

Sarah Dulaney's Sample Code

LOCALSTORAGE IN ANGULAR

```
/** @fileoverview Angular Provider. Dependencies ($angularCacheFactory)***/

(function(angular) {
  "use strict";

  angular.module("odin.data").factory("benefitsClaimTileCacheProvider", ["$angularCacheFactory",
function($angularCacheFactory) {
  var cache = $angularCacheFactory("benefitsClaimTile", {
    storageMode: "localStorage"
  });

  return {
    cache: cache
    //,
    //ON_USER_CHANGE: "USER_ID_CHANGE"
  };
}]);
})(window.angular);
```

FILTER

```
fundFilters.filter('naCheckPercent', function ($filter) {
  return function (naCheckValue, precision) {

    var naCheckValueReturn = $filter('naCheckPrecision')(naCheckValue, precision);

    if (angular.isDefined(naCheckValueReturn) && naCheckValueReturn !== "na") {
      naCheckValueReturn = naCheckValueReturn + "%";
    }

    return naCheckValueReturn;
  };
});
```

CONTROLLER

```
(function () {
  'use strict';

  /**
   * Controller for handling the Claim Tile
   */
  angular.module("odin.page").controller("benefitsClaimTileCtrl",

  ["$scope", "$http", "benefitsDashboardProvider", 'flashProvider', "viewedClaimsTileProvider", "benefitsClaimTileCacheProvider", "benefitClaimsProvider", "$window", "$filter",
   function
  ($scope, $http, benefitsDashboardProvider, flashProvider, viewedClaimsTileProvider, benefitsClaimTileCacheProvider, benefitClaimsProvider, $window, $filter) {

    $scope.showTile = false;

    /**
     * Returns the claim data for the claim status based on the date of claim
     */
    var fetchClaimTimeData = function(){

      $scope.data = {}; //Scope Data

      //For asynch Data
      $scope.isError = false;
      $scope.showContent = false;
      $scope.loadedData = false;

      var localStorage = $window.localStorage;
      var totalClaimCount;
      var claimNumber;
      var cachedClaimCount = 0;
      var claimsArray;

      benefitClaimsProvider.query().then(function (providerData){

        //Check for error up front:
        if(providerData.error){
          flashProvider.to('global').error(providerData.errorMsg);
          $scope.loadedData = true;
          $scope.isError = true;
        }
        else{
```

```

//Set the data for our scope
$scope.data = providerData;
$scope.loadedData = true;
$scope.showContent = true;
$scope.claimCount = providerData.newClaims;

//Inspect local stored cached claim data and set service params.
//We need local storage because we cannot use route params due to CQ's use of JSP's for
components

if(localStorage){
  claimsArray = benefitsClaimTileCacheProvider.cache.get("claimTileNumberSearch");
  if ( claimsArray ) {
    for (var i = providerData.claims.length; i--;) {
      if ( !providerData.claims[i].isNewClaim ) continue;
      claimNumber = providerData.claims[i].claimNumber;
      // compare the claim number to the cached claim number the user might have viewed only if
it's a new claim
      for(var j = claimsArray.length; j--;) {
        if (claimNumber === claimsArray[j].claimNumber) {
          cachedClaimCount++;
          break;
        }
      }
    }
    $scope.claimCount = $scope.claimCount - cachedClaimCount;
    if ( $scope.claimCount < 0 ) $scope.claimCount = 0;
  }
}
});
/**
 * Returns the proper DATETIME for the claim status based on the date of claim
viewedClaimsTileProvider.setViewClaimsUpdate();
 */

/**
 * Check and see if we can show this tile from our Benefits Dashboard Provider
 * @param {dashboardKey} string String key for dashboard to check on
 */

benefitsDashboardProvider.getAvailableUserTiles("CLAIMACTIVITY").then(function(data){
  if(data){

```

```

        if(data.errMsg){
            flashProvider.to('global').error(data.errMsg);
        }
        else {
            $scope.showTile = true;
            fetchClaimTimeData();
        }
    }

}, function(response) {
    flashProvider.to('global').error("Error calling benefits dashboard provisioning");
    fetchClaimTimeData();
});
}
]);
})();

```

PROVIDER

```

(function(angular) {
    "use strict";
    angular.module("odin.common").factory("viewedClaimsTileProvider",
        ['$http', "benefitClaimsProvider", "benefitsCacheProvider", "$filter",
function($http, benefitClaimsProvider, benefitsCacheProvider, $filter) {

    /** Returns the proper count for the claim status based on the date of claim */
    return {

        setViewClaimsUpdate: function() {
            var localTime = new Date();
            var updatedTime = $filter("toDateTime")(localTime);
            var timestampUrl = ids.data.hapi.baseUrl('/benefits/user/tracks/v1.0/index.json');
            //var timestampUrl = "/etc/designs/odin/clientlibs/js/mock-json/user-tracks-timestamp.json";

            var request = $http({
                method: "post",
                url: timestampUrl,
                headers: { 'Content-Type': 'application/json' },
                data:
{"TrackingInfo":{"userInterfaceTrackingType":{"lastViewedDateTime":{"dateTime":updatedTime}}}}
            });
            return request;
        },

        compareTimeClaimCount: function (providerData){
            var claimCount = 0;

```

```

var clmLastDateOfSrvcConv;
var claimLastViewed;

if (providerData.claims.length > 0){
  for(var l = 0; l < providerData.claims.length; l++){
    clmLastDateOfSrvcConv = $filter("toDate")(providerData.claims[l].clmLastDateOfSrvc);
    claimLastViewed = providerData.claims[l].claimLastViewed;
    if (clmLastDateOfSrvcConv > claimLastViewed) {
      claimCount++;
    } else {
    }
  }
}
else{
  // defaults to 0 as defined
}
return claimCount;
}
};
});
})(window.angular);

```

DIRECTIVE

```

fundsBasicsDirectives.directive('topTenHoldings', function () {
  var highchartsObj = {
    priority: 15,
    replace: true,
    transclude: false,
    template: "<div id=\"topTenHoldingsHighChartContainer\" class=\"summary-pie-donut floatLeft\"></div>",
    link: function (scope, element, attrs) {
      attrs.$observe('topTenHoldings', function (topTenHoldingsValue) {

        var topTenHoldingsText;
        var topTenHoldings;
        if (topTenHoldingsValue !== '' && typeof(topTenHoldingsValue) !== 'undefined' &&
topTenHoldingsValue !== 0) {
          topTenHoldings = topTenHoldingsValue*1;
          topTenHoldingsText = topTenHoldingsValue + '%';
        } else {
          topTenHoldingsText = 'na';
          topTenHoldings = 0;
        }

        var otherTopTenHoldingsValue = 100-topTenHoldings;

```

```

var topTenHoldingsPieStart = 0;
if (76 < topTenHoldings && topTenHoldings <= 99.95) {
    topTenHoldingsPieStart = 45 + ((-0.5) * topTenHoldings);
} else if (46 < topTenHoldings && topTenHoldings <= 75) {
    topTenHoldingsPieStart = (45 - topTenHoldings)
} else if (26 < topTenHoldings && topTenHoldings <= 45) {
    topTenHoldingsPieStart = (otherTopTenHoldingsValue / 2);
} else if (16 < topTenHoldings && topTenHoldings <= 26) {
    topTenHoldingsPieStart = (45 + (topTenHoldings / 2));
} else if (8 < topTenHoldings && topTenHoldings < 16) {
    topTenHoldingsPieStart = (45 + (topTenHoldings * 2));
} else if (topTenHoldings < 8) {
    topTenHoldingsPieStart = (45 + (topTenHoldings * 5));
}

var left = 84;
var top = 51.5;
var size = 10;
//topTenHoldingsChart is initialized in the model.js
topTenHoldingsChart.plotOptions.pie.startAngle = topTenHoldingsPieStart;
topTenHoldingsChart.chart.renderTo = 'topTenHoldingsHighChartContainer';
topTenHoldingsChart.plotOptions.pie.pieStart = topTenHoldingsPieStart;
topTenHoldingsChart.title.text = '<span style="font-size:12px; font-weight:bold">' +
topTenHoldingsText + '</span>';
topTenHoldingsChart.series[0].data = [];
topTenHoldingsChart.series[0].data.push({name: 'topTenHoldings', y: topTenHoldings, color:
'#8cab65'});
topTenHoldingsChart.series[0].data.push({name: 'otherIssuerValue', y: otherTopTenHoldingsValue,
color: '#cccccc'});
topTenHoldingsChart.plotOptions.pie.visible = true;
if (topTenHoldings >= 8) {
    var chartJ = new Highcharts.Chart(topTenHoldingsChart, function (chart) { // on complete
        chart.renderer.image('/content/fund/app/img/diamond.gif', left,top, size, size).add();
    });
}
else {
    var chartJ = new Highcharts.Chart(topTenHoldingsChart);
}
})
},
restrict: 'A'
};
return highchartsObj;
})

```

JQUERY SIMPLE EXAMPLE

```
$(document).ready(function(){
  $("#clicker").toggle(
    function(){$("#clicker").css({"color":"purple"});},
    function(){$("#clicker").css({"color":"blue"});},
    function(){$("#clicker").css({"color":"orange"});
  });
});
```

<p id=clicker">Click here.</p>

AJAX

```
$('#document').ready(function () {
  $.ajax({
    url: "/app/content/ecl",
    dataType: "json",
    type: "get",
    data: {
      taxonomyNodeId: "1344468",
      requestingHTML: false
    },
    success: function (json_response) {
      var banner="banner";
      showHTML(json_response, $('#.banner'));
    },
    error: function (xhr, ajaxOptions, errorThrown) {
      // Show Nothing
    }
  }); // end of ajax call
}); // end document ready
```

/ *****Reformat ECL Banner Response***** */*

```
function showHTML(json_response, banner) {
  if (json_response.response[0].status == "success") {
    var bannerObj = json_response.response[1].taxonomyNodeList;
    var markup = "";
    var carousel = [];
    carousel.push("<ul class=\"articles_list bqqs flexslider\">" + "<li>" + "<ul class=\"slides\">");
    for (x = 0; x < bannerObj.length; x++) {
      carousel.push("<li>" + "<div id=\"imageHolder\">" + "<a href=\"#\ " class=\"article\">" + "<div
class=\"articleImageContainer\">" + "<div class=\"articleImage\">" +
getCarouselImage(bannerObj[x].itemTypeId, bannerObj[x].contentItemId,
bannerObj[x].itemLinkDomainName, bannerObj[x].itemThumbnailRelativeUrlText)+ "</div>" + "<div
class=\"articleShadow\">" + "</div>" + "</div>" + "</a>" + "</div>");
    }
  }
}
```

```

        carousel.push("<h2 class=\"article_title\">" + "<a href=\"#" + bannerObj[x].taxonomyNodeName +
"</a>" + "</h2>");
        carousel.push("<div id=\"article_date\">" + bannerObj[x].itemLastModifiedDate + "</div>");
        carousel.push("<div id=\"article_body\">" + "<a href=\"#" + bannerObj[x].itemDescription + "</a>" +
"</div>" + "</li>");
    }
    carousel.push("</ul>");
}
//join markup
banner.html(carousel.join(""));

```

```

$( document ).ajaxSend(function( event, request, settings ) {
    $( "#requestMessage" ).append( "<li>Sending request for " + settings.url + "</li>" );
});

```