

Table of Contents



- [1. Installation and Setup](#)
 - [Install Ember CLI \(Globally\):](#)
 - [Create a New Ember Project:](#)
- [2. Project Structure \(Key Folders\)](#)
- [3. Running the Application](#)
- [4. Generating New Files](#)
- [5. Creating a Route](#)
 - [Generate a Route:](#)
 - [app/router.js:](#)
 - [app/routes/about.js:](#)
 - [app/templates/about.hbs:](#)
- [6. Creating a Component](#)
 - [Generate Component:](#)
 - [app/components/hello-world.js:](#)
 - [app/templates/components/hello-world.hbs:](#)
 - [Using the Component \(in application.hbs\):](#)
- [7. Template Syntax \(Handlebars\)](#)
 - [Dynamic Data Binding:](#)
 - [Conditional Rendering:](#)
 - [Looping \(each\):](#)
- [8. Actions \(Event Handling\)](#)
- [9. Two-Way Data Binding \(@tracked\)](#)
- [10. Models \(Ember Data\)](#)
 - [Generate Model:](#)
 - [app/models/user.js:](#)
- [11. Fetch Data from API \(Model Hook\)](#)
- [12. Forms and Input Binding](#)
- [13. Services \(Global State Management\)](#)
 - [Generate Service:](#)
 - [app/services/auth.js:](#)
 - [Injecting Service \(Component\):](#)
- [14. Helper Functions](#)
- [15. Testing \(Ember QUnit\)](#)
- [16. Useful Commands](#)

1. Installation and Setup

Install Ember CLI (Globally):

```
npm install -g ember-cli
```

Create a New Ember Project:

```
ember new my-ember-app  
cd my-ember-app  
ember serve
```

- Visit:

<http://localhost:4200>

2. Project Structure (Key Folders)

my-ember-app/

— app/	# Main app code
— components/	# Reusable components
— controllers/	# Controllers for routes
— models/	# Data models (Ember Data)
— routes/	# Route handlers
— services/	# Shared logic/services
— templates/	# Handlebars templates
— styles/	# CSS/SCSS styling
— public/	# Static files
— tests/	# Automated tests
— config/	# Environment configuration
— package.json	# Project dependencies

3. Running the Application

```
ember serve
```

- Development Server:

`http://localhost:4200`

4. Generating New Files

```
ember generate component my-component
ember generate route about
ember generate model user
ember generate service auth
```

5. Creating a Route

Generate a Route:

```
ember generate route about
```

app/router.js:

```
import EmberRouter from '@ember/routing/router';
import config from './config/environment';

const Router = EmberRouter.extend({
  location: config.locationType,
  rootURL: config.rootURL
});

Router.map(function() {
  this.route('about');
});
```

```
export default Router;
```

app/routes/about.js:

```
import Route from '@ember/routing/route';

export default class AboutRoute extends Route {
  model() {
    return { message: 'Welcome to the About Page!' };
  }
}
```

app/templates/about.hbs:

```
<h1>{{model.message}}</h1>
```

6. Creating a Component

Generate Component:

```
ember generate component hello-world
```

app/components/hello-world.js:

```
import Component from '@glimmer/component';

export default class HelloWorldComponent extends Component {
  message = 'Hello from Ember!';
}
```

app/templates/components/hello-world.hbs:

```
<h1>{{this.message}}</h1>
```

Using the Component (in application.hbs):

```
<HelloWorld />
```

7. Template Syntax (Handlebars)

Dynamic Data Binding:

```
<h1>{{model.title}}</h1>
```

Conditional Rendering:

```
{{#if user.isLoggedIn}}  
  <p>Welcome back!</p>  
{{else}}  
  <p>Please log in.</p>  
{{/if}}
```

Looping (each):

```
<ul>  
  {{#each this.items as |item|}}  
    <li>{{item.name}}</li>  
  {{/each}}  
</ul>
```

8. Actions (Event Handling)

```
<button {{on "click" this.sayHello}}>Click Me</button>
```

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

```
export default class ButtonComponent extends Component {  
  @action  
  sayHello() {  
    alert('Hello, Ember!');  
  }  
}
```

9. Two-Way Data Binding (@tracked)

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

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

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

<p>Count: {{this.count}}</p>
<button {{on "click" this.increment}}>Increment</button>
```

10. Models (Ember Data)

Generate Model:

```
ember generate model user
```

app/models/user.js:

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

export default class UserModel extends Model {
  @attr name;
  @attr email;
}
```

11. Fetch Data from API (Model Hook)

```
import Route from '@ember/routing/route';
import { service } from '@ember/service';

export default class UsersRoute extends Route {
  @service store;

  model() {
    return this.store.findAll('user');
  }
}
```

12. Forms and Input Binding

```
<input type="text" value={{this.username}} {{on "input"
this.updateName}}>
<p>Your name: {{this.username}}</p>

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

export default class NameFormComponent extends Component {
  @tracked username = '';

  @action
  updateName(event) {
    this.username = event.target.value;
  }
}
```

13. Services (Global State Management)

Generate Service:

```
ember generate service auth
```

app/services/auth.js:

```
import Service from '@ember/service';
import { tracked } from '@glimmer/tracking';

export default class AuthService extends Service {
  @tracked isLoggedIn = false;

  login() {
    this.isLoggedIn = true;
  }
}
```

Injecting Service (Component):

```
import Component from '@glimmer/component';
import { inject as service } from '@ember/service';

export default class NavbarComponent extends Component {
  @service auth;
}

{{#if this.auth.isLoggedIn}}
  <p>Welcome User!</p>
{{else}}
  <button {{on "click" this.auth.login}}>Login</button>
{{/if}}
```

14. Helper Functions

```
ember generate helper capitalize
```

```
import { helper } from '@ember/component/helper';
```

```
export default helper(function capitalize([text]) {  
  return text.toUpperCase();  
});
```

```
<p>{{capitalize "hello ember"}}</p>
```

15. Testing (Ember QUnit)

```
ember test
```

- Run Specific Test:

```
ember test --filter="component:hello-world"
```

16. Useful Commands

ember serve	# Start development server
ember build	# Build project
ember test	# Run tests
ember generate route about	# Generate route
ember generate component nav	# Generate component
ember generate model post	# Generate model

This cheat sheet covers essential topics to kickstart your Ember.js projects