Article
Chris Stewart · Sep 8, 2017 3m read

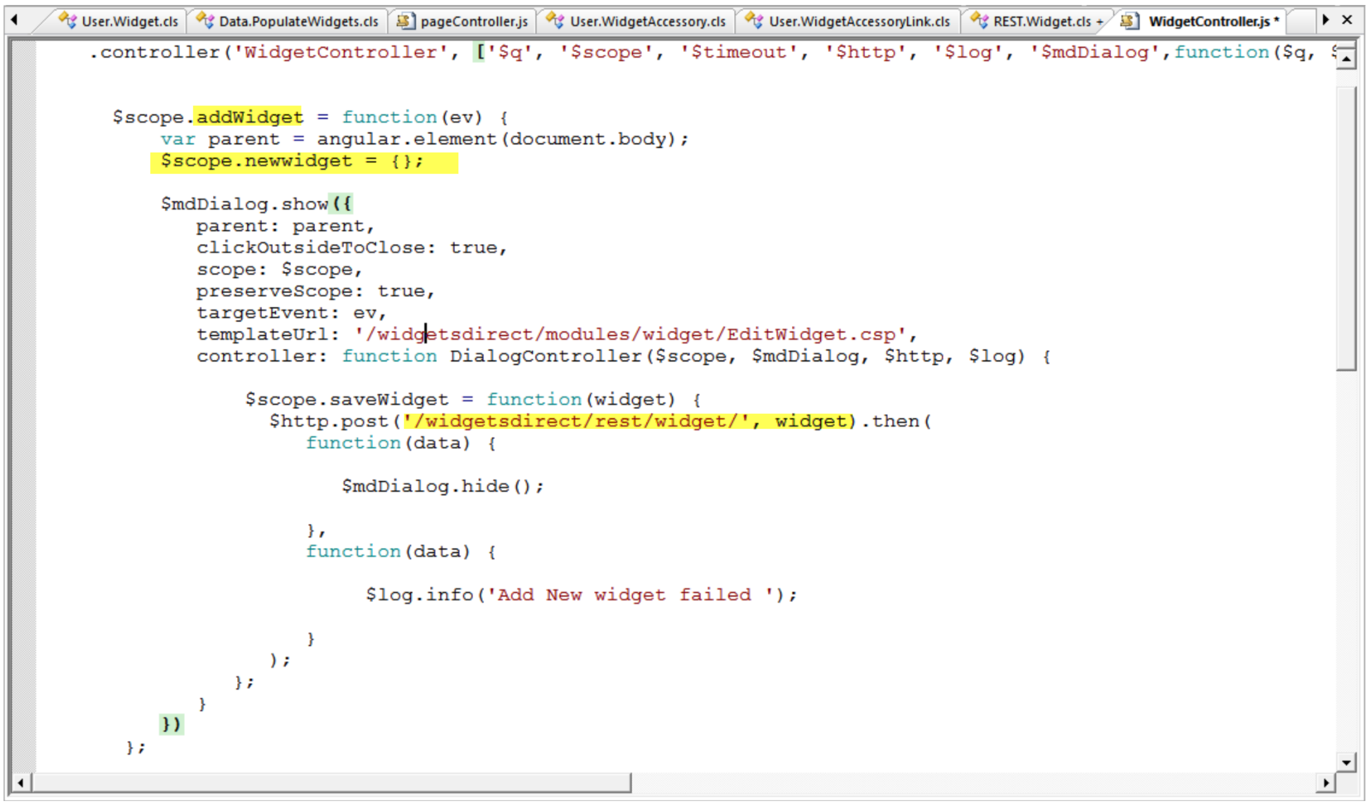
Let's write an Angular 1.x app with a Caché REST backend - Part 13

a.k.a.. "The World of Widgets Returns!" or "Paternity leave damages Instructional Series momentum"

In our last lesson, we combined 2 separate classes to appear as the same property. We now have the ability to Update our Widget catalog, but what if we want to Create a Widget? Thankfully, we've already done 90% of what we need, just by implementing Edits

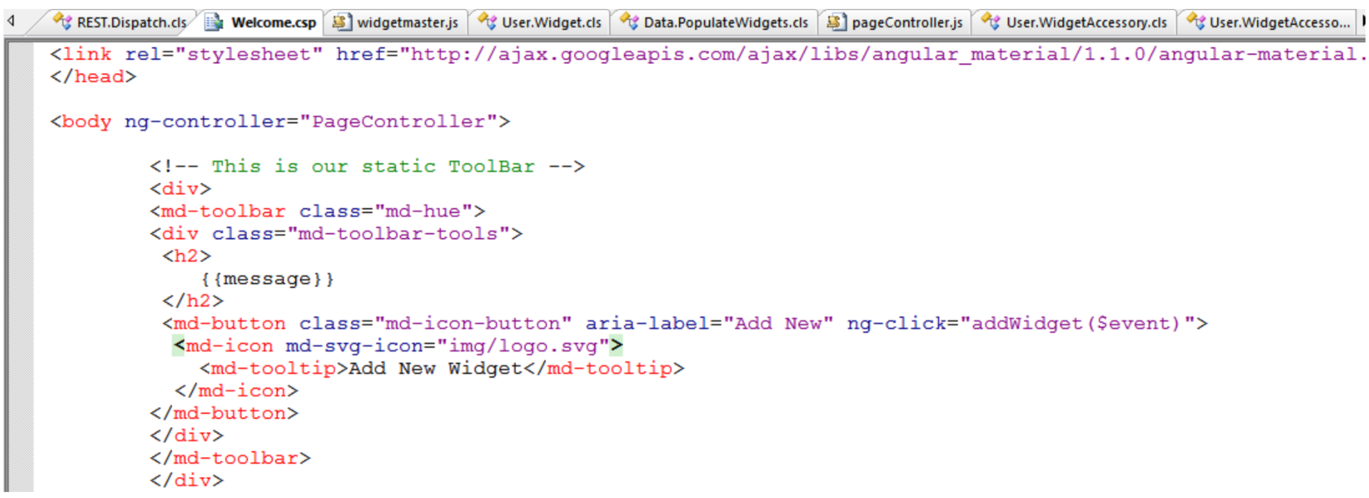
As we mentioned when creating the REST Services for PUT and POST, the only real difference between creating and updating a record is whether we are passing in an existing ID or creating a %New record. The actual content of the Widget JSON is exactly the same, so this allows us to be a little lazy and reuse the form and controller code we have previously written, with just some minor edits to allow it to work for New Widgets

First, we copy the editWidget function and create addWidget from it in WidgetController.js. This will not take in an existing Widget, and will instead create a blank JSON object to bind the scope to. We will remove that parameter from the method, and change the assignment to an object constructor. The only remaining change we need to make is to make this version of the saveWidget function call the POST method without a supplied Widget ID. We don't need to make any changes to the form, so we can just include this template as it is



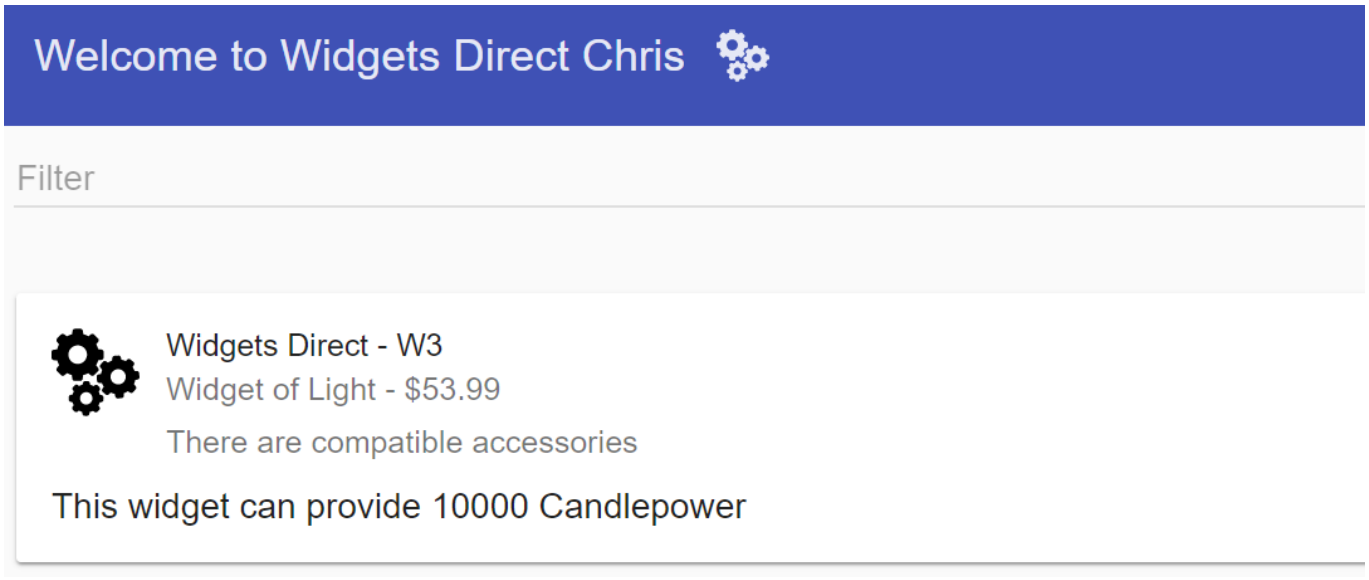
```
    .controller('WidgetController', ['$q', '$scope', '$timeout', '$http', '$log', '$mdDialog',function($q,


      $scope.addWidget = function(ev) {
          var parent = angular.element(document.body);
          $scope.newwidget = {};

          $mdDialog.show({
             parent: parent,
             clickOutsideToClose: true,
             scope: $scope,
             preserveScope: true,
             targetEvent: ev,
             templateUrl: '/widgetsdirect/modules/widget/EditWidget.csp',
             controller: function DialogController($scope, $mdDialog, $http, $log) {

                 $scope.saveWidget = function(widget) {
                   $http.post('/widgetsdirect/rest/widget/', widget).then(
                      function(data) {

                         $mdDialog.hide();

                      },
                      function(data) {

                           $log.info('Add New widget failed ');

                      }
                   );
                };
             }
          })
      };
```

So, all we need to do now, is add a button to the Toolbar on Welcome.csp to trigger this new function (we'll just use the existing icon image for this for now)

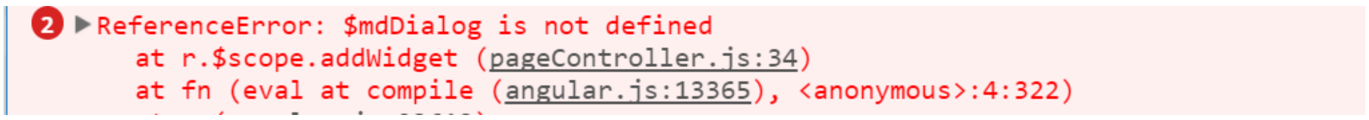


```
<link rel="stylesheet" href="http://ajax.googleapis.com/ajax/libs/angular_material/1.1.0/angular-material.
</head>

<body ng-controller="PageController">

        <!-- This is our static ToolBar -->
        <div>
        <md-toolbar class="md-hue">
        <div class="md-toolbar-tools">
         <h2>
            {{message}}
         </h2>
         <md-button class="md-icon-button" aria-label="Add New" ng-click="addWidget($event)">
          <md-icon md-svg-icon="img/logo.svg">
            <md-tooltip>Add New Widget</md-tooltip>
          </md-icon>
        </md-button>
        </div>
        </md-toolbar>
        </div>
```

Now when we load Welcome.csp, we should see our new button. Let's click it to Create a new Widget.

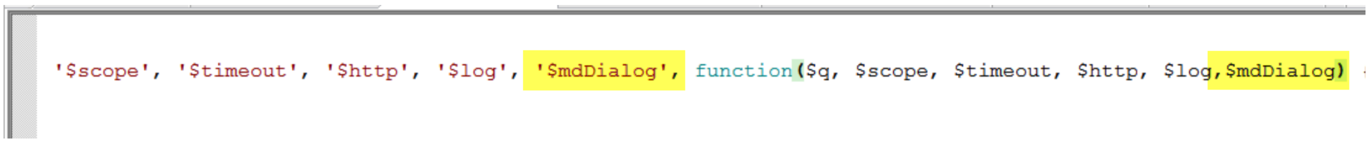

Welcome to Widgets Direct Chris
Filter
Widgets Direct - W3
Widget of Light - $53.99
There are compatible accessories
This widget can provide 10000 Candlepower

Nothing happens. It's almost like the function doesn't exist. This is because to the current $scope at that part of the page, the function doesn't exist. We don't actually reference WidgetController until the repeating section of Widgets, so the page header does not know how to service the ng-click command. To solve this, move the AddWidget function to pageController.js, as this is the active scope. Refresh your page, and click the button again. It's still failing. So let's fire up F12 Debug tools to search for clues

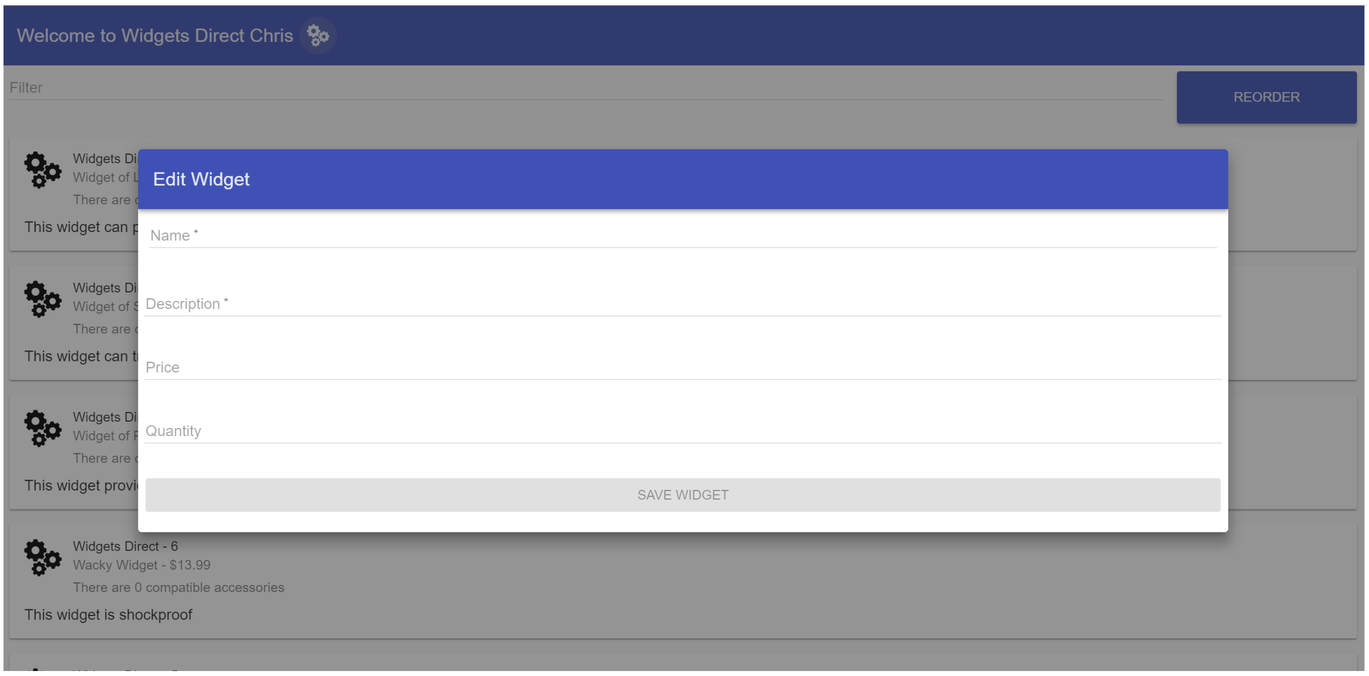


```
2 ▶ReferenceError: $mdDialog is not defined
      at r.$scope.addWidget (pageController.js:34)
      at fn (eval at compile (angular.js:13365), <anonymous>:4:322)
```

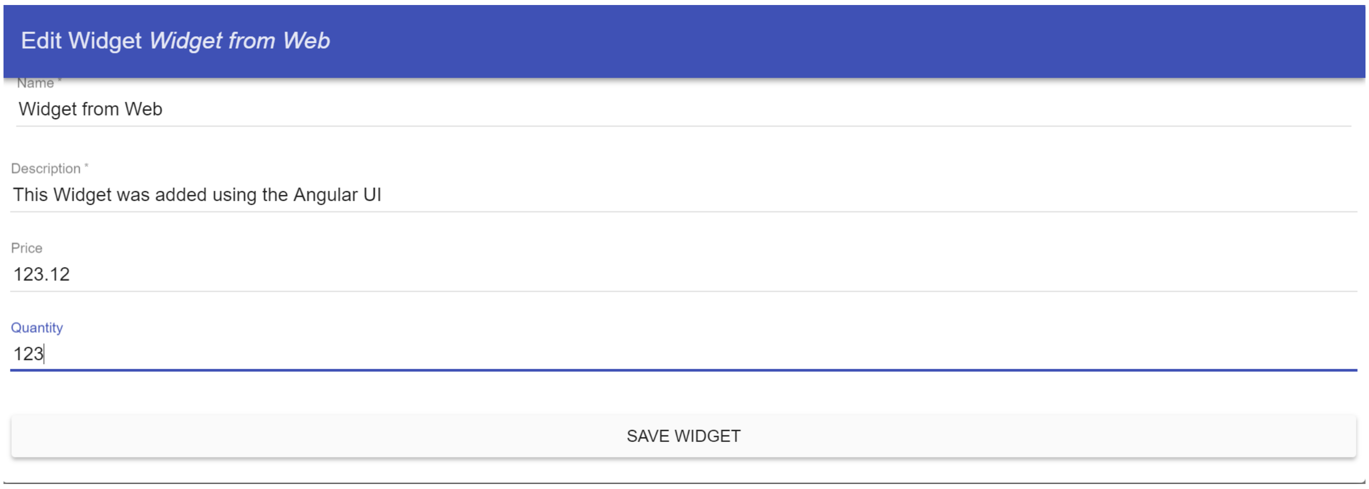
The PageController needs to have the $mdDialog service marked as a Dependancy, and injected into the Controller. Let's add that it and see if it makes a difference.



```
'$scope', '$timeout', '$http', '$log', '$mdDialog', function($q, $scope, $timeout, $http, $log,$mdDialog)
```



We now have a Dialog we can use to Add a New Widget. Let's fill in some details and Create a new Widget. Once "Save Widget" is clicked, the Widget should be added to the database



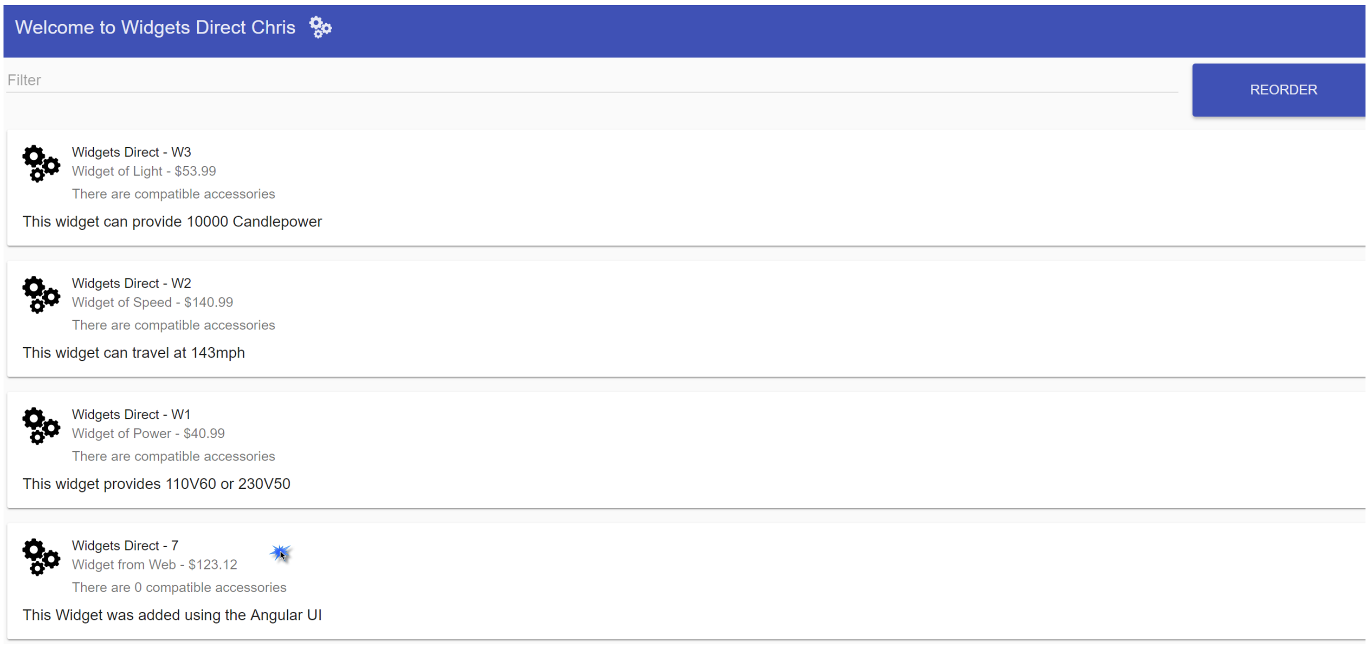
Name *
Widget from Web

Description *
This Widget was added using the Angular UI

Price
123.12

Quantity
123

Once we refresh Welcome.csp, we should see our new Widget displayed in the list. Success!


Welcome to Widgets Direct Chris

Filter

REORDER

Widgets Direct - W3
Widget of Light - $53.99
There are compatible accessories
This widget can provide 10000 Candlepower

Widgets Direct - W2
Widget of Speed - $140.99
There are compatible accessories
This widget can travel at 143mph

Widgets Direct - W1
Widget of Power - $40.99
There are compatible accessories
This widget provides 110V60 or 230V50

Widgets Direct - 7
Widget from Web - $123.12
There are 0 compatible accessories
This Widget was added using the Angular UI

Today we:

1. implemented a "Add New Widget" form
2. debugged issues caused by functions being on inaccessible controllers, and controllers with missing dependencies

Next time we will:

- Implement a Delete button to round off our CRUD operations
- Implement an autorefresh of the page when Widgets are added or edited
- explore different ways of viewing the data held in our controller

This article is part of a multi-part series on using Angular on top of Caché REST services. The listing of the full series can be found at the Start Here *page*

#Angular #JSON #REST API #Frontend #SOAP #Caché

Source URL:https://community.intersystems.com/post/lets-write-angular-1x-app-cach%C3%A9-rest-backend-part-13