cards.nextCard();
    assert.equal(result.suit, "Spade");
    assert.equal(result.value, "2");
  });
});
```



Test Failure

NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 0 of 6 SUCCESS (0 secs / 0 secs)

NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 1 of 6 SUCCESS (0 secs / 0.006 secs)

NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 2 of 6 SUCCESS (0 secs / 0.008 secs)

NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 3 of 6 SUCCESS (0 secs / 0.008 secs)

NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 4 of 6 SUCCESS (0 secs / 0.009 secs)

NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19) **Cards next card should return the 3 of spades FAILED**

AssertionError: expected 'Heart' to equal 'Spade'

at Context.<anonymous> (data/data/org.nativescript.carder/files/app/tests/cardTest.js:41:11)

at callFn (eval at <anonymous> (data/data/org.nativescript.carder/files/app/tns_modules/nativescript-unit-test-runner/main-view-model.js:203:21), <anonymous>:4351:21)

at Test.Runnable.run (eval at <anonymous> (data/data/org.nativescript.carder/files/app/tns_modules/nativescript-unit-test-runner/main-view-model.js:203:21), <anonymous>:4344:7)

at Runner.runTest (eval at <anonymous> (data/data/org.nativescript.carder/files/app/tns_modules/nativescript-unit-test-runner/main-view-model.js:203:21), <anonymous>:4825:10)

at eval (eval at <anonymous> (data/data/org.nativescript.carder/files/app/tns_modules/nativescript-unit-test-runner/main-view-model.js:203:21), <anonymous>:4931:12)

at next (eval at <anonymous> (data/data/org.nativescript.carder/files/app/tns_modules/nativescript-unit-test-runner/main-view-model.js:203:21), <anonymous>:4745:14)

at eval (eval at <anonymous> (data/data/org.nativescript.carder/files/app/tns_modules/nativescript-unit-test-runner/main-view-model.js:203:21), <anonymous>:4755:7)

at next (eval at <anonymous> (data/data/org.nativescript.carder/files/app/tns_modules/nativescript-unit-test-runner/main-view-model.js:203:21), <anonymous>:4681:14)

at eval (eval at <anonymous> (data/data/org.nativescript.carder/files/app/tns_modules/nativescript-unit-test-runner/main-view-model.js:203:21), <anonymous>:4723:5)

at timeslice (eval at <anonymous> (data/data/org.nativescript.carder/files/app/tns_modules/nativescript-unit-test-runner/main-view-model.js:203:21), <anonymous>:75:27)

NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 5 of 6 (1 FAILED) (0 secs / 0.082 secs)

NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 6 of 6 (1 FAILED) (0 secs / 0.087 secs)

NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 6 of 6 (1 FAILED) (0.24 secs / 0.091 secs)

Broken Code

```
exports.nextCard = function() {  
  currentCard++;  
  if (currentCard > 13) {  
    currentCard = 0;  
    currentSuit++;  
    if (currentSuit > 3) {  
      currentSuit = 0;  
    }  
  }  
  return exports.getCurrentCard();  
};
```



Fixed Code

```
exports.nextCard = function() {  
  currentCard++;  
  if (currentCard > 12) {  
    currentCard = 0;  
    currentSuit++;  
    if (currentSuit > 3) {  
      currentSuit = 0;  
    }  
  }  
  return exports.getCurrentCard();  
};
```



Tests Passing

NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 0 of 6 SUCCESS (0 secs / 0 secs)
NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 1 of 6 SUCCESS (0 secs / 0.005 secs)
NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 2 of 6 SUCCESS (0 secs / 0.007 secs)
NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 3 of 6 SUCCESS (0 secs / 0.008 secs)
NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 4 of 6 SUCCESS (0 secs / 0.009 secs)
NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 5 of 6 SUCCESS (0 secs / 0.083 secs)
NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 6 of 6 SUCCESS (0 secs / 0.085 secs)
NativeScript / 19 (4.4.4; Samsung Galaxy S5 - 4.4.4 - API 19): Executed 6 of 6 SUCCESS (0.217 secs / 0.089 secs)



End to End Testing

- Appium
 - Supports iOS and Android
 - Full end to end testing
- NativeScript-dev-Appium
 - **tns install appium**
 - Installs everything you need
 - All tests are located in **/e2e-tests**



Finding the Elements

- Returns a Single Element:
 - `elementByAccessibilityId`
 - `elementByXPath`
- Returns multiple elements:
 - `elementsByAccessibilityId`
 - `elementsByXPath`
- XPath helper function:
 - `nsAppium.xpath(<NativeScript Class Name>)`



Appium Test

```
var nsAppium = require("nativescript-dev-appium");

describe("Carder app tests", function () {
  this.timeout(100000);
  var driver;

  before(function () {
    driver = nsAppium.createDriver();
  });

  it("should find an the card title", function () {
    return driver
      .elementByAccessibilityId("message")
      .text().should.become("Back of Card");
  });

  it("should flip the card over", function() {
    var spade = '♠'; // This is the Spade PIP character
    var xpath = "//*[@text='"+spade+"']";

    return driver.elementByAccessibilityId("card")
      .should.eventually.exist
      .tap()
      .elementsByXPath(xpath).should.eventually.have.length(3);
  });
});
```


Declarative UI - AccessibilityId

```
<Page>
  <StackLayout>
    <Label id="cardExample" text="Card Example"/>
    <Label id="message" automationText="message" text="{{ text }}" />
    <StackLayout automationText="card" class="{{ color }}" tap="click">
      <GridLayout rows="auto,auto,*,auto,auto" columns="auto,*,auto">
        <Label text="{{ value }}" row="0" col="0" class="cardLeftValue"/>
        <Label text="{{ pip }}" row="1" col="0" class="cardLeftPip"/>
        <Label text="{{ pip }}" row="2" col="1" class="pip"/>
        <Label text="{{ pip }}" row="3" col="2" rotate="180" class="cardRightPip"/>
        <Label text="{{ value }}" row="4" col="2" rotate="180" class="cardRightValue"/>
      </GridLayout>
    </StackLayout>
    <StackLayout orientation="horizontal" class="button">
      <Button automationText="prior" text="Prior" tap="prior" />
      <Button automationText="random" text="Random" tap="random" />
      <Button automationText="next" text="Next" tap="next" />
    </StackLayout>
  </StackLayout>
</Page>
```

So why do testing?



So you aren't the

♠ JOKER



Resources

- NativeScript Unit Testing:
<http://docs.nativescript.org/tooling/testing>
- Mocha - <https://mochajs.org/>
- Chia Assert – <http://chaijs.com/api/assert/>
- Appium - <http://appium.io/>
- Appium WebDriver - <https://github.com/admc/wd>
- Getting Started with NativeScript (my book) -
<http://nativescript.tools/product/8>
 - Instructions on Unit Testing, Proxyquire Testing, and Appium Testing.

Contact: nathan@master-technology.com