

```

describe('Login Controller', function() {
  beforeEach(module('cgiAdpq'));

  var $rootScope,
      controller,
      scope,
      $q,
      deferred,
      $httpBackend,
      authService,
      $localStorage,
      $state;

  beforeEach(inject(function ($rootScope, $controller, _$q_, _$rootScope_,
    _$httpBackend_, _authService_, _localStorage_, _$state_) {
    $rootScope = _$rootScope_;
    $q = _$q_;
    $httpBackend = _$httpBackend_;
    authService = _authService_;
    $localStorage = _localStorage_;
    $state = _$state_;

    deferred = $q.defer();

    $httpBackend.whenGET('main/home.html').respond('home view');
    //$httpBackend.expectGET('staticPages/home.html');

    $httpBackend.whenGET('user/login.html').respond('login page');
    //$httpBackend.expectGET('user/login.html');

    //spyOn(authService, 'login').and.returnValue(deferred.promise);

    scope = $rootScope.$new();
    controller = $controller('LoginController', {
      $scope: scope
    });
  }));

  afterEach(function() {
    $httpBackend.flush();
  });

  it('has blank credentials', function () {
    expect(scope.credentials).toEqual({email: '', password: ''});
  });

  it('successfully logs in', function () {
    var password = 'testing' + new Date().getDate();
    scope.credentials = {email: 'testing@test.com', password: password};

    scope.login();

    scope.$apply();
  });

  it('fails logging in', function () {
    var password = 'testing';
    scope.credentials = {email: 'testing@test.com', password: password};
  });

```

```
scope.login();
scope.$apply();
});
it('remembers the last route prior to login', function () {
  var password = 'testing' + new Date().getDate();
  scope.credentials = {email: 'testing@test.com', password: password};

  var nextState = {name: 'home', params: {}};
  $localStorage.nextState = nextState;

  scope.login();
  scope.$apply();
  //expect($localStorage.nextState).toBe(undefined);
});
});
```