```
describe('Controller: OnlineMortgage.StrategyCtrl', function() {

    // Instantiate a new version of module before each test
    beforeEach(module('OnlineMortgage.StrategyApp'));

    var scope, onlineMortgageStrategyCtrl, onlineMortgageMock;

    // Before each unit test, instantiate a new instance of the
    // controller
    beforeEach(inject(function($rootScope, $controller) {
        // define the mock for onlineMortgage service
        onlineMortgageMock = {
            tranches: []
        };
        scope = $rootScope.$new();
        scope.onlineMortgage = onlineMortgageMock;
        onlineMortgageStrategyCtrl =
        $controller('OnlineMortgage.StrategyCtrl', {
            $scope: scope
        });
    }));

    it('remove justAddedTranche tranche', function() {
        var justAddedTranche = {
            kind: 'variable',
            duration: 4,
            amount: 300000
        };
        scope.onlineMortgage.tranches = [justAddedTranche];
        expect(scope.onlineMortgage.tranches.length).toEqual(2);
        expect(justAddedTranche.id).not.toBeDefined();
        var justAddedTrancheIndex =
        scope.onlineMortgage.tranches.indexOf(
            justAddedTranche
        );
        scope.deleteTranche(justAddedTrancheIndex);
        expect(scope.onlineMortgage.tranches.length).toEqual(1);
        expect(scope.onlineMortgage.tranches)
        .not.toContain(justAddedTranche);
    });
```

JASMINE: SPEC STYLE OF TESTING

Jasmine Syntax

- describe
- beforeEach
- it
- afterEach
- expect

Useful Jasmine Matchers

- toEqual
- toBe
- toBeTruthy
- toBeFalsy
- toBeDefined
- toBeUndefined
- toBeNull
- toContain
- toMatch

MOCKING OUT SERVICES

Inline mock

- The first way is to override the service during the unit test, as an **inline mock**

Global mock

- The second option to override services would be at a **global level** instead of a unit test level.