# **Module Configuration**

### **Configure your states in your module's config method.**

**//首先要确保引入了ui.router模块**

Be sure to include ui.router as a module dependency.

angular.module("myApp", ["ui.router"]).config(function($stateProvider){

$stateProvider.state(stateName, stateConfig);

})

# **$stateProvider**

## **$stateProvider.state(stateName, stateConfig)**

Creates a new application state. For alternate usage, see [Object-based States](https://github.com/angular-ui/ui-router/wiki/Nested-States-&-Nested-Views" \l "object-based-states)

The parameters for .state() are:

## **stateName**

****String****

A unique state name, e.g. "home", "about", "contacts". To create a parent/child state use a dot, e.g. "about.sales", "home.newest".

// The state() method takes a unique stateName (String) and a stateConfig **$stateProvider.state(stateName, stateConfig);**

// stateName can be a single top-level name (must be unique).

$stateProvider.state("home", {});

// Or it can be a nested state name. This state is a child of the above "home" state.

$stateProvider.state("home.newest", {});

// Nest states as deeply as needed.

$stateProvider.state("home.newest.abc.xyz.inception", {});

// state() returns $stateProvider, so you can chain state declarations.

$stateProvider.state("home", {}).state("about", {}).state("contacts", {});

## **stateConfig**

****Object****

The stateConfig object has the following acceptable properties.

### **template, templateUrl, templateProvider**

Three ways to set up your templates. Only use one per state (or view, see below)!

template String HTML content, or function that returns an HTML string

templateUrl String URL path to template file OR Function, returns URL path string

templateProvider Function, returns HTML content string

### **controller, controllerProvider**

A controller paired to the state

controller Function OR name as String

controllerProvider Function (injectable), returns the actual controller function or string.

### **resolve**

A map of dependencies which should be injected into the controller

resolve Object

* keys - name of dependency to be injected into controller
* factory - {string|function} If string then it is alias for service. Otherwise if function, it is injected and return value it treated as dependency. If result is a promise, it is resolved before its value is injected into controller

### **url**

A url with optional parameters. When a state is navigated or [transitioned](https://github.com/angular-ui/ui-router/wiki/" \l "wiki-transitiontoto-toparams-updatelocation) to, the [$stateParams](https://github.com/angular-ui/ui-router/wiki/" \l "wiki-stateparams)service will be populated with any parameters that were passed.

url String

### **params**

A map which optionally configures parameters declared in the url, or defines additional non-url parameters. Only use this within a state if you are not using url. Otherwise you can specify your parameters within the url. When a state is navigated or [transitioned](https://github.com/angular-ui/ui-router/wiki/" \l "wiki-transitiontoto-toparams-updatelocation) to, the [$stateParams](https://github.com/angular-ui/ui-router/wiki/" \l "wiki-stateparams) service will be populated with any parameters that were passed.

params Object

### **views (nested views is more useful)**

Use the views property to set up multiple views. If you don't need multiple views within a single state this property is not needed. Tip: remember that often nested views are more useful and powerful than multiple sibling views.

views Object

* keys - {string} name of ui-view
* view config - {object} view configuration object can set up its own [templates] and ([https://github.com/angular-ui/ui-router/wiki#the-simplest-form-of-state](https://github.com/angular-ui/ui-router/wiki" \l "the-simplest-form-of-state)) [controllers].

[Learn more about multiple named views](https://github.com/angular-ui/ui-router/wiki/Multiple-Named-Views)

### **abstract**

An abstract state will never be directly activated, but can provide inherited properties to its common children states.

abstract Boolean - (default is false)

[Learn more about abstract states](https://github.com/angular-ui/ui-router/wiki/Nested-States-&-Nested-Views" \l "abstract-states)

### **onEnter, onExit**

Callback functions for when a state is entered and exited. Good way to trigger an action or dispatch an event, such as opening a [dialog](https://github.com/angular-ui/bootstrap/tree/master/src/dialog" \l "dialogprovider-service-in-uibootstrap).

* onEnter Function, injected including resolves
* onExit Function, injected including resolves

[Learn more about state callbacks](https://github.com/angular-ui/ui-router/wiki" \l "onenter-and-onexit-callbacks)

### **reloadOnSearch v0.2.5**

Boolean (default true). If false will not retrigger the same state just because a search/query parameter has changed. Useful for when you'd like to modify $location.search() without triggering a reload.

### **data**

Arbitrary data object, useful for custom configuration.

data Object