Play by Play – Angular Testing

***URL***: <https://app.pluralsight.com/library/courses/play-by-play-angular-testing-papa-bell>   
  
***Local dir:*** C:\Bob\dev\pluralsight\angular-test

***Nodejs***: to check what version of node is running 🡪 > **nvm list**  
  
***From cmd prompt:***  
> gulp dev-serve --force  
  
> gulp test  
  
> gulp serve-specs  
  
  
  
**TESTING**  
  
 >gulp serve -specs  
 >gulp test

Test URL: <http://localhost:3002/specs.html>

Mock Data:  
 In src\client\app\people.controller.spec.js  
 var people = mockData.getMockPeople();  
  
 Mock Data comes from --> src\client\test-helpers\mock-data.js  
  
  
 **Bard**  
 A lib written by Ward Bell (from this course)  
 It does not allow the use of an injected $state, so bard.inject('$state') does not work  
 Bard provides general purpose utils for testing purposes  
 bard.inject will inject services for you, then removes them after each test.  
 ex/ in people.controller.spec.js  
  
 bard.inject('$controller', '$log', '$q', '$rootScope', 'dataservice')  
  
 We also mock the promise using $when, which returns a full-resolved promse  
 ex/ return $q.when(people);  
  
 ADDING A NEW SPEC TEST  
 We must invoke the gulp dev-spec in order to inject the new spec.js file into index.html  
  
 Ajax calls in angular mocks:  
 You cannot make ajax calls via $http  
 You must use $httpBackend, which simulate a real backend call with http responses  
  
  
 Bard Testing Issues  
 - In people.controller.spec.js, we reference $state in one of the tests, but it fails  
 - In chrome console, we get error "$state.go is not a function"  
 - They suggested adding "bard.appModule('app.people');" at the top to use Angular test module, but it still doesn't work  
 - In chrome console, $state returns "Object {\_faked: "this is the faked $state service"}"  
  
 Test Coverage Report  
 - Launch C:\Bob\dev\pluralsight\angular-test\report\coverage\report-html\index.html