

# Exercise (Instructions): Ionic Lists: Advanced Features Part 2

## Objectives and Outcomes

In this exercise, you will continue exploring the advanced features of Ionic lists. In particular you will learn about using the delete feature, and also implement a custom Angular filter. At the end of this exercise, you will be able to:

- Use the delete button feature of Ionic lists
- Design and implement a custom Angular filter

## Updating favoriteFactory

- Open *services.js* and update favoriteFactory by add two more functions as follows:

```
favFac.deleteFromFavorites = function (index) {
    for (var i = 0; i < favorites.length; i++) {
        if (favorites[i].id == index) {
            favorites.splice(i, 1);
        }
    }
}

favFac.getFavorites = function () {
    return favorites;
};
```

## Creating favorites.html Template

- In the templates folder, create a new file named favorites.html, and add the following code into the file:

```
<ion-view view-title="My Favorites">
  <ion-nav-buttons side="secondary">
    <div class="buttons">
      <button class="button button-icon icon ion-ios-minus-outline"
        ng-click="toggleDelete()"></button>
    </div>
```

```

    </ion-nav-buttons>
<ion-content>
  <ion-list show-delete="shouldShowDelete">
    <ion-item ng-repeat="dish in dishes | favoriteFilter:favorites" href="#/app/menu/{{
dish.id}}" class="item-thumbnail-left" on-swipe-left="deleteFavorite(dish.id)">
      
      <h2>{{dish.name}}
        <span style="font-size:75%">{{dish.price | currency}}</span>
        <span class="badge badge-assertive">{{dish.label}}</span></h2>
      <p>{{dish.description}}</p>
      <ion-delete-button class="ion-minus-circled"
        ng-click="deleteFavorite(dish.id)">
      </ion-delete-button>
    </ion-item>
  </ion-list>
</ion-content>
</ion-view>

```

## Updating app.js

- Open app.js, and add the new state into the config there as follows:

```

.state('app.favorites', {
  url: '/favorites',
  views: {
    'mainContent': {
      templateUrl: 'templates/favorites.html',
      controller: 'FavoritesController'
    }
  }
})

```

## Adding FavoritesController

- Open controllers.js and add FavoritesController as follows:

```

.controller('FavoritesController', ['$scope', 'menuFactory', 'favoriteFactory', 'baseURL'
, '$ionicListDelegate', function ($scope, menuFactory, favoriteFactory, baseURL, $ionicLi
stDelegate) {

    $scope.baseURL = baseURL;
    $scope.shouldShowDelete = false;

    $scope.favorites = favoriteFactory.getFavorites();

    $scope.dishes = menuFactory.getDishes().query(
        function (response) {
            $scope.dishes = response;
        },
        function (response) {
            $scope.message = "Error: " + response.status + " " + response.statusText;
        });
    console.log($scope.dishes, $scope.favorites);

    $scope.toggleDelete = function () {
        $scope.shouldShowDelete = !$scope.shouldShowDelete;
        console.log($scope.shouldShowDelete);
    }

    $scope.deleteFavorite = function (index) {

        favoriteFactory.deleteFromFavorites(index);
        $scope.shouldShowDelete = false;

    }

}}])

```

## Implementing favoriteFilter

- Open controllers.js and add the filter as shown below:

```

.filter('favoriteFilter', function () {
    return function (dishes, favorites) {
        var out = [];
        for (var i = 0; i < favorites.length; i++) {

```

```
        for (var j = 0; j < dishes.length; j++) {
            if (dishes[j].id === favorites[i].id)
                out.push(dishes[j]);
        }
    }
    return out;
});
```

## Updating sidebar.html

- Add the following new item into the list in sidebar.html to add the "My Favorites" option there:

```
<ion-item menu-close href="#/app/favorites">
  My Favorites
</ion-item>
```

- Save all the changes and examine the updated app.

## Conclusions

In this exercise, you explored the use of the delete button and also learnt about implementing a custom Angular filter.