Exercise (Instructions): Using Local Storage

Objectives and Outcomes

In this exercise, you will learn about HTML5 local storage and how you can leverage this to support the storing of information in your app in local storage. At the end of this exercise, you will be able to:

- Design a new Angular service using local storage
- Make use of the Local Storage to remember user information

Adding $localStorage Service

- Open *services.js* and add the following $localStorage service code to it:

```
.factory('$localStorage', ['$window', function($window) {
  return {
    store: function(key, value) {
      $window.localStorage[key] = value;
    },
    get: function(key, defaultValue) {
      return $window.localStorage[key] || defaultValue;
    },
    storeObject: function(key, value) {
      $window.localStorage[key] = JSON.stringify(value);
    },
    getObject: function(key,defaultValue) {
      return JSON.parse($window.localStorage[key] || defaultValue);
    }
  }
}])


```

Updating AppCtrl

- Open *controllers.js* and update the AppCtrl controller as follows to inject $localStorage:

```
.controller('AppCtrl', function ($scope, $ionicModal, $timeout, $localStorage) {


```

- Then update the initialization of loginData JavaScript object as follows:

```
    $scope.loginData = $localStorage.getObject('userinfo','{}');


```

- Finally, within the doLogin() function in AppCtrl, update it as follows:

```
1       $scope.doLogin = function () {
2           console.log('Doing login', $scope.loginData);
3           $localStorage.storeObject('userinfo',$scope.loginData);
4
5           . . .
6
7        };
```

- Save the changes and check the updated application.

Conclusions

In this exercise, you learnt to make use of the HTML5 local storage within your application, first by creating a service to wrap the local storage, and then inject into a controller in order to use the storage.

✓ Complete