**UI Route**

Below step we need to take care while implementing UI Routering mechanism.

1. Add angular-ui-router.js reference file into main page.
2. We need to inject ‘ui.router’ into module . like below code line.

var myapp = angular.module('myapp', ["ui.router"]).

1. Use two services(($stateProvider, $urlRouterProvider) into config function like below code

myapp.config(function($stateProvider, $urlRouterProvider){

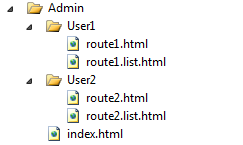
1. Use ui-sref , where user click event fire just like href.
2. Use ui-view, because ui-view is the place where you insert your view.

**State Manager**

The new $stateProvider works similar to Angular's v1 router, but it focuses purely on state.

* A state corresponds to a "place" in the application in terms of the overall UI and navigation.
* A state describes (via the controller / template / view properties) what the UI looks like and does at that place.
* States often have things in common, and the primary way of factoring out these commonalities in this model is via the state hierarchy, i.e. parent/child states aka nested states.

**Example:-**



Index.html

<!DOCTYPE html>

<html ng-app="myapp">

<head>

<title>AngularJS: UI-Router Quick Start</title>

<!-- Bootstrap CSS -->

<link href="//cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/2.3.1/css/bootstrap.min.css" rel="stylesheet">

</head>

<body class="container">

<p><i>Best viewed in pop-out mode to see location changes. Click blue button on the right.</i></p>

<div class="navbar">

<div class="navbar-inner">

<a class="brand" href="#">Quick Start</a>

<ul class="nav">

<li><a ui-sref="route1">Route 1</a></li>

<li><a ui-sref="route2">Route 2</a></li>

</ul>

</div>

</div>

<div class="row">

<div class="span12">

<div class="well" ui-view></div>

</div>

</div>

<!-- Angular -->

<script src="//ajax.googleapis.com/ajax/libs/angularjs/1.2.4/angular.js"></script>

<!-- UI-Router -->

<script src="//angular-ui.github.io/ui-router/release/angular-ui-router.js"></script>

<!-- App Script -->

<script>

var myapp = angular.module('myapp', ["ui.router"])

myapp.config(function($stateProvider, $urlRouterProvider){

// For any unmatched url, send to /route1

$urlRouterProvider.otherwise("/User1/route1")

$stateProvider

.state('route1', {

url: "route1",

templateUrl: "User1/route1.html"

})

.state('route1.list', {

url: "/list",

templateUrl: "User1/route1.list.html",

controller: function($scope){

$scope.items = ["Lappy", "Mobile", "Desk", "Pen"];

}

})

.state('route2', {

url: "route2",

templateUrl: "User2/route2.html"

})

.state('route2.list', {

url: "/list",

templateUrl: "User2/route2.list.html",

controller: function($scope){

$scope.things = ["A", "Set", "Of", "Things"];

}

})

})

</script>

</body>

</html>



Route1.html

<h1>Route 1</h1>

<hr/>

<a ui-sref=".list">Show List</a>

<div ui-view></div>

Route1.list.html

<h3>List of Route 1 Items</h3>

<ul>

<li ng-repeat="item in items">{{item}}</li>

</ul>



Route2.html

<h1>Route 2</h1>

<hr/>

<a ui-sref=".list">Show List</a>

<div ui-view></div>

Route2.list.html

<h3>List of Route 2 Things</h3>

<ul>

<li ng-repeat="thing in things">{{thing}}</li>

</ul>