

Basics

App.Router.map(fn)

allows you to add routes and resources to your app

App.advanceReadiness()

call this function when your app is ready to be initialized

App.deferReadiness()

delays initialization until advanceReadiness is called

App.inject(type, property, injection)

add a property onto every object of a specific type

Ember.ArrayController.extend({})

represents several objects

Ember.Controller.extend({})

grouping of specific, non object related functionality (e.g authentication or search)

Ember.Object.create({})

create an instance of an object

Ember.Object.createWithMixins(mixins, {})

create an instance of an object with mixins

Ember.Object.destroy()

set isDestroying to true and schedule removal of all bindings and observing for the end of run loop

Ember.Object.get(keyName)

return value of property by given name while respecting computed and observed properties

Ember.Object.reopenClass({})

add methods and properties to a class

Ember.Object.set(keyName, value)

set the value of property while respecting computed properties, unknown properties, property observers and chaining

Ember.ObjectController.extend({})

represents a single object

Basics (cont)

Ember.isNone(obj)

Returns true if the passed value is null or undefined.

Ember.Application.initializer

after: 'someInitializer'

name of the initializer to run before running this initializer

name: 'preload'

name for this initializer

initialize: function(container, application)

function to execute when an app is initializing

Ember.Application

Ember.Application.create: ({})

creates an instance that will be your app and your app's namespace

LOG_ACTIVE_GENERATION: true

activate logging of automatically generated routes and controllers

LOG_STACKTRACE_ON_DEPRECATION: true

activate logging of deprecated method or property usage

LOG_TRANSITIONS: true

activate basic logging of successful transitions

LOG_TRANSITIONS_INTERNAL: true

activate detailed logging of all routing steps

LOG_VIEW_LOOKUPS: true

activate logging of results of view and template searches by routes

rootElement

DOM element or jQuery-compatible selector string where your app will be rendered

});

Ember.Route

Ember.Route.extend: (mixins, {

beforeModel: function(transition)

hook executed before resolving models (use for early redirection)

activate: function

hooked called when router enters route the first time

afterModel: function(model, transition)

hooked called after models are resolved (use for late redirect)

deactivate: function

hook executes when the router completely exits this route

model: function(params)

provides data to be used by the controller and the view

renderTemplate: function(controller, model)

hook to override default template rendered for this route

serialize: function(model)

converts model into parameters for the url

setupController: function(controller, model)

function that can be used to configure the controller

actions

object with properties

actions: { willTransition: function(transition) }

called whenever transition triggered on current route

actions: { error }

});



Ember.View

Ember.View.extend: (*mixins*, {

attributeBindings: [*dataSize*, *href*]

array of View's property names used to calculate View's DOM element's attributes

classNameBindings: [*isAvailable*, *color*]

array of View's property names used to calculate View's DOM element's class attribute

classNames: [*color*, *size*]

array or string of View's class attribute

controller: *Ember.Controller.create*({ })

instance of descendants of the *Ember.Controller*

defaultTemplate:

Ember.Handlebars.compile('...')

compiled Handlebars template used when the view doesn't have the template or *templateName* property specified

eventManager: { }

an object with properties named after events that this view handles and values are functions that process these events

layout: *Ember.Handlebars.compile*('...')

compiled Handlebars template that wraps the view

tagName: *'em'*

string HTML tag to be used for View's DOM element

template: *Ember.Handlebars.compile*('...')

compiled Handlebars template used to render this view

templateName: *'some-template'*

string name of the template to be used to render this view (used instead of *template*)

});

Ember.Object

Ember.Object.extend: (*mixins*, {

init: *function*()

method called when an instance of this class is created

});

Courtesy of

<http://embersherpa.com/cheatsheet/>



By Denis PoÅ¼enel (mwore)
cheatography.com/mwore/
mwore.com

Published 25th September, 2013.
Last updated 1st June, 2014.
Page 2 of 2.

Sponsored by CrosswordCheats.com
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>