

ISSUES WE MET

2. Mocking service that returns class with extended prototype

```
function OnlineMortgageCalculationService($log) {  
  
    function CalculationClass(options) {  
        var defaults = {  
            LTV_LIMIT_1: {  
                MAIN_RESIDENCE: 0.65,  
                SECOND_RESIDENCE: 0.65  
            },  
            RATE_DELTA: 0.01  
        };  
  
        angular.extend(defaults, options);  
  
        this.LTV_LIMIT_1 = defaults.LTV_LIMIT_1;  
        this.RATE_DELTA = defaults.RATE_DELTA;  
        this.onlineMortgage = defaults.onlineMortgage;  
        this.constants = getDjangoParam('constants');  
    }  
  
    CalculationClass.prototype.getWishedMortgageAmount = function() {  
  
        if (this.onlineMortgage.is_refinancing) {  
            return this.onlineMortgage.current_mortgage_amount;  
        }  
  
        var totalAssets = parseInt(this.onlineMortgage.total_assets, 10);  
        valueOfProperty = parseInt(this.onlineMortgage.value_of_property, 10);  
  
        var mortgageAmount = Math.max(  
            valueOfProperty * this.getLTV1Limit(),  
            valueOfProperty - totalAssets  
        );  
  
        return Math.floor(mortgageAmount);  
    };  
};
```

```
this.initMortgageAmount = function() {  
    var mortgageAmountService =  
        new OnlineMortgageCalculationService(  
            {onlineMortgage: $scope.onlineMortgage}  
        );  
    $scope.recommendedMortgageAmount =  
        mortgageAmountService.getWishedMortgageAmount();  
  
    this.getMortgageAmount();  
};
```

```
// Before each test create mock for getWishedMortgageAmount method in  
// OnlineMortgageCalculationService and return recommendedMortgageAmount  
beforeEach(module(function($provide) {  
    function mockCalculationClass() {}  
    mockCalculationClass.prototype.getWishedMortgageAmount = function() {  
        return recommendedMortgageAmount;  
    };  
    $provide.value(  
        'OnlineMortgageCalculationService', mockCalculationClass  
    );  
}));
```

TEST COVERAGE

all files online-mortgage/controllers/

65.71% Statements 46/70 50% Branches 9/18 52.94% Functions 9/17 65.71% Lines 46/70

File	Statements	Branches	Functions	Lines
strategy-ctrl.js	65.71% 46/70	50% 9/18	52.94% 9/17	65.71% 46/70

```
79 1x function deleteTranche(index) {
80 3x   if ($scope.onlineMortgage.tranches[index].id) {
81 1x     $scope.onlineMortgage.tranches[index].to_delete = $scope.onlineMortgage.tranches[index].id;
82   } else {
83 2x     $scope.onlineMortgage.tranches.splice(index, 1);
84   }
85   // if the last one was deleted, add empty one
86 3x   if ($scope.onlineMortgage.tranches.length === 0) {
87 1x     $scope.addEmptyTranche();
88   }
89 }
90
91 1x function openTrancheDatepicker(event, tranche) {
92 1x   event.preventDefault();
93 1x   event.stopPropagation();
94 1x   tranche.opened = true;
95 }
96
97 // -----Callbacks-----
98 9x var trancheSaveSuccessCallback = function(errors) {
99   window.location.href = $scope.onlineMortgage.next_step_url;
100 };
101
102 9x var trancheSaveErrorCallbackFactory = function(tranchesInfoPromise) {
103   var trancheSaveErrorCallback = function(resp) {
104     serverSideValidation.groups.tranchesForm.showErrorsOnFail(tranchesInfoPromise);
105
106     if (resp.data.tranches.hasOwnProperty('non_field_errors')) {
107       var errorText = '';
108       var nonFields = resp.data.tranches.non_field_errors;
109       if (nonFields) {
110         if (nonFields.length > 1) {
111           _.forEach(nonFields, function (error) {
112             errorText = errorText + '<br>' + error;
113           });
114           return errorText;
115         }
116       }
117     }
118   };
119   return trancheSaveErrorCallback;
120 }
```