

Package Control

BROWSE

AngularJS

by [angular-ui](#) **ST2/ST3**  TOP 100

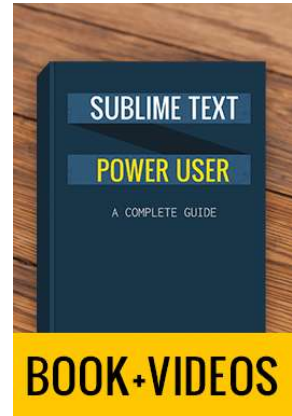
AngularJS code completion, snippets, go to definition, quick panel search, and more.

LABELS [auto-complete](#), [code navigation](#), [snippets](#)

SUPPORTED VIA 

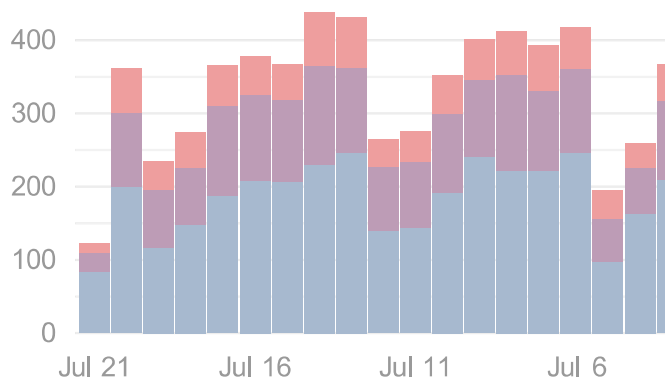
Details

VERSION 1.2.1
HOMEPAGE [github.com](#)
ISSUES [github.com](#)
MODIFIED 12 months ago
LAST SEEN 16 minutes ago
FIRST SEEN 3 years ago



Installs

TOTAL 150K WIN 69K OS X 54K
LINUX 27K



Readme

SOURCE [raw.githubusercontent.com](#)

Installation

Browse

Search

Docs

News

Stats

About

Say Thanks

AngularJS Sublime Text Package

Installation Options

- Package Control
- Download this repo, rename it to 'AngularJS', and place it within your **Packages** folder. This can be found within Sublime Text at **Preferences > Browse Packages...**
- Clone the repo into your **Packages** folder `git clone git://github.com/angular-ui/AngularJS-sublime-package.git` AngularJS

Recommended Settings

Update your User Settings to the following. This setting update will automatically open the completion list for HTML attributes. You can add scopes for other preprocessor to get the list to automatically show.

(this is currently a ST3 default)

```
// Controls what scopes auto complete will b
"auto_complete_selector": "source - comment,
```



(this is currently a ST3 default, sans 'text.haml')

```
// For haml you could add
"auto_complete_selector": "source - comment,
```



```
// add for auto triggering controller comple
"auto_complete_triggers":
```

```
[
  {
    "characters": "ng-controller=\"*"
    "selector": "punctuation.definit
  }
]
```



Keymaps

super+ctrl+l

If not indexed: Indexes current project; If indexed: Opens quick panel with a list of definitions to search through [command: angularjs_find]

super+ctrl+alt+l

Attempts to goto definition (project must be indexed first)
[command: angularjs_go_to_definition]

super+shift+ctrl+l

Attempts to open browser to directive documentation at current cursor location [command: angularjs_go_to_documentation]

notice

The above keymaps could be used by other plugins as well, so, you may need to remap them via

Preferences > Key Bindings - User

To check if another plugin is using the above keymaps all you have to do is open the ST console (ctrl+) and write out **sublime.log_commands(True)** and that will log all the commands ran in ST.

Command Palette

- AngularJS: Rebuild Search Index
- AngularJS: Delete Cache
- AngularJS: Prune Cache (removes missing files from index)
- AngularJS: Look Up Definition
- AngularJS: Toggle - Disable All Completions
- AngularJS: Toggle - Disable Indexed Directive Completions
- AngularJS: Toggle - Disable Indexed Isolate Completions
- AngularJS: Toggle - Disable Built-in Directive Completions
- AngularJS: Toggle - Disable Built-in Element Completions
- AngularJS: Toggle - Disable JS Completions

- AngularJS: Toggle - Enable data- Prefix

Completion Options

Preferences > Package Settings > AngularJS > **Completions**
- **User**

You can use the following properties to either extend or override the default completions provided.

extended_attribute_list: []; Allows you to extend the plug-in with more attributes

angular_elements: [*]; Default list of directives that can be used as HTML elements

filter_list: [*]; Default list of filters

core_attribute_list: [*]; Default list of filters

[*] - Adding any of these properties to your User file will override all default values for that setting

Example *Completions - User*

```
{
    "extended_attribute_list": [
        ["my-directive\tMy Directives", "my-"]
    ]
}
```



Checkout the default completions settings (*Preferences > Package Settings > AngularJS > Completions - Default*) to see more examples.

Completion Settings

Preferences > Package Settings > AngularJS > **Settings - User**

js_scope: "source.js - string.quoted - comment - meta.brace.square"; Scope to return JS completions in

filter_scope: "text.html string.quoted"; Scope to return filters in

js_prefixes: [“, ”]; add characters that you want to prevent completion triggers

disable_all_completions: false,

disable_indexed_directive_completions: false; *bare-bones completion of any directives that have been index*

disable_indexed_isolate_completions: false; *attempts to provide isolate scope completions when a directive is used as an element*

disable_default_directive_completions: false;

disable_default_element_completions: false;

disable_default_js_completions: false;

enable_data_prefix: bool (false); *adds the 'data-' prefix to attribute completions, note that you must still type 'ng-' to get autocomplete-list*

Example *Settings* - User, enable “data-” prefix

```
{
    "enable_data_prefix": true
}
```

Indexing Options

Preferences > Package Settings > AngularJS > **Settings - User** (*requires project to be re-indexed*)

The regex that's used for look up expects the definitions to start like one of the the following examples:

```
filter('interpolate', ['version', function(v
    .filter('interpolate', ['version', function(
    ('chained')).filter('interpolate', ['version'

/**
 * If you use something other than `app` as
```

Navigation icons: back, forward, search, etc.

```
match_expression: "((^[ \\\\t]*\\.\\{0}|^[ \\\\t]*{0}|angular\\.\\{0}|\\.\\.\\{0}|app\\.\\{0})[ ]*\\([ ]*[\"\\'\\'\\'])([\\w\\.\\$]*)([\"\\'\\'\\'])"
```

◀ 111 ▶

You can override the default project folders by setting the AngularJS **folders** variable within your projects settings

Project > Edit Project

Example:

```
{
  "folders":
  [
    {
      "follow_symlinks": true,
      "path": "/Users/username/Project
    }
  ],
  "settings": {
    "AngularJS": {
      "folders": [
        "/Users/username/Projects/ex
        "/Users/username/Vendors/som
      ]
    }
  }
}
```



You can use 'shell-like' wildcards within your folder paths, they're expanded via the [glob](#) module.

Quick Panel Options

Preferences > Package Settings > AngularJS > **Settings - User** (*Sublime Text 3 Only*)

show_file_preview: bool(true); As you type, the current file and definition will be shown in the buffer

Example *Settings - User*, hide file preview

```
{
  "show_file_preview": false
}
```

Plug-in Details

Syntax File

Provides a syntax file, HTML (Angular.js), that you can set to your view which adds the HTML scope back to `<script type="text/ng-template">` tags.

```

1 <!-- angular template -->
2 <script type="text/ng-template">
3 <!-- smet completion: div>span(items)+3 -->
4 <div>
5 <span>items</span>
6 <span>items</span>
7 <span>items</span>
8 </div>
9
10 <?php
11 $test = 'asdf';
12 public function test($value='')
13 {
14     = code...
15 }
16 >
17 var test = 'ing'
18 </script>
19
20 <!-- JS script -->
21 <script>
22 var test = 'ing';
23 </script>
24
25 <!-- Preserves PHP -->
26 <?php
27 $test = 'asdf';
28 public function test($value='')
29 {
30     = code...
31 }
32 >
33
34 <!-- Normal HTML -->
35 <div class="test" ng-class="test">
36 testing
37 </div>
  
```

Completions

Provides auto-completion of core AngularJS directives, such as `ng-repeat`, `ng-click`, as well as any custom directives you create.

```

4 <div ng-repeat="item in someItems">
5 <div>
6 </div>
  
```

Autocomplete suggestions for `ng` directives:

- check
- class
- click
- contenteditable
- custom-directive-two
- custom-directive-one
- ngChecked
- ngClass
- ngClick
- ngAttr
- ngIndexed
- ngIndexed
- ngIndexed

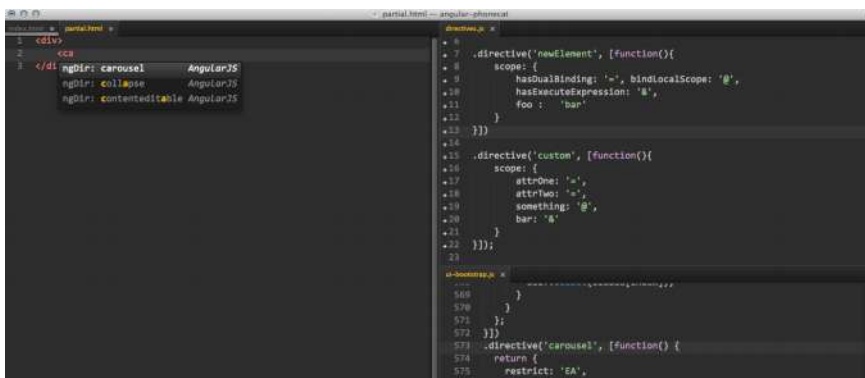
```

4
5
6 angular.module('myApp.directives', []).
7   directive('appVersion', ['version', function(version) {
8     return function(scope, elm, attrs) {
9       elm.text(version);
10     };
11   }])
12   .directive('customDirectiveOne', [])
13   .directive('customDirectiveTwo', []);
  
```

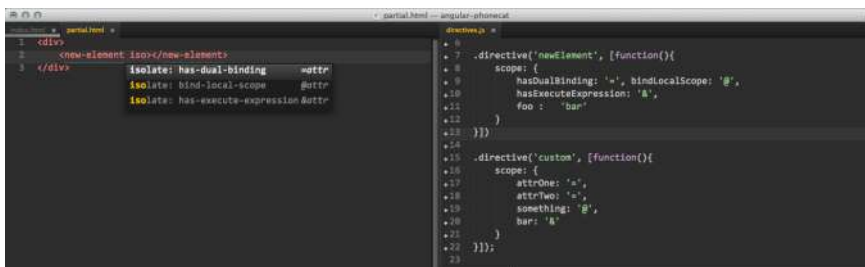
The following completions require you to index your project

(Index your project via `super+ctrl+l`)

Provides auto-completions for any directive as an HTML element, prefixed with `ngDir` to easily find via fuzzy search.



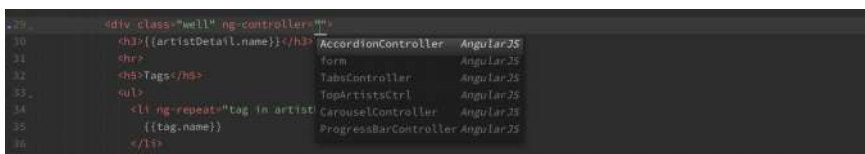
Provides **isolate** scope completions (with binding types hinted on the right) specific to any directive used as an element, prefixed with **isolate** to easily find via fuzzy search.



You can also get completions for filters within HTML markup. Trigger the completions list via **ctrl+space** right after typing `|` (pipe plus a space) and you should find all your filters that have been indexed within the completion list.



Provides **controller** completions when the cursor is within the double quotes of `ng-controller=""`. You can have this triggered automatically via the settings provided in the 'Recommending Settings' section

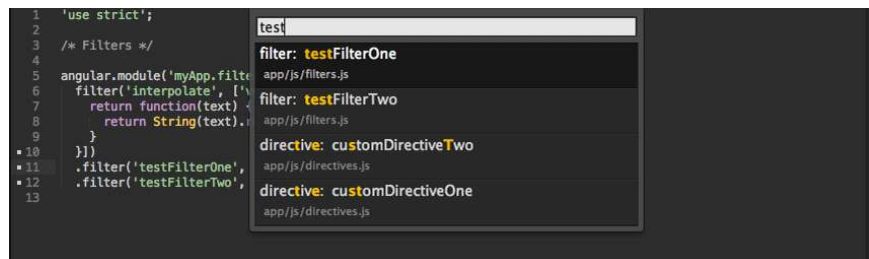


Goto Definition

Once you have your project indexed you can use the keyboard shortcut **super+alt+ctrl+1** when your cursor is within directive/services/etc.. and you'll be transported to the file that contains the definition.

Quick Panel Definition Look Ups

Quickly find your directives/filters/modules/factories via the quick_panel. Once your project has been indexed, by either executing the command 'AngularJS: Rebuild Search Index' from the command palette or executing the shortcut **super+ctrl+1** to open the quick_panel search for the first time.



Each time you save a file that file will be reindexed, if you have already triggered indexing, so that the quick_panel search stays up-to-date.

Javascript Completions

Global Context

- angular - **angular**
- \$animate - **\$animate**
- \$animateProvider - **\$animateProvider**
- \$cacheFactory - **\$cacheFactory(cacheId[, options])**
- \$compile - **\$compile(element, transclude, maxPriority)**
- \$compileProvider - **\$compileProvider**
- \$controller - **\$controller(constructor, locals)**
- \$controllerProvider - **\$controllerProvider**
- \$exceptionHandler - **\$exceptionHandler(exception[, cause])**
- \$exceptionHandlerProvider - **\$exceptionHandlerProvider**
- \$filter - **\$filter(name)**
- \$filterProvider - **\$filterProvider**
- \$http - **\$http**
- \$httpBackend - **\$httpBackend**
- \$injector - **\$injector**
- \$interpolate - **\$interpolate(text[, mustHaveExpression, trustedContext])**
- \$interpolateProvider - **\$interpolateProvider**
- \$interval - **\$interval**
- \$locale - **\$locale**

- `$location` - `$location`
- `$locationProvider` - `$locationProvider`
- `$log` - `$log`
- `$logProvider` - `$logProvider`
- `$parse` - `$parse(expression)`
- `$parseProvider` - `$parseProvider`
- `$provide` - `$provide`
- `$q` - `$q`
- `$rootElement` - `$rootElement`
- `$rootScope` - `$rootScope`
- `$rootScopeProvider` - `$rootScopeProvider`
- `$sce` - `$sce`
- `$sceDelegate` - `$sceDelegate`
- `$sceDelegateProvider` - `$sceDelegateProvider`
- `$sceProvider` - `$sceProvider`
- `$scope` - `$scope`
- `$templateCache` - `$templateCache`
- `$timeout` - `$timeout`
- `$window` - `$window`
- `$cookies` - `$cookies`
- `$cookieStore` - `$cookieStore`
- `$resource` - `$resource(url[, paramDefaults, actions])`
- `$route` - `$route`
- `$routeParams` - `$routeParams`
- `$routeProvider` - `$routeProvider`
- `$sanitize` - `$sanitize(html)`
- `$swipe` - `$swipe`

Context Specific

- **angular**
 - `bind`
 - `bootstrap`
 - `copy`
 - `element`
 - `equals`
 - `extend`
 - `forEach`
 - `fromJson`
 - `identity`
 - `injector`
 - `isArray`
 - `isDate`

- isDefined
- isElement
- isFunction
- isNumber
- isObject
- isString
- isUndefined
- lowercase
- mock
- module
- noop
- toJson
- uppercase
- version
- **\$animate**
 - addClass
 - enter
 - leave
 - move
 - removeClass
- **\$animateProvider**
 - classNameFilter
 - register
- **\$compileProvider**
 - aHrefSanitizationWhitelist
 - directive
 - imgSrcSanitizationWhitelist
- **\$controllerProvider**
 - register
- **\$exceptionHandlerProvider**
 - mode
- **\$filterProvider**
 - register
- **\$http**
 - delete
 - get
 - head
 - jsonp
 - post
 - put
 - defaults

- pendingRequests

- **\$httpBackend**

- expect
- expectDELETE
- expectGET
- expectHEAD
- expectJSONP
- expectPATCH
- expectPOST
- expectPUT
- flush
- resetExpectations
- verifyNoOutstandingExpectation
- verifyNoOutstandingRequest
- when
- whenDELETE
- whenGET
- whenHEAD
- whenJSONP
- whenPOST
- whenPUT

- **\$injector**

- annotate
- get
- has
- instantiate
- invoke

- **\$interpolateProvider**

- endSymbol
- startSymbol

- **\$interval**

- cancel
- flush

- **\$locale**

- id

- **\$location**

- absUrl
- hash
- host
- path
- port
- protocol

- replace
 - search
 - url
- **\$locationProvider**
 - hashPrefix
 - html5Mode
- **\$log**
 - debug
 - error
 - info
 - log
 - warn
 - assertEmpty
 - reset
- **\$logProvider**
 - debugEnabled
- **\$parseProvider**
 - logPromiseWarnings
 - unwrapPromises
- **\$provide**
 - constant
 - decorator
 - factory
 - provider
 - service
 - value
- **\$q**
 - all
 - defer
 - reject
 - when
- **\$rootScope, \$scope**
 - \$apply
 - \$broadcast
 - \$destroy
 - \$digest
 - \$emit
 - \$eval
 - \$evalAsync
 - \$new
 - \$on

- \$parent
- \$root
- \$watch
- \$watchCollection
- \$id
- **\$rootScopeProvider**
 - digestTtl
- **\$sce**
 - getTrusted
 - getTrustedCss
 - getTrustedHtml
 - getTrustedJs
 - getTrustedResourceUrl
 - getTrustedUrl
 - parse
 - parseAsCss
 - parseAsHtml
 - parseAsJs
 - parseAsResourceUrl
 - parseAsUrl
 - trustAs
 - trustAsHtml
 - trustAsJs
 - trustAsResourceUrl
 - trustAsUrl
 - isEnabled
- **\$sceDelegate**
 - getTrusted
 - trustAs
 - valueOf
- **\$sceDelegateProvider**
 - resourceUrlBlacklist
 - resourceUrlWhitelist
- **\$sceProvider**
 - enabled
- **\$timeout**
 - cancel
 - flush
- **\$cookieStore**
 - get
 - put

- remove
- **\$route**
 - reload
 - current
 - routes
- **\$routeProvider**
 - otherwise
 - when
- **\$swipe**
 - bind
- **mock**
 - dump
 - module
- **events**
 - \$locationChangeStart
 - \$locationChangeSuccess
 - \$destroy
 - \$includeContentLoaded
 - \$includeContentRequested
 - \$routeChangeError
 - \$routeChangeStart
 - \$routeChangeSuccess
 - \$routeUpdate
 - \$viewContentLoaded
- **attrs**
 - \$addClass
 - \$observe
 - \$removeClass
 - \$set
 - \$updateClass
 - \$attr

verbose

verbose_\$http

```
$http('GET|POST|PUT|DELETE', url, post, func
  // success
}, function(status, response){
  // error
});
```




verbose_\$filter

```
$filter('currency|date|filter|json|limitTo|l
```



verbose_\$interval

```
$interval(fn, delay, count, invokeApply)
```

verbose_\$timeout

```
$timeout(function() {
```

```
}, delay);
```

verbose_directive

```
directive('', [''], function() {
  // Runs during compile
  return {
    // name: '',
    // priority: 1,
    // terminal: true,
    // scope: {}, // {} = isolate, true = ch
    // controller: function($scope, $element
    // require: 'ngModel', // Array = multip
    // restrict: 'A', // E = Element, A = At
    // template: '',
    // templateUrl: '',
    // replace: true,
    // transclude: true,
    // compile: function(tElement, tAttrs, f
    link: function($scope, iElm, iAttrs, con
```

```
  }
};
}]);
```



verbose_module

```
/**
 * Module
 *
```

```
* Description  
*/  
angular.module('', []).
```

providers

config

```
config(['', function() {  
  
}])
```

constant

```
constant('name', value)
```

controller

```
controller('name', ['', function() {  
  
}])
```

factory

```
factory('name', ['', function() {  
    return function name() {  
  
    };  
}])
```

run

```
run('name', ['', function() {  
  
}])
```

service

```
service('name', ['', function() {  
  
}])
```

value

```
value('name', value)
```

© 2015