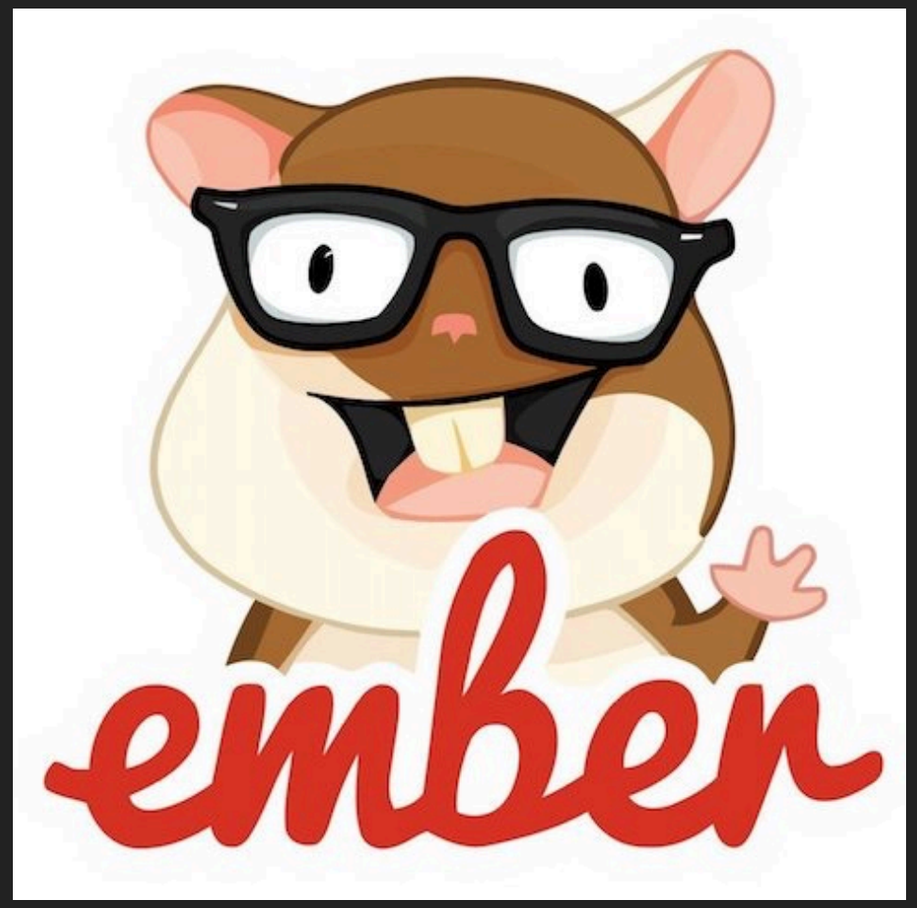


AN OVERVIEW FOR CREATING AMBITIOUS APPLICATIONS

JOHN DERR - SR. APPLICATION DEVELOPER - CARESOURCE

EMBERJS

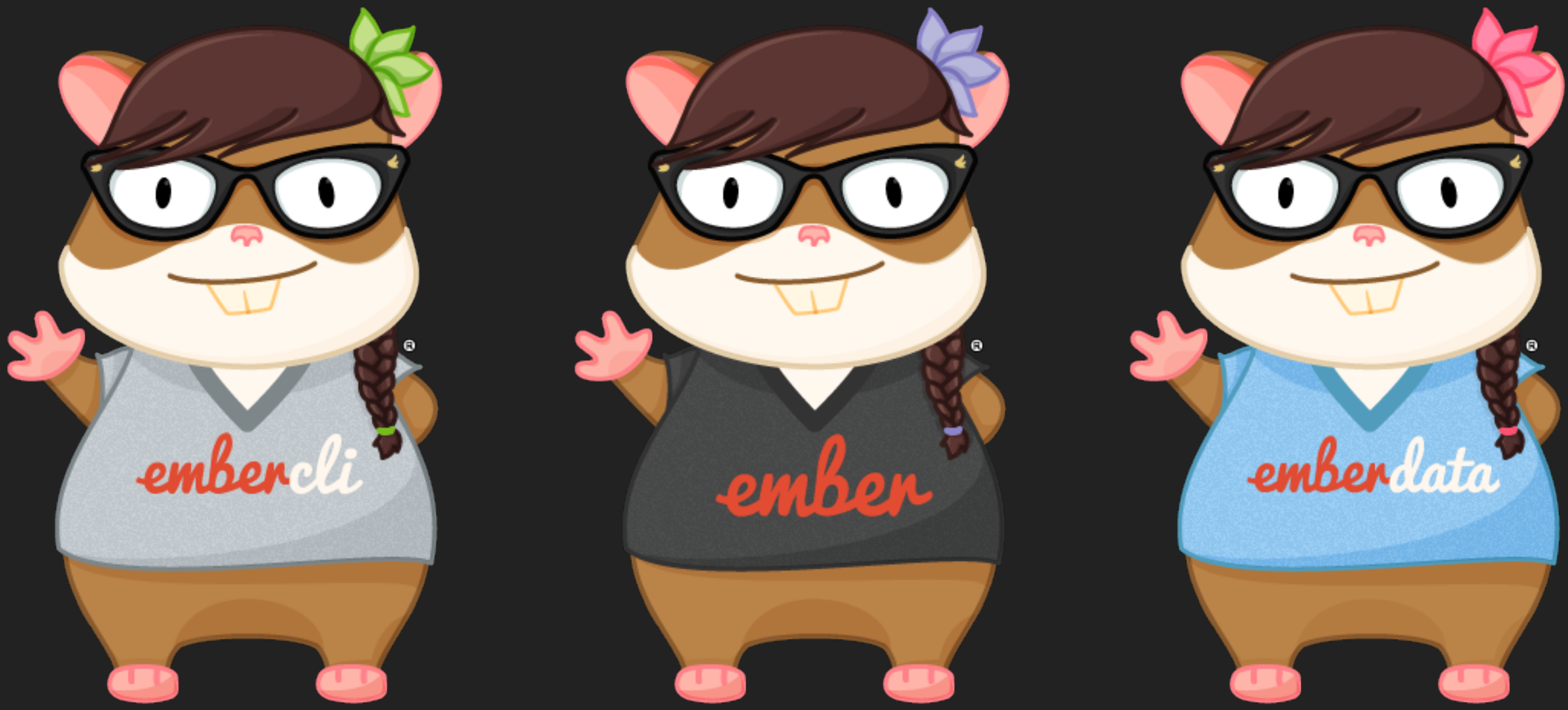
HISTORY

- Roots in Sproutcore a widget library used by apple to create MobileMe (c. 2008)
- Sproutcore 2.0 was an MVC framework that was open sourced and eventually because EmberJS (c. 2011)
- Ember 1.0 released 9/2013
- Ember 2.0 released 8/2015
- Current version is 2.15

PHILOSOPHY

- Open Source SDK For the Web
- Convention over Configuration
- Emphasis Developer Ergonomics
- Anyone can propose a feature for Ember via the RFC process.

MAKE UP



FEATURES

- MVC Architecture
- CLI designed to improve developer efficiency
- Ember Data provides global store for persisting state and a client side ORM for querying your api
- Strong conventions minimize design decisions and focus on productivity
- Component driven design since Ember 2.0
- Deprecation workflow
- LTS releases starting with Ember 2.4
- Advanced features include lazy loading of code (engines) and server side rendering (ember fast boot)
- Tooling extends to the browser with Ember Inspector dev tool for Chrome and Firefox

EMBER CLI

- Provides and runs tooling
- Create a new application with `ember new <app-name>`
- local dev server with `ember serve` supports live reload and backend proxying
- code generation with `ember generate <blueprint>`
- test runner with `ember test` or going to `http://localhost/tests`
- dependency management with `npm` or `yarn` for addons with `ember install`
- Runs build and asset compilation with `ember build`
- Build process uses babel transpiler

EMBER OBJECT MODEL

- Most objects in an Ember app extend from Ember.Object.
- Aligns as much as possible with with ES2015
- Property access occurs through the use of `get` and `set` methods instead of dot notation
- Supports computed properties and observers
- Computed Properties - function as property, pass in properties as dependent keys, updates when those properties change assuming you are running `get` on that property
- Observers - observe properties and run when they change

EMBER DATA

- Data persistence layer for Ember applications. Persists data using a store that is injected into the app as a service.
- Model attributes are defined in the models directory, allows for easy CRUD operations, supports relationships.
- Methods for finding collections or single records.
- Data fetching occurs though the adapter, default is JSONAPIAdapter supporting the JSONAPI spec, RESTAdapter is available.
- Custom adapters can be implemented to work with an API that is not RESTful

CORE CONCEPTS

- URL http://localhost/feed
- Router
- Route Handler
- Template
- Components

ROUTER

- Manages application state and URLs
- Maps URL to a route handler
- `Ember.Router.map(`*`callback`*`)` sets up the router
- Routes are added as `this.route` entries in router.js
- `this.route('feed')`
- `this.route('post', { path: '/post/:id' })`

ROUTE HANDLER

- Extends `Ember.Route`
- Contains lifecycle hooks that setup model
- Renders one or more templates
- Handle a redirect to another route
- Define and Handle route actions

CONTROLLERS

- Extends `Ember.Controller`
- Receives the model from the route
- Define actions and properties for use in route template
- Same name as route
- If a controller has not been generated for a given route Ember will create it at run time
- Slated for deprecation in favor of routable components

ROUTE LIFECYCLE HOOKS

- Model related hooks - route transition is paused if these hooks return an promise. Data is typically fetched using ember-data but not a requirement.
  - `beforeModel(transition)` - abort, retry a transition, fetch data needed before the model is needed.
  - `model(params, transition)` - fetch data needed by the route.
  - `afterModel(model, transition)` - after the model is resolved, perform logic that requires the routes model.
  - `modelFor(`*nameOfParent*`)` - return a parent or ancestor route's resolved model.
- `setupController(model, controller)` - setup the controller with the resolved model. Make additional ember-data calls.

TEMPLATES

- Each route has one or more associated Handlebars templates that are saved as .hbs files.
- Templates are rendered into the `{{outlet}}` tag of its parent.
- Templates are composable using components and template helpers.
- Also supports the use of partial templates using the `{{partial}}` helper, components are preferred.

TEMPLATE HELPERS

- Display properties `{{get}}` `{{concat}}`
- Flow control `{{if}}` `{{unless}}` `{{each}}` `{{each-in}}`
- Links `{{link-to}}`
- Actions `{{action}}`
- Input helpers `{{input}}` defaults to text, 'type' property can specify checkbox
- See Ember.template.helpers

COMPONENTS

- Building blocks of UI
- ember-cli blueprint generates a template, js source file and integration test
- Components can be inline or block style with use of `{{yield}}` helper
- Supports the use of the `{{action}}` helper in conjunction with the ‘on’ keyword to trigger touch, mouse, and keyboard events.
- Actions from controller must be passed in to interact with controller and route

COMPONENT LIFECYCLE HOOKS

- On component render
  - `init`, **`didReceiveAttrs`**, `willRender`, **`didInsertElement`**, `didRender`
- On component re-render
  - **`didUpdateAttrs`**, **`didReceiveAttrs`**, `willUpdate`, `willRender`, `didUpdate`, `didRender`
- On component destroy
  - `willDestroyElement`, `willClearRender`, `didDestroyElement`

ACTIONS

- Routes, Controllers, and Components all have an actions hash
- Actions can be triggered via the `{{action}}` helper in a template
- Actions can be triggered in code using the method `sendAction('<actionName>', args)`
- `sendAction` allows actions to bubble up from controller to route
- Ember 1.13 introduced closure actions. This is the preferred method of passing in actions going forward.
- `{{some-component saveAction=(action 'someAction')}}`

MIXINS AND SERVICES

- `Ember.Mixin` class allows you to add properties to other objects that implement the mixin.
- Services are singlentons that can be shared via dependency injection. Useful for sharing code and state such as user sessions.
- Inserted into components, controllers and routes using dependency injection.

ADDONS

- Ecosystem for bringing in reusable libraries to solve other problems.
  - Authentication, i18n, template helpers
  - 3rd party UI components, css frameworks, fonts
  - Build tools, SASS/LESS compilation, code linting

ADDON SAMPLING

- ember-i18n - https://github.com/jamesarosen/ember-i18n
- ember-cli-flash - https://github.com/poteto/ember-cli-flash
- ember-cp-validations - http://offirgolan.github.io/ember-cp-validations/
- ember-concurrency - http://ember-concurrency.com
- ember-cli-mirage - http://www.ember-cli-mirage.com

TESTING

- Ember uses qunit as its testing framework
- Tests can be run a few ways
  - `localhost:4200/tests`
  - `ember test` (runs once)
  - `ember test --server` (live reload in terminal)
- Command line tests can use PhantomJS or Headless Chrome
- Runs code against linting rules along with tests

TESTING

- 3 levels of tests
- Unit tests - test small portions of code in isolation
- Acceptance tests - automated browser testing
- Integration tests - test components in isolation, render components and drive user interaction

MORE ADDONS

- Ember Observer - https://emberobserver.com
- ember-simple-auth - torii
- ember-power-select
- ember-cli-bootstrap-4
- ember-font-awesome
- ember-truth-helpers
- ember-modal-dialog - ember-wormhole
- eslint-plugin-ember-suave
- ember-cli-deprecation-workflow
- ember-exam
- ember-web-app - ember-service-worker

LEAR MORE ABOUT EMBER

- Columbus Ember Meetup
- Ember Community Slack
- Jen Weber - How to learn EmberJS in a hurry
- Ember Weekend Postcast
- Frontend Happy Hour - Ember Gin and Tomster
- Books
  - https://leanpub.com/ember-cli-101 (free - $9.99)
  - https://balinterdi.com/rock-and-roll-with-emberjs ($39 - $89)
- Videos
  - https://vimeo.com/globalembermeetup (free)
  - Lauren Tan Ember Conf github repos
  - https://tomdale.net/talks/
  - http://yehudakatz.com/talks

TRAINING AND CONSULTING

- Training
  - Frontend Masters($) - Mike North
  - https://www.emberscreencasts.com (free)
  - https://www.emberschool.com ($)
  - https://embermap.com ($) - consulting and mentoring
  - https://www.201-created.com ($) - consulting and mentoring

EMBER TALKS I REALLY LIKE

- Toran Billups EMBERCONF 2015 - TEST-DRIVEN DEVELOPMENT BY EXAMPLE
- Miguel Camba EMBERCONF 2017 - Higher Order Components
- Edward Faulkner Empowering the Next Million Creators

CONTACT INFO

- Email: john.derr@gmail.com
- Code: https://github.com/jderr-mx/kitter
- LinkedIn: https://www.linkedin.com/in/john-derr-40559b6
- Twitter: @johnderrdotnet