

UI-Router Directives

ui-sref="state.name({ stateParam: value })"

ui-sref-active="activeClass"

Applies the class if nearest **ui-sref** or a descendant state is active

ui-sref-active-eq="activeClass"

Applies the class if nearest **ui-sref** state is active

ui-view="viewName"

<ui-view **autoscroll**="condition">

Scroll into view on load

UI-Router Filters

string | **isState**

alias for **\$state.is**('stateName')

string | **includedByState**

alias for **\$state.includes**('stateName')

UI-Router Events

\$stateChangeSuccess

Fired after completion

\$stateChangeError

Fired on state resolution error **USEFUL FOR DEBUGGING**

\$stateChangeStart

\$stateNotFound

\$viewContentLoaded

\$viewContentLoaded

UI-Router Services

\$state

\$stateParams

contains current url params (tokens) as properties

\$stateProvider

config only

\$urlRouterProvider

config only

\$state Service

go(string, [params], [options])

alias for **transitionTo**()

transitionTo(string, [params], [options])

'**contact.detail**': go to 'contact.detail'. '^': go to parent. '^**.sibling**': go to sibling. '**.child.grandchild**': go to *current state's* [grand]child.

reload()

Force reload current state tree

includes(string, [params]) {boolean}

Specifies if passed state is currently active (ancestry)

is(string, [params]) {boolean}

Specifies if passed state is current (exact)

href(string, [params], [options]) {string}

get([string]) {object/array}

Returns specified state or all states

current {object}

The current state object

List of methods and properties of the \$state service

State Definition Object

url {string}

Url's are appended to parent states. Create tokens using **"/:token"** or using RegEx **"/{token:[a-zA-Z0-9]}"**

template {string/function}

String containing template HTML or injectable function that returns the string value. (ignored if using **views** option)

templateUrl {string/function}

String containing path to HTML file or injectable function that returns the string value. (ignored if using **views** option)

controller {string/function}

String name of controller or the controller function (ignored if using **views** option)

State Definition Object (cont)

abstract {boolean[false]}

If you can **\$state.go()** to this view directly

onEnter {function}

Resolved dependency injectable function. Called when entering the state.

onExit {function}

Resolved dependency injectable function. Called when exiting the state.

reloadOnSearch {boolean[true]}

if false will not retrigger the same state when the search parameters change

params {array}

parameter names or regular expressions *when* the state is activated

views {object}

controller and **templates** for specific views

data {object}

stores static configuration data

resolve {object}

Creates injectable resources: { resource: function() { return promise | data; } }

\$stateProvider.state(stateName, config)