

PROJECTS

Ember

Ember Data

TAG: V2.11.0

MODULES

ember

ember-application

ember-debug

ember-extension-support

ember-glimmer

ember-htmlbars

ember-metal

ember-routing

ember-runtime

ember-testing

ember-views

NAMESPACES

Ember

Ember.AutoLocation

Ember.FEATURES

Ember.Instrumentation

Ember.Location

Ember.String

Ember.computed

Ember.inject

Ember.run

CLASSES

Backburner

BucketCache

Container

ContainerProxyMixin

Descriptor

Ember.ActionHandler

Ember.ActionSupport

Ember.Application

Ember.ApplicationInstance

Ember.ApplicationInstance.BootO...

Ember.Array

Ember.ArrayProxy

EMBER.COMPUTED NAMESPACE



PUBLIC

DEFINED IN: `packages/ember-metal/lib/computed.js:450`**MODULE:** `ember-metal`

This helper returns a new property descriptor that wraps the passed computed property function. You can use this helper to define properties with mixins or via `Ember.defineProperty()`.

If you pass a function as an argument, it will be used as a getter. A computed property defined in this way might look like this:

```
1 let Person = Ember.Object.extend({
2   init() {
3     this._super(...arguments);
4
5     this.firstName = 'Betty';
6     this.lastName = 'Jones';
7   },
8
9   fullName: Ember.computed('firstName', 'lastName', function() {
10    return `${this.get('firstName')} ${this.get('lastName')}`;
11  })
12 });
13
14 let client = Person.create();
15
16 client.get('fullName'); // 'Betty Jones'
17
18 client.set('lastName', 'Fuller');
19 client.get('fullName'); // 'Betty Fuller'
```

You can pass a hash with two functions, `get` and `set`, as an argument to provide both a getter and setter:

```
1 let Person = Ember.Object.extend({
2   init() {
3     this._super(...arguments);
4
5     this.firstName = 'Betty';
6     this.lastName = 'Jones';
7   },
8
9   fullName: Ember.computed('firstName', 'lastName', {
10    get(key) {
11      return `${this.get('firstName')} ${this.get('lastName')}`;
12    },
13    set(key, value) {
14      let [firstName, lastName] = value.split(/\s+/);
15      this.setProperties({ firstName, lastName });
16      return value;
17    }
18  });
19 });
20
21 let client = Person.create();
22 client.get('firstName'); // 'Betty'
23
24 client.set('fullName', 'Carroll Fuller');
25 client.get('firstName'); // 'Carroll'
```

- Ember.Binding
- Ember.Checkbox
- Ember.ClassNamesSupport
- Ember.Comparable
- Ember.Component
- Ember.ComputedProperty
- Ember.ContainerDebugAdapter
- Ember.Controller
- Ember.ControllerMixin
- Ember.Copyable
- Ember.CoreObject
- Ember.CoreView
- Ember.DataAdapter
- Ember.Debug
- Ember.DefaultResolver
- Ember.EachProxy
- Ember.Engine
- Ember.EngineInstance
- Ember.Enumerable
- Ember.Error
- Ember.EventDispatcher
- Ember.Evented
- Ember.Freezable
- Ember.HTMLBars
- Ember.HashLocation
- Ember.Helper
- Ember.HistoryLocation
- Ember.InjectedProperty
- Ember.LinkComponent
- Ember.Map
- Ember.MapWithDefault
- Ember.Mixin
- Ember.MutableArray
- Ember.MutableEnumerable
- Ember.Namespace
- Ember.NativeArray
- Ember.NoneLocation
- Ember.Object
- Ember.ObjectProxy
- Ember.Observable
- Ember.OrderedSet
- Ember.PromiseProxyMixin
- Ember.ProxyMixin
- Ember.Route
- Ember.Router
- Ember.Service
- Ember.TargetActionSupport
- Ember.Templates.helpers
- Ember.Test

The `set` function should accept two parameters, `key` and `value`. The value returned from `set` will be the new value of the property.

*Note: This is the preferred way to define computed properties when writing third-party libraries that depend on or use Ember, since there is no guarantee that the user will have **prototype Extensions** enabled.*

The alternative syntax, with prototype extensions, might look like:

```
1  fullName: function() {
2    return this.get('firstName') + ' ' + this.get('lastName');
3  }.property('firstName', 'lastName')
```

Index	Methods
Show: <input checked="" type="checkbox"/> Inherited <input type="checkbox"/> Protected <input type="checkbox"/> Deprecated	
METHODS	
alias	match
and	max
bool	min
collect	none
deprecatingAlias	not
empty	notEmpty
equal	oneWay
filter	or
filterBy	readOnly
gt	reads
gte	setDiff
intersect	sort
lt	sum
lte	union
map	uniq
mapBy	uniqBy

- Ember.Test.Adapter
- Ember.Test.QUnitAdapter
- Ember.TextArea
- Ember.TextField
- Ember.TextSupport
- Ember.View
- Ember.ViewMixin
- Function
- Libraries
- Registry
- RegistryProxyMixin
- RoutingService
- String

COPYRIGHT 2017 TILDE INC.

TEAM | SPONSORS

SECURITY | LEGAL | LOGOS

COMMUNITY GUIDELINES

Ember.js is free, open source and always will be.

