Ember.js

Un Framework pour des interfaces web robustes

ember

PRÉSENTÉ PAR

FLORES HERNANDEZ Marisela
ZABKOWSKI Kacper

Objectifs

Contenu

- Présenter Ember.js comme technologie moderne.
- Comprendre les concepts fondamentaux du framework (routes, modèles, templates, composants).
- Comparer Ember.js avec ses technologies concurrentes (React, Angular).
- Identifier les outils de l’écosystème Ember et leurs usages.
- Apprendre à créer une application simple grâce à un tutoriel “HelloWorld”.

.

PRESENTATION

01 La cible applicative

02 Les avantages et inconvénients

03 Les bases de la technologie

04 Les outils associés

05 Les technologies concurrentes

06 Les acteurs du marché

07 Libraires et logiciels à installer

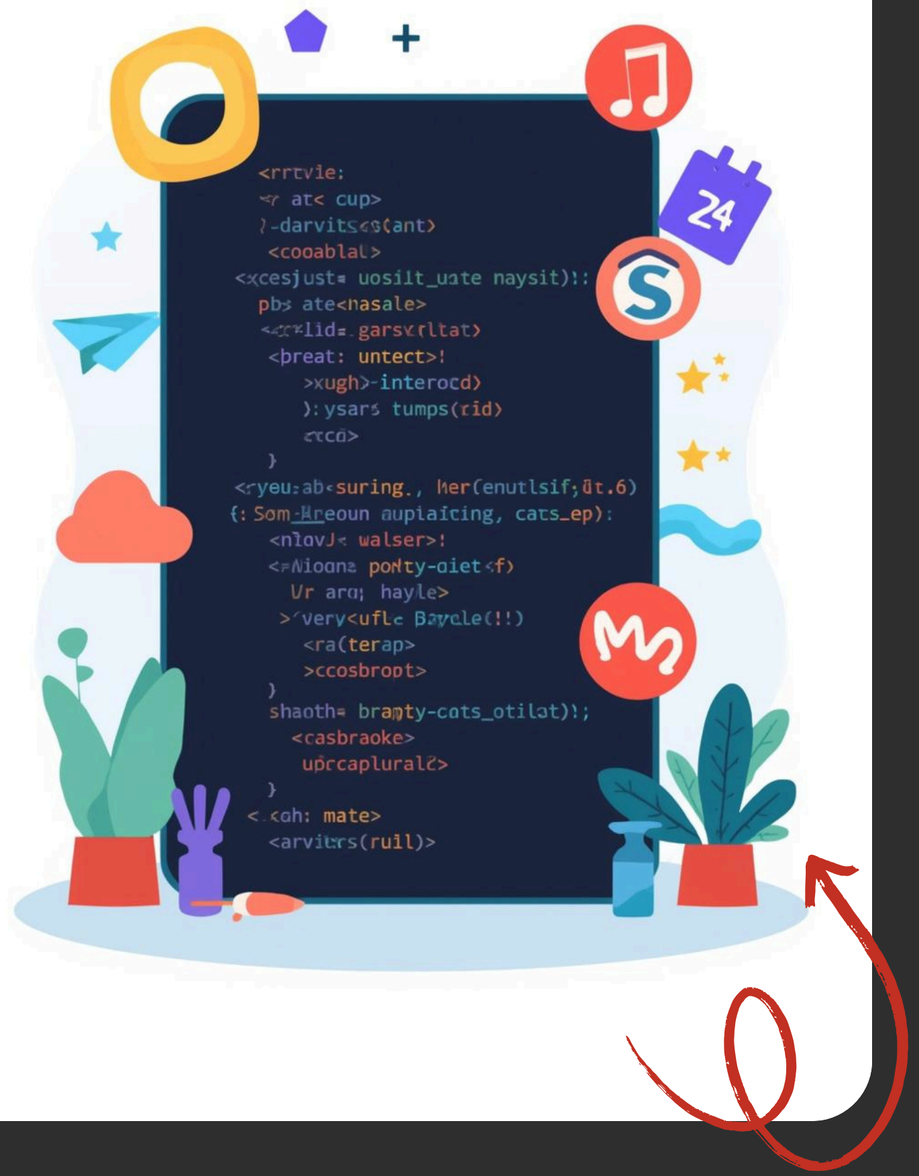
08 Exemple et tutoriel

09 Les références bibliographiques

Introduction à Ember.js

Un framework JavaScript

- Ember.js est un framework JavaScript open-source conçu pour développer des applications web ambitieuses.
- Créé en 2011, il facilite la création d'interfaces riches, évolutives et maintenables grâce à sa structure opportune.
- Adopte une philosophie “Convention over Configuration”, facilitant la structure et la maintenance des projets.



Philosophie derrière Ember.js

Convention over Configuration

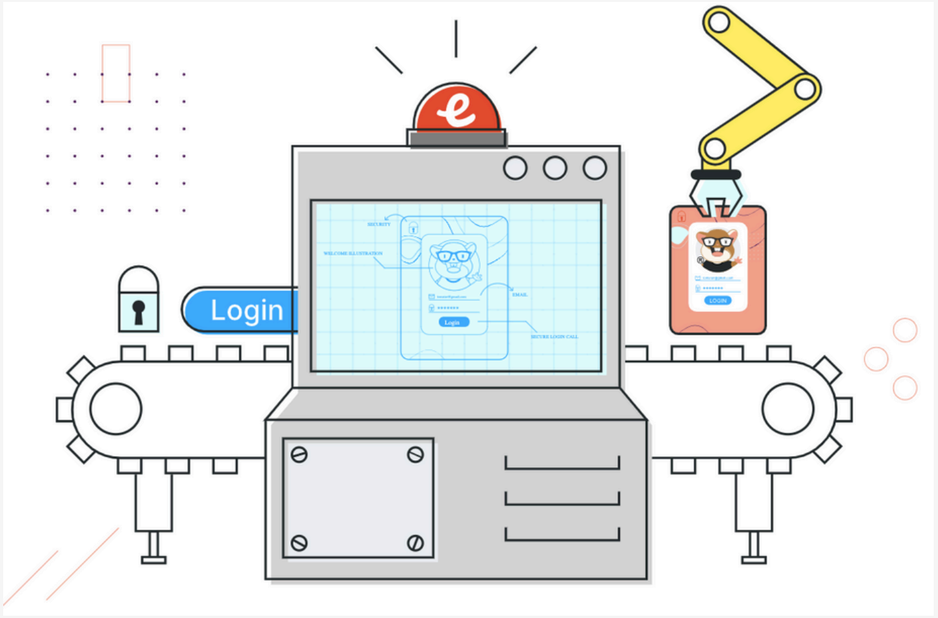
Ember favorise la cohérence et la productivité grâce à des conventions fortes qui évitent les configurations manuelles.

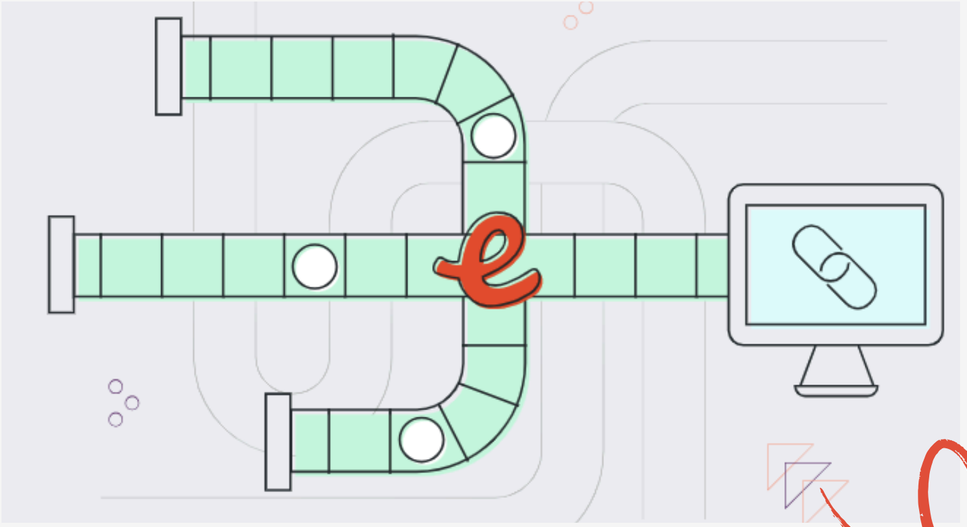
Batteries incluses

Router, CLI, Data, moteur de rendu : tout est fourni pour créer une application moderne dès le premier jour.

Stabilité et longévité

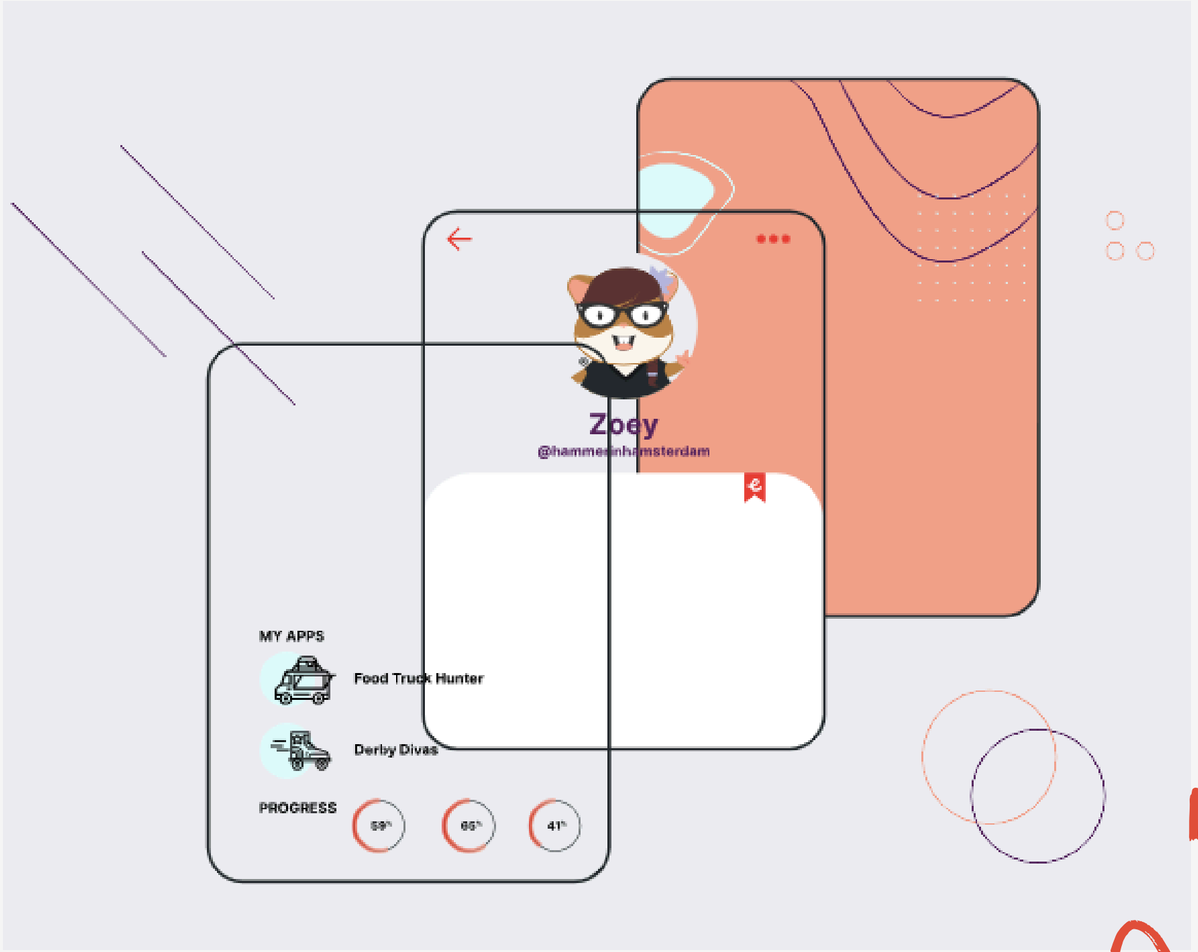
Ember garantit des mises à jour progressives, une architecture stable et un cycle de développement prévisible.





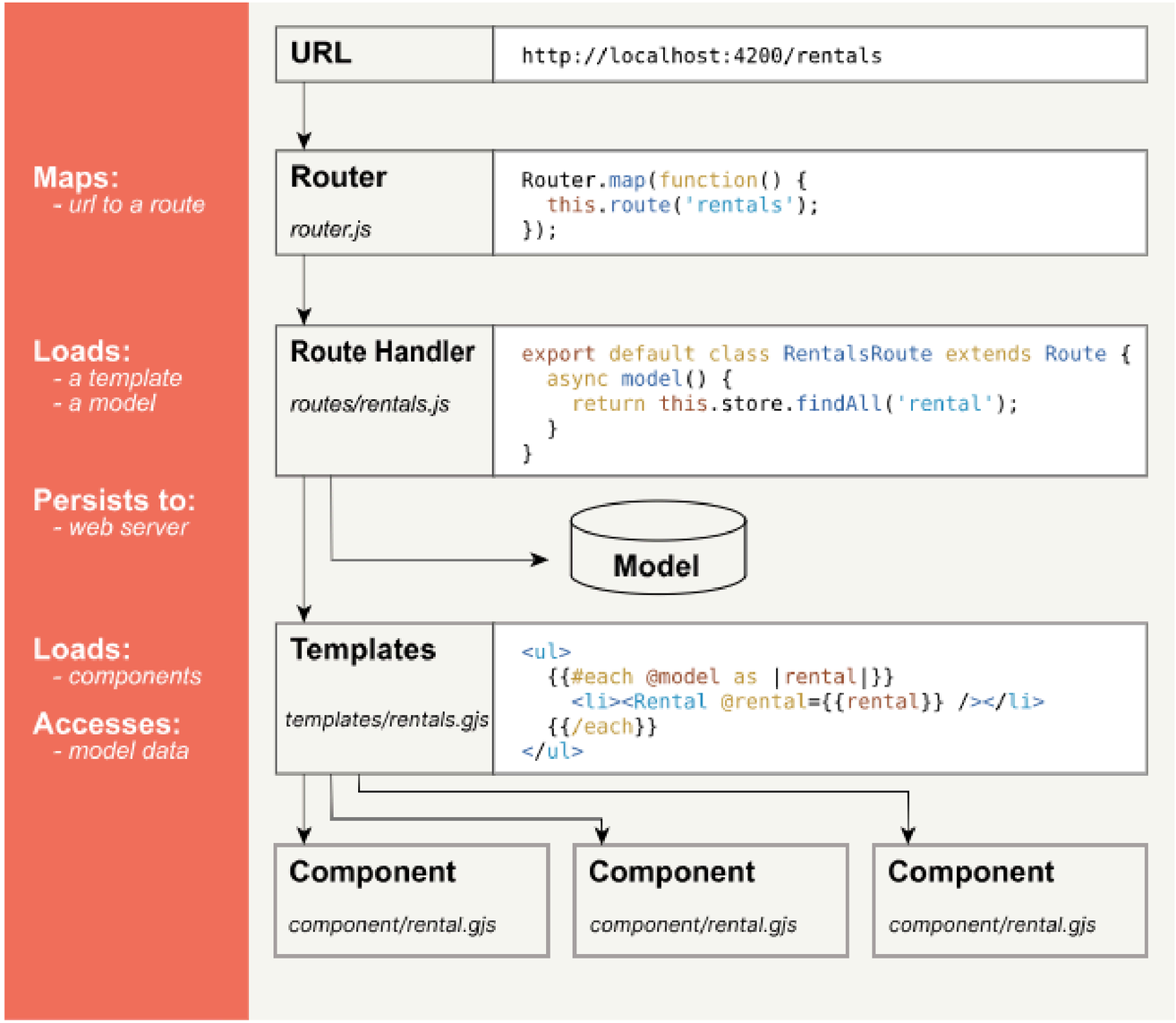
Ember.js: Un Framework JavaScript puissant et moderne

- Basé entièrement sur le langage JavaScript.
- Gère toute la logique d’interface en JavaScript.
- Offre une architecture complète : routes, modèles, templates, composants
- Suit les standards modernes du langage (ES6, classes, décorateurs).
- Permet de créer des Single Page Applications modernes.



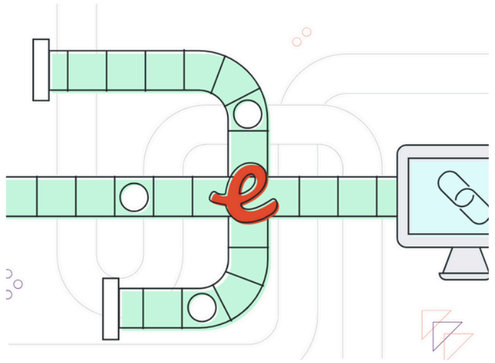
Concepts clés de Ember.js

Une architecture MVVM



Architecture interne : le cycle Route → Model → Template → Component

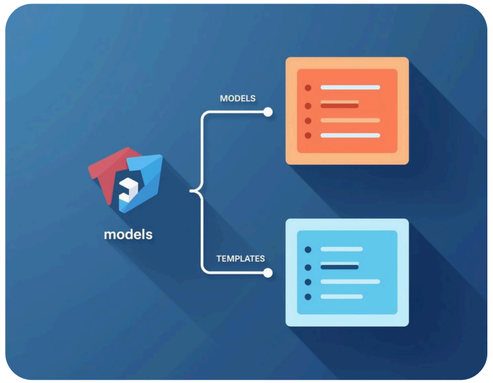
Route



Point d'entree :

- Analyse d'URL
- Décide quelles données charger
- Coordonne l'affichage des écrans

Model



Données de l'application :

• Représente l'état actuel de l'interface
• Chargé depuis une API (via Ember Data) et calculé localement (client)
• Fournit l'information nécessaire au template

Template

Interface déclarative :

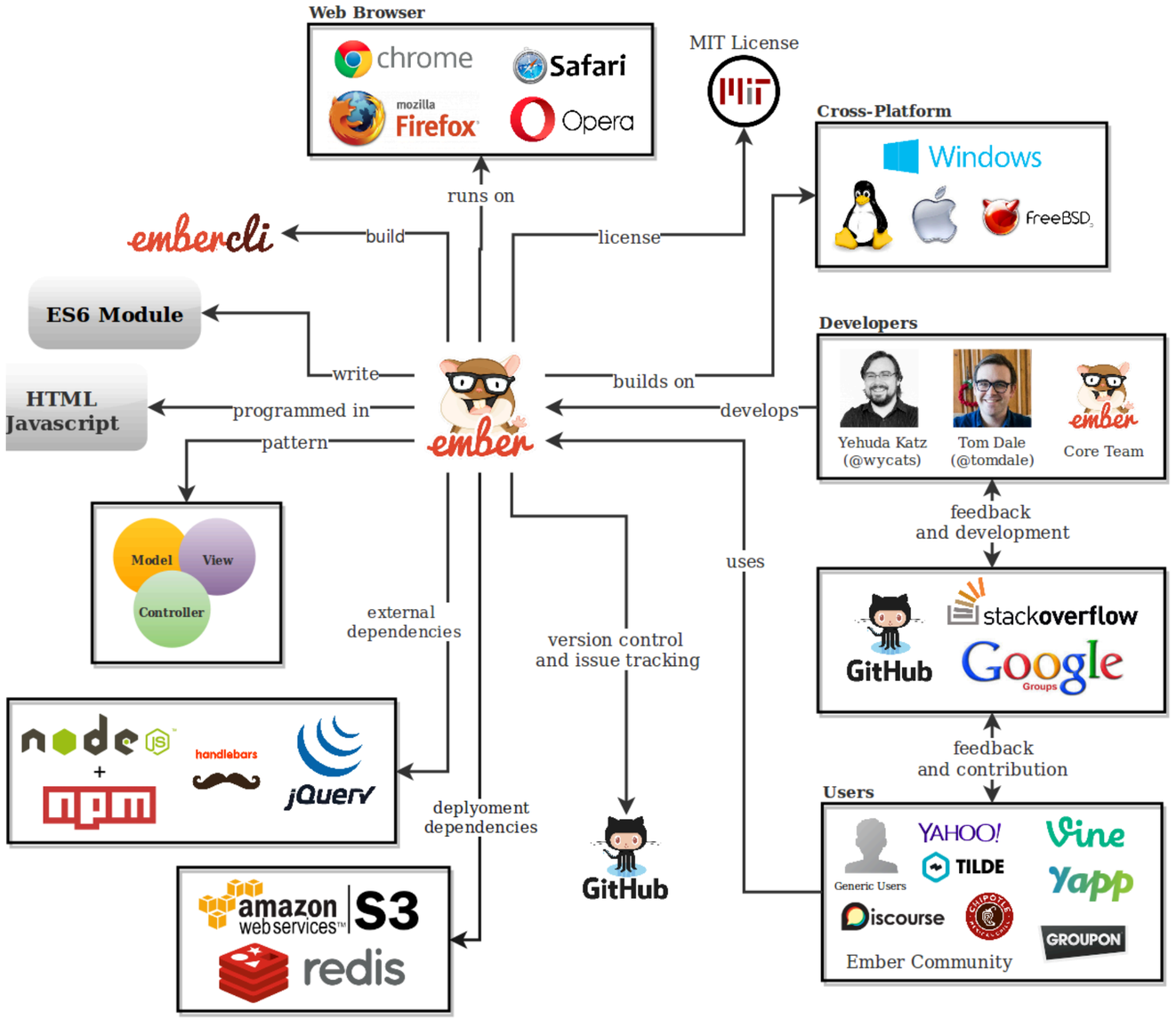
• Écrit en HTMLBars / Handlebars
• Décrit ce que l'utilisateur voit
• Se met à jour automatiquement quand les données changent

Component

Interaction et logique locale :

• Éléments UI réutilisables
• Contiennent leur propre logique + template
• Gèrent les interactions (clics, formulaires, événements)

Diagramme de vue contextuelle d'Ember.js



Ember.js permet de créer les Single Page Applications

SPA

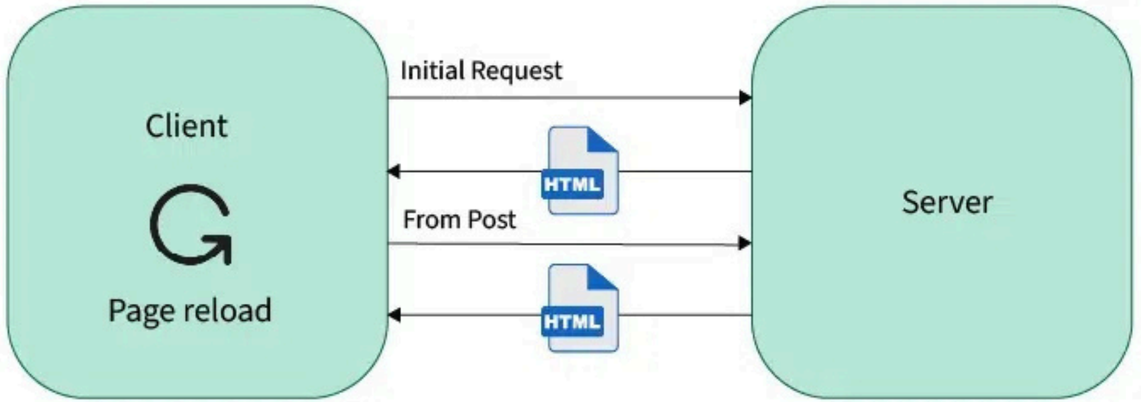
Single Page Application: est une application ou site web qui interagit avec l'utilisateur par la réécriture dynamique de la page. Elle utilise les données en provenance du serveur plutôt que charger une toute nouvelle page .

Quand l'utiliser?

- Interactions dynamiques: usage fréquent sans rafraichissement
- Consistance Cross-Platformes
- Mobile-Friendly
- Diminution de la charge du serveur
- Mises-à-jour en temps réel

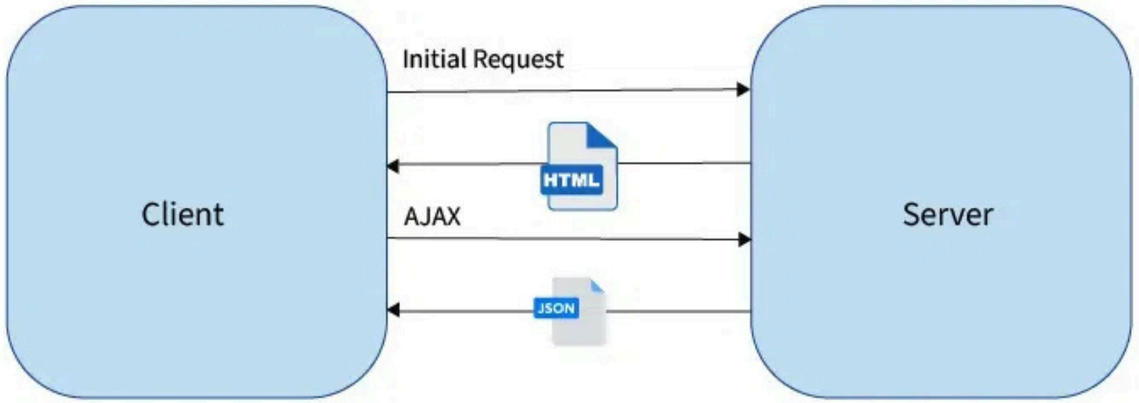
Traditional Page Lifecycle

Once the user accesses the page and performs any kind of action on that page, the page gets reloaded with the changes that were done on the server-side.



Single Page Lifecycle

Once the user accesses the page and performs any kind of action on that page, they get an almost-instant reaction from the page (think of Facebook, when you comment on someone's post.)



Évolution d'Ember.js

Lancement initial

Ember.js a été lancé en 2011.Naissance du projet basé sur SproutCore.

Version 1.0

La première version stable 1.0 d'Ember.js a été publiée en 2014.

Ember CLI

La sortie en 2015 d’Ember CLI : révolutionne l'écosystème avec des outils modernes.

Ember Data

En 2016 Ember Data a été introduit pour gérer les modèles et les données, améliorant l'efficacité du développement avec des fonctionnalités de liaison de données.

Glimmer VM

Nouveau moteur : Glimmer VM.
Et en 2019, Ember Octane : architecture moderne et composants Glimmer.

2020 - 2025

Améliorations continues, compatibilité ES6, TypeScript.
Dernière version 6.8

LA CIBLE APPLICATIVE

Framework Ember.js

→ Développement Web →
- Single Page Applications (SPA)
- Dashboards complexes
- Portails internes / apps d'entreprise
- Applications longue durée


→ Développement Mobile →

Applications mobiles (avec Cordova / Capacitor)
- Apps hybrides simples
- Progressive Web Apps (PWA)

Applications desktop via Electron
- Applications desktop basées sur la même base de code
- Outils internes multi-plateforme

Pourquoi choisir Ember.js ?

Choisir la bonne technologie IU

Développement moderne

Ember.js est conçu pour les **applications web ambitieuses**, ce qui en fait un choix idéal pour les développeurs modernes qui cherchent à créer des expériences utilisateur interactives et évolutives.

Facilite le développement

Grâce à sa conception basée sur des conventions, Ember.js simplifie le processus de développement, permettant de se concentrer sur la création d'applications plutôt que de se perdre dans les détails de configuration.

Écosystème robuste

Ember.js dispose d'un solide écosystème d'outils et de bibliothèques, fournissant aux développeurs tout ce dont ils ont besoin, du routage et de la gestion des données au soutien de la communauté pour un développement fluide.

Les outils associés

Ember Data

Ember CLI

glimmer

Outil de débogage propre (Ember Inspector)

Les outils associés

L'outil officiel de ligne de commande.

- **Génère la structure du projet automatiquement**
- **Fournit un serveur de développement, un système de build et les tests**
- **Assure une organisation uniforme pour toutes les applications Ember**

La bibliothèque de gestion des données.

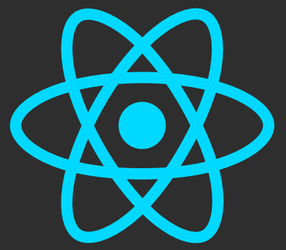
- **Fournit un ORM côté client**
- **Gère la synchronisation avec les API (REST, JSON:API, adapters)**
- **Définit des modèles, relations et chargements asynchrones**

Extension Chrome/Firefox conçue pour comprendre et déboguer les applications Ember.

Extension Ember CLI qui permet d'exécuter les applications dans Node.js.

Bibliothèque d'animations pour Ember. Permet d'ajouter des transitions fluides entre routes. Très utilisé pour des animations d'UI cohérentes et idéal pour de SPA interactives

03 TECHNOLOGIES CONCURRENTES



React

vs

ember

Critère	React	ember	Qui a l'avantage ?
Popularité et communauté	🟢 Très populaire, énorme communauté	🔴 Communauté plus petite	**React**
Flexibilité	🟢 Très flexible (router, store au choix)	🔴 Moins flexible (conventions fortes)	**React**
Facilité de démarrage	🟢 Facile à commencer	🔴 Conventions à apprendre	**React**
Structure du projet	🔴 Pas imposée → hétérogène	🟢 Structure claire et standardisée	**Ember**
Écosystème intégré	🔴 Fragmenté : router, store, tests séparés	🟢 « Batteries incluses » (CLI, Data, Router, Tests)	**Ember**
Mises à jour	🔴 Écosystème instable / ruptures possibles	🟢 Releases prévisibles + codemods	**Ember**
Petits projets / MVP	🟢 Très adapté	🔴 Pas idéal, trop "lourd"	**React**
Grands projets durables	🔴 Nécessite discipline et outils	🟢 Parfait pour projets longs et complexes	**Ember**

02 TECHNOLOGIES CONCURRENTES

VS

Critère	ANGULAR	ember	Qui a l'avantage ?
Popularité et communauté	🟢 Très populaire dans les grandes entreprises	🔴 Communauté plus réduite	**Angular**
Écosystème intégré	🟢 Framework complet (CLI, router, forms…)	🟢 Complet aussi (CLI, router, Data, tests)	**Équivalence**
Complexité	🔴 Très complexe, nombreuses notions (DI, RxJS…)	🟢 Plus simple grâce aux conventions	**Ember**
Courbe d'apprentissage	🔴 Difficile, beaucoup de concepts	🔴 Aussi difficile au début	**Difficile dans les deux cas**
Flexibilité	🔴 "Très opinionated", règles strictes	🔴 Opinionated aussi	**Équivalence**
Performance	🟢 Très performant, optimisations avancées	🟢 Très performant avec Glimmer VM	**Équivalence**
Taille du framework	🔴 Large, lourd à charger	🔴 Ember aussi assez lourd	**Les deux sont lourds**
Mises à jour	🔴 Migrations parfois difficiles	🟢 Upgrades simples + codemods	**Ember**
Idéal pour	🟢 Très grandes applis d'entreprise	🟢 Grandes applis long terme	**Même public cible**
Petits projets	🔴 Trop lourd	🔴 Trop lourd	**Ni Angular ni Ember**

Avantages d'Ember.js

Forces du framework moderne

Forte convention

Ember.js adopte des **conventions robustes** qui facilitent la collaboration entre les développeurs, améliorant la lisibilité et la maintenance du code, ce qui augmente l’efficacité du développement à long terme.

Stabilité

Ce framework offre une **stabilité exceptionnelle**, soutenue par une communauté active et des mises à jour régulières, assurant que les applications restent performantes et pertinentes dans un environnement technologique en évolution.

Écosystème intégré

L'écosystème d'Ember.js inclut des **outils et bibliothèques** intégrés qui simplifient le processus de développement, permettant aux équipes de se concentrer sur la création d'expériences utilisateur riches et dynamiques.

Inconvénients d'Ember.js

Limitations à considérer

Courbe d'apprentissage

Ember.js présente une **courbe d'apprentissage** initiale relativement élevée, ce qui peut décourager les nouveaux développeurs et retarder la mise en œuvre de projets pour ceux qui ne sont pas familiers avec le framework.

Popularité relative

Comparé à des frameworks comme React et Angular, Ember.js a une **popularité relative** plus faible, ce qui signifie moins de ressources communautaires, de tutoriels et de bibliothèques tierces disponibles pour les développeurs.

Taille du framework

La taille et la complexité du framework peuvent poser des défis en matière de **performance et de déploiement**, en particulier pour les petites applications ou celles qui nécessitent des temps de chargement rapides.

06

LES ACTEURS DU MARCHÉ

DE NOMBREUX SPONSORS, FINANCIERS ET TECHNIQUES

Tilde

AuditBoard

Mainmatter

CARDSTACK

ShipShape

06

LES ACTEURS DU MARCHÉ

ANCIENS SPONSORS

Et d’autres…

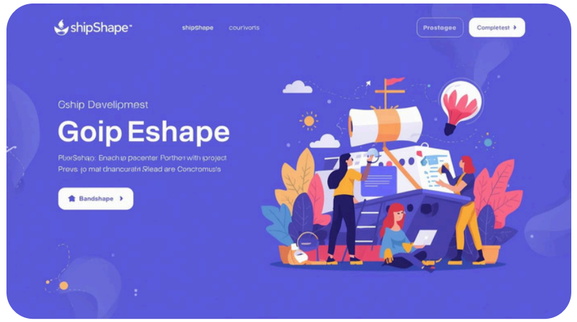
Acteurs clés dans l'écosystème Ember.js

Tilde

Entreprise spécialisée dans les solutions **modernes** de développement web.

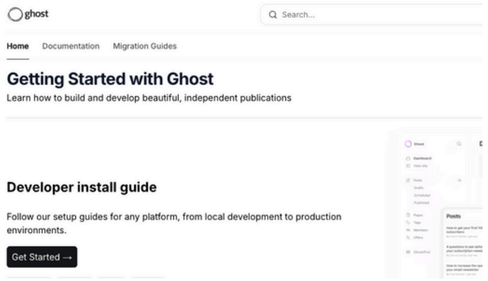
LinkedIn

Utilisation d'Ember.js pour des **interactions** utilisateur puissantes.



ShipShape

Innovation dans la gestion de **projets** logiciels.



Ghost (Ghost Admin)

Le panneau d’administration du Ghost est construit en Ember, car il nécessite une interface bien structurée, modulable et très réactive pour la gestion de contenu.

06

LES ACTEURS DU MARCHÉ

QUI UTILISE EMBER.JS?

Microsoft

dentally

LinkedIn

NETFLIX

PagerDuty

thoughtbot

squire

07 LIBRAIRIES ET LOGICIELS

Obligatoires:

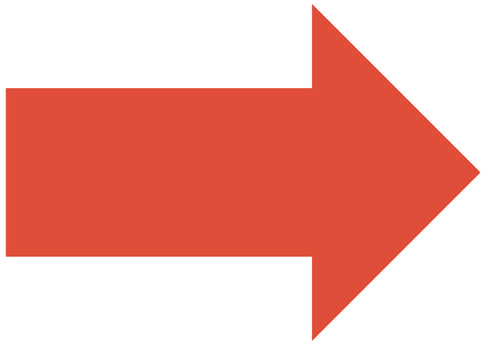
- Node.js ==> gestionnaire de paquets
- Ember.js ==> environnement de travail Ember.js

Recommandé (utilisateurs VSCode):

Extensions:

- Prettier - Code formatter
- Glimmer Templates Syntax for VS Code

```
3   <template>
4     <div class="jumbo">
5       <div class="right tomster"></div>
6       <h2>Welcome to Super Rentals!</h2>
7       <p>We hope you find exactly what you're looking for in a place to stay.
8       <LinkTo @route="about" class="button">About Us</LinkTo>
9     </div>
10  </template>
11
```



```
3   <template>
4     <div class="jumbo">
5       <div class="right tomster"></div>
6       <h2>Welcome to Super Rentals!</h2>
7       <p>We hope you find exactly what you're looking for in a place to stay
8       <LinkTo @route="about" class="button">About Us</LinkTo>
9     </div>
10  </template>
11
```

07 LIBRAIRIES ET LOGICIELS

Étapes à suivre:

1. Installer npm, yarn, ou pnpm
2. Installation de git (recommandé)
3. Si utilisateur Linux ou Mac → Installer Watchman
4. Installer Node.js → Inclut dans yarn et pnpm

Pourquoi Watchman?

Pour certaines versions Linux et Mac, le serveur Ember ne se réinitialise pas automatiquement lors des changements.

08 Exemple et tutoriel

1/Installation Ember.js

1. npm install -g ember-cli
2. ember new <my-app-name> [options]

Création de notre dossier Ember. On choisit le nom et des options supplémentaires (“--strict” pour lancer le mode strict de JavaScript)

08 Exemple et tutoriel

2/ Se placer dans le dossier et activer le projet

- cd <nom-de-notre-application>
- npm start

On a activé notre projet Ember.js.

!!! Si utilisateur Windows, alors il pourrait être nécessaire de lancer CMD en mode administrateur.

```
VITE v7.2.2  ready in 227721 ms

➜  Local:   http://localhost:4200/
➜  Network: use --host to expose
➜  press h + enter to show help
- building...
```

adresse de l'appli

08

Page créée automatiquement, retrouvée dans :

<nom-appli-ember>\app\templates\application.gjs

Congratulations, you made it!

You've officially spun up your Ember app. You've got one more decision to make: what do you want to do next? We'd suggest one of the following to help you get going:

- Quick Start - a quick introduction to how Ember works. Learn about defining your first route, writing a UI component and deploying your application.
- Tutorial - this is our more thorough, hands-on intro to Ember. Your crash course in Ember philosophy, background and some in-depth discussion of how things work (and why they work the way they do).

If you run into problems, please join our community's Discord server or visit our forums for ideas and answers — our community is filled with friendly folks who are willing to help! We enjoy helping new Ember developers get started, and our Ember Community is incredibly supportive.

08 3/Génération de nouvelles routes

- ember generate route <nom-de-votre-route>

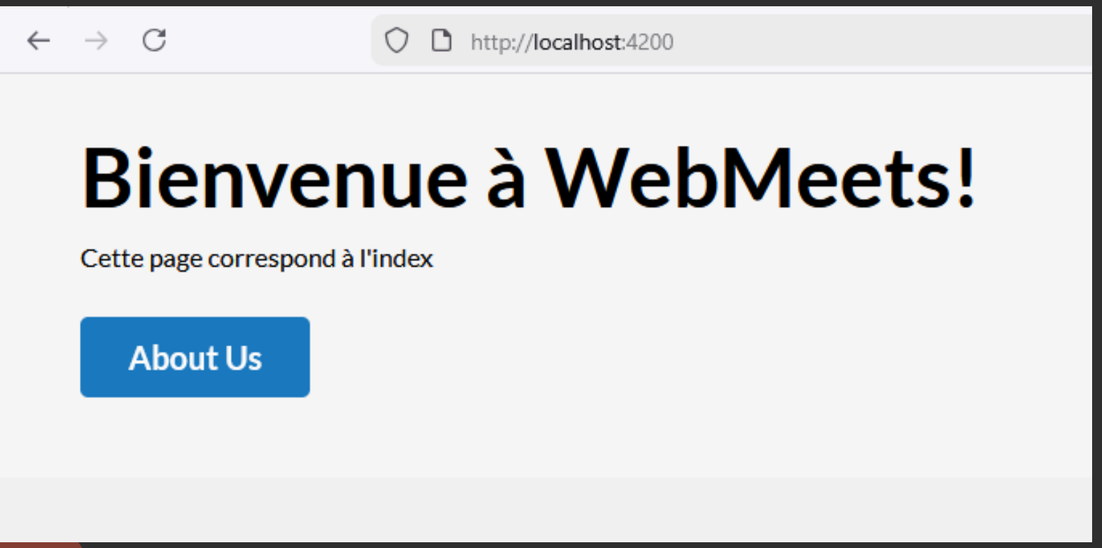
```
C:\Users\zabko\Desktop\Master-DCISS-M2\Semestre1\IHM\ember-quickstart>ember generate route about
installing route
  create app\routes\about.js
  create app\templates\about.gjs
updating router
  add route about
installing route-test
  create tests\unit\routes\about-test.js
```

On utilise cette commande pour créer une nouvelle route, template et fichier de routage.

On peut aussi le faire à la main:

- créer un fichier <nom>.gjs dans fichier /templates
- ajouter une route dans router.js

08 3/Génération de nouvelle route



IHM > ember-quickstart > app > templates > index.gjs

```gjs
import { LinkTo } from '@ember/routing';

<template>
  <div class="jumbo">
    <div class="right tomster"></div>
    <h2>Bienvenue à WebMeets!</h2>
    <p>Cette page correspond à l'index</p>
    <LinkTo @route="about" class="button">About Us</LinkTo>
  </div>
</template>

```

Ember.js utilise ses propres mots-clés pour certaines fonctions comme par exemple le cas du bouton.

08

3/Génération de nouvelle route

IHM > ember-quickstart > app > JS router.js > TypeScript > Router.map() callback

```
1   import EmberRouter from '@embroider/router';
2   import config from 'ember-quickstart/config/environment';
3
4   export default class Router extends EmberRouter {
5     location = config.locationType;
6     rootURL = config.rootURL;
7   }
8
9   Router.map(function () {
10    this.route('contact', { path: '/getting-in-touch' });
11    this.route('about');
12  });
13
```

On peut aussi définir les routes dans “router.js”.

Le paramètre “path” représente bien la philosophie de Ember.

Conclusion

- Ember.js est un framework complet et structuré pour les applications web ambitieuses.
- Sa philosophie “Convention over Configuration” facilite la maintenance et le travail en équipe.
- Son écosystème intégré (CLI, Data, Glimmer, Inspector) permet un développement cohérent et efficace.
- Malgré une popularité moindre que React/Angular, Ember reste très solide pour les projets long terme.

09 Les références bibliographiques

- Guides officiels : https://guides.emberjs.com
- Documentation : https://emberjs.com
- Références API : https://api.emberjs.com
- Blog officiel : https://blog.emberjs.com
- Github : https://github.com/emberjs/ember.js
- Projets open-source : Discourse, Ghost Admin
- Wikipedia: https://en.wikipedia.org/wiki/Ember.js
- GeekForGeeks: https://www.geeksforgeeks.org/javascript/ember-js-core-concepts/
  https://www.geeksforgeeks.org/reactjs/emberjs-vs-reactjs/

Merci

de votre attention