

Ember.js Cheat Sheet

Getting Started & CLI

Ember.js is a convention-driven SPA web framework built for developer productivity.

- **Create Project:** `npx ember-cli new my-app`
- **Start Dev Server:** `ember serve` (port 4200)
- **Generate Component:** `ember generate component user-card`
- **Generate Route:** `ember generate route posts`
- **Generate Service:** `ember generate service auth`
- **Run Tests:** `ember test / ember test --server`
- **Production Build:** `ember build -prod`

Glimmer Components

Lightweight reactive components powered by the `@tracked` decorator.

Component Class (`app/components/counter.js`)

```
import Component from '@glimmer/component';
import { tracked } from '@glimmer/tracking';
import { action } from '@ember/object';

export default class CounterComponent extends Component {
  @tracked count = 0;

  get doubleCount() {
    return this.count * 2;
  }

  @action
  increment() {
    this.count++;
  }
}
```

Handlebars Template (`app/components/counter.hbs`)

```
<div class="counter">
  <p>Count: {{this.count}} (Double: {{this.doubleCount}})</p>
  <button type="button" {{on "click" this.increment}}>+1 Count</button>
</div>
```

```
<p>Received Title: {{@title}}</p>
</div>
```

Handlebars Template Syntax

- **Property**
Evaluation: `{{this.propertyName}}`
- **Argument Access:** `{{@argumentName}}`
- **Event Modifiers:** `{{on "click" this.handleClick}}`
- **Conditionals:** `{{#if this.isLoggedIn}} ... {{else}} ... {{/if}}`
- **Iteration:**
`<ul> {{#each this.items as |item index|}} <li>{{index}}: {{item.name}}</li> {{/each}} </ul>`
- **Routing Links:** `<LinkTo @route="posts.show" @model={{post.id}}>View</LinkTo>`

Routing & Model Hook

Loads data based on URL routes and renders hierarchy.

```
// app/routes/posts.js
import Route from '@ember/routing/route';
import { service } from '@ember/service';

export default class PostsRoute extends Route {
  @service store;

  async model(params) {
    return this.store.findAll('post');
  }
}
```

Ember Data & Models

Model Definition (`app/models/post.js`)

```
import Model, { attr, belongsTo, hasMany } from '@ember-data/model';
```

```
export default class PostModel extends Model {
  @attr('string') title;
  @attr('string') content;
  @attr('date') createdAt;

  @belongsTo('user', { async: true, inverse: 'posts' }) author;
  @hasMany('comment', { async: true, inverse: 'post' }) comments;
}
```

Store CRUD Operations

- **Find All:** `const posts = await this.store.findAll('post');`
- **Find Record:** `const post = await this.store.findRecord('post', id);`
- **Query Records:** `const results = await this.store.query('post', { tag: 'tech' });`
- **Create Record:**
`const newPost = this.store.createRecord('post', { title: 'New Post' });`
`await newPost.save();`
- **Delete Record:**
`await post.destroyRecord();`

Services & Dependency Injection

```
// app/services/auth.js
import Service from '@ember/service';
import { tracked } from '@glimmer/tracking';

export default class AuthService extends Service {
  @tracked currentUser = null;
  login(user) { this.currentUser = user; }
}

// Injected into component or route
import { service } from '@ember/service';
// ...
@service auth;
```