

inertia x rails =

How to build modern monolithic SPAs 🧐

Let's start our exploration →



Why is inertia.js great?

Why is inertia.js great?

It allows you to build

Why is inertia.js great?

It allows you to build **fully client-side rendered SPA**,

Why is inertia.js great?

It allows you to build fully client-side rendered SPA, **without the complexity.**

Why is inertia.js great?

It allows you to build **fully client-side rendered SPA**, **without the complexity**.

By leveraging existing server-side patterns that you already know & love.

Why is inertia.js great?

It allows you to build **fully client-side rendered SPA**, **without the complexity**.

By leveraging existing server-side patterns that you already know & love.

- No need to build REST/GraphQL APIs

Why is inertia.js great?

It allows you to build **fully client-side rendered SPA**, **without the complexity**.

By leveraging existing server-side patterns that you already know & love.

- No need to build REST/GraphQL APIs
- No client-side routing required

Why is inertia.js great?

It allows you to build **fully client-side rendered SPA**, **without the complexity**.

By leveraging existing server-side patterns that you already know & love.

- No need to build REST/GraphQL APIs
- No client-side routing required
- No client-side state management headaches

Why is inertia.js great?

It allows you to build **fully client-side rendered SPA**, **without the complexity**.

By leveraging existing server-side patterns that you already know & love.

- No need to build REST/GraphQL APIs
- No client-side routing required
- No client-side state management headaches
- No separate authentication/authorization system

Why is inertia.js great?

It allows you to build **fully client-side rendered SPA**, **without the complexity**.

By leveraging existing server-side patterns that you already know & love.

- No need to build REST/GraphQL APIs
- No client-side routing required
- No client-side state management headaches
- No separate authentication/authorization system
- No new deployment strategy

Why is inertia.js great?

It allows you to build **fully client-side rendered SPA**, **without the complexity**.

By leveraging existing server-side patterns that you already know & love.

- No need to build REST/GraphQL APIs
- No client-side routing required
- No client-side state management headaches
- No separate authentication/authorization system
- No new deployment strategy
- Write react/vue/svelte components while sticking to rails conventions

Why is inertia.js great?

It allows you to build **fully client-side rendered SPA**, **without the complexity**.

By leveraging existing server-side patterns that you already know & love.

- No need to build REST/GraphQL APIs
- No client-side routing required
- No client-side state management headaches
- No separate authentication/authorization system
- No new deployment strategy
- Write react/vue/svelte components while sticking to rails conventions

Think of inertia.js as the glue between your Rails backend and the JS frontend of your choice (react, vue, svelte..).

Why is inertia.js great?

It allows you to build **fully client-side rendered SPA**, **without the complexity**.

By leveraging existing server-side patterns that you already know & love.

- No need to build REST/GraphQL APIs
- No client-side routing required
- No client-side state management headaches
- No separate authentication/authorization system
- No new deployment strategy
- Write react/vue/svelte components while sticking to rails conventions

Think of inertia.js as the glue between your Rails backend and the JS frontend of your choice (react, vue, svelte..).

Allowing you to leverage the **JS native ecosystem** rails is **still missing out on**.

What is inertia.js?

What is inertia.js?

A new approach to building classic server-driven web apps

What is inertia.js?

A new approach to building classic server-driven web apps

1. You build your app using Rails' existing patterns

What is inertia.js?

A new approach to building classic server-driven web apps

1. You build your app using Rails' existing patterns
 - Routes, controllers, middleware, auth, etc.
 - Everything stays the same except the view layer

What is inertia.js?

A new approach to building classic server-driven web apps

1. You build your app using Rails' existing patterns
 - Routes, controllers, middleware, auth, etc.
 - Everything stays the same except the view layer
2. Instead of ERB templates, your views are JS components

What is inertia.js?

A new approach to building classic server-driven web apps

1. You build your app using Rails' existing patterns
 - Routes, controllers, middleware, auth, etc.
 - Everything stays the same except the view layer
2. Instead of ERB templates, your views are JS components
 - React, Vue, or Svelte components
 - Full power of modern frontend frameworks

What is inertia.js?

A new approach to building classic server-driven web apps

1. You build your app using Rails' existing patterns
 - Routes, controllers, middleware, auth, etc.
 - Everything stays the same except the view layer
2. Instead of ERB templates, your views are JS components
 - React, Vue, or Svelte components
 - Full power of modern frontend frameworks
3. The magic: client-side regular routing

What is inertia.js?

A new approach to building classic server-driven web apps

1. You build your app using Rails' existing patterns
 - Routes, controllers, middleware, auth, etc.
 - Everything stays the same except the view layer
2. Instead of ERB templates, your views are JS components
 - React, Vue, or Svelte components
 - Full power of modern frontend frameworks
3. The magic: client-side regular routing
 - Click an inertia powered `<Link>` or submit a form with inertia `<Router>` → intercepted, made via XHR
 - Server returns JSON (component name + props) instead of HTML
 - inertia swaps the page component & updates browser history

What is inertia.js?

A new approach to building classic server-driven web apps

1. You build your app using Rails' existing patterns
 - Routes, controllers, middleware, auth, etc.
 - Everything stays the same except the view layer
2. Instead of ERB templates, your views are JS components
 - React, Vue, or Svelte components
 - Full power of modern frontend frameworks
3. The magic: client-side regular routing
 - Click an inertia powered `<Link>` or submit a form with inertia `<Router>` → intercepted, made via XHR
 - Server returns JSON (component name + props) instead of HTML
 - inertia swaps the page component & updates browser history

Smooth SPA experience 🚀

Under the hood – First Visit

Under the hood – First Visit

Regular **request** / response cycle rendering HTML

Under the hood – First Visit

Regular **request** / response cycle rendering HTML

```
1  REQUEST
2  GET: http://showcase.stellaire.studio/startups
3  Accept: text/html, application/xhtml+xml
```

Under the hood – First Visit

Regular **request** / response cycle rendering HTML

```
1  REQUEST
2  GET: http://showcase.stellaire.studio/startups
3  Accept: text/html, application/xhtml+xml
```

Under the hood – First Visit

Under the hood – First Visit

Regular request / **response** cycle rendering HTML

Under the hood – First Visit

Regular request / **response** cycle rendering HTML

```
1  RESPONSE
2  HTTP/1.1 200 OK
3  Content-Type: text/html; charset=utf-8
```

Under the hood – First Visit

Regular request / **response** cycle rendering HTML

```
1  RESPONSE
2  HTTP/1.1 200 OK
3  Content-Type: text/html; charset=utf-8
```

Under the hood – First Visit

Regular request / **response** cycle rendering HTML

```
1  RESPONSE
2  HTTP/1.1 200 OK
3  Content-Type: text/html; charset=utf-8
```

```
1  <html>
2    <head>
3      <title>Startups showcase</title>
4      <link href="/css/app.css" rel="stylesheet">
5      <script src="/js/app.js" defer></script>
6    </head>
7    <body>
8      <div id="app" data-page='{
9        "component": "Startups/Index",
10       "props": {
11         "featured_startups": [
12           {
13             "id": 1,
14             "name": "Astreinte Vitale",
15             "description": "Because each second counts.",
16             "preview_image_url": "/images/startups/1.jpg"
17           },
18           {
```


Under the hood – First Visit

Regular request / **response** cycle rendering HTML

```
1  RESPONSE
2  HTTP/1.1 200 OK
3  Content-Type: text/html; charset=utf-8
4
5  <div data-bbox="147 375 686 946" data-label="Text">
6      name : "Science viduals",
7      "description": "Because each second counts.",
8      "preview_image_url": "/images/startups/1.jpg"
9  },
10  {
11      "id": 2,
12      "name": "kwali",
13      "description": "The best after sale service you will ever give.",
14      "preview_image_url": "/images/startups/2.jpg"
15  }
16  ],
17  "filters": {
18      "sort": "newest"
19  }
20  },
21  "url": "/startups",
22  "version": "c32b8e4965f418ad16eaebba1d4e960f"
23  }'></div>
24  </body>
```

Under the hood – First Visit

Regular request / **response** cycle rendering HTML

```
1  RESPONSE
2  HTTP/1.1 200 OK
3  Content-Type: text/html; charset=utf-8
4
5  <div data-bbox="147 375 686 946" data-label="Text">
6      name : "Science viduals",
7      "description": "Because each second counts.",
8      "preview_image_url": "/images/startups/1.jpg"
9  },
10  {
11      "id": 2,
12      "name": "kwali",
13      "description": "The best after sale service you will ever give.",
14      "preview_image_url": "/images/startups/2.jpg"
15  }
16  ],
17  "filters": {
18      "sort": "newest"
19  }
20  },
21  "url": "/startups",
22  "version": "c32b8e4965f418ad16eaebba1d4e960f"
23  }'></div>
24  </body>
```

Under the hood – First Visit

Regular request / **response** cycle rendering HTML

[Documentation](#)

```
1  RESPONSE
2  HTTP/1.1 200 OK
3  Content-Type: text/html; charset=utf-8

14      "name": "Science vitude",
15      "description": "Because each second counts.",
16      "preview_image_url": "/images/startups/1.jpg"
17  },
18  {
19      "id": 2,
20      "name": "kwali",
21      "description": "The best after sale service you will ever give.",
22      "preview_image_url": "/images/startups/2.jpg"
23  }
24  ],
25  "filters": {
26      "sort": "newest"
27  }
28  },
29  "url": "/startups",
30  "version": "c32b8e4965f418ad16eaebba1d4e960f"
31  }'></div>
32  </body>
```

Under the hood - Subsequent Visits

Under the hood – Subsequent Visits

inertia intercepts the request and returns a **JSON** response

Under the hood – Subsequent Visits

inertia intercepts the request and returns a **JSON** response

```
1  REQUEST
2  GET: /startups/1
3  X-Inertia: true
4  X-Inertia-Version: c32b8e ...
```

Under the hood – Subsequent Visits

inertia intercepts the request and returns a **JSON** response

```
1  REQUEST
2  GET: /startups/1
3  X-Inertia: true
4  X-Inertia-Version: c32b8e ...
```

Under the hood – Subsequent Visits

inertia intercepts the request and returns a **JSON** response

```
1  REQUEST
2  GET: /startups/1
3  X-Inertia: true
4  X-Inertia-Version: c32b8e ...
```

```
1  RESPONSE
2  HTTP/1.1 200 OK
3  Content-Type: application/json
```


Under the hood – Subsequent Visits

inertia intercepts the request and returns a **JSON** response

```
1  REQUEST
2  GET: /startups/1
3  X-Inertia: true
4  X-Inertia-Version: c32b8e ...
```

```
1  RESPONSE
2  HTTP/1.1 200 OK
3  Content-Type: application/json
```

Under the hood – Subsequent Visits

inertia intercepts the request and returns a **JSON** response

```
1  REQUEST
2  GET: /startups/1
3  X-Inertia: true
4  X-Inertia-Version: c32b8e ...
```

```
1  RESPONSE
2  HTTP/1.1 200 OK
3  Content-Type: application/json
```

```
2    "component": "Startup",
3    "props": {
4      "startup": {
5        "id": 1,
6        "name": "Astreinte Vitale",
7        "description": "Because each second counts.",
8        "preview_image_url": "/images/startups/1.jpg",
9        "team_members": [ ... ]
10     }
11   },
12   "url": "/startups/1",
13   "version": "c32b8e ..."
14 }
```

Under the hood – Subsequent Visits

inertia intercepts the request and returns a **JSON** response

[Documentation](#)

```
1  REQUEST
2  GET: /startups/1
3  X-Inertia: true
4  X-Inertia-Version: c32b8e ...
```

```
1  RESPONSE
2  HTTP/1.1 200 OK
3  Content-Type: application/json
```

```
2    "component": "Startup",
3    "props": {
4      "startup": {
5        "id": 1,
6        "name": "Astreinte Vitale",
7        "description": "Because each second counts.",
8        "preview_image_url": "/images/startups/1.jpg",
9        "team_members": [ ... ]
10     }
11   },
12   "url": "/startups/1",
13   "version": "c32b8e ..."
14 }
```

Under the hood - Protocol flow

Setting up inertia in rails

Setting up inertia in rails

1. Server-side setup

Setting up inertia in rails

1. Server-side setup

```
# Gemfile
gem 'inertia_rails'
gem 'vite_rails'
```

Setting up inertia in rails

1. Server-side setup

```
# Gemfile
gem 'inertia_rails'
gem 'vite_rails'
```

```
bin/rails generate inertia:install
```


Setting up inertia in rails

1. Server-side setup

```
# Gemfile
gem 'inertia_rails'
gem 'vite_rails'
```

```
bin/rails generate inertia:install
```

This will:

Setting up inertia in rails

1. Server-side setup

```
# Gemfile
gem 'inertia_rails'
gem 'vite_rails'
```

```
bin/rails generate inertia:install
```

This will:

- Set up vite.js for asset bundling
- Configure TypeScript (optional)
- Install Tailwind CSS (optional)
- Setup the rails app to work with inertia
- Set up example route, controller & view component

Setting up inertia in rails

Setting up inertia in rails

2. Client-side setup

Setting up inertia in rails

2. Client-side setup

```
# Install Inertia's React adapter  
yarn add @inertiajs/react react react-dom
```

Setting up inertia in rails

2. Client-side setup

```
# Install Inertia's React adapter  
yarn add @inertiajs/react react react-dom
```

3. Initialize the inertia app

Setting up inertia in rails

2. Client-side setup

```
# Install Inertia's React adapter
yarn add @inertiajs/react react react-dom
```

3. Initialize the inertia app

```
1  // app/frontend/entrypoints/application.js
2  import { createInertiaApp } from '@inertiajs/react'
3  import { createRoot } from 'react-dom/client'
4
5  createInertiaApp({
6    resolve: (name) => {
7      const pages = import.meta.glob('../pages/**/*.jsx', { eager: true })
8      return pages[`../pages/${name}.jsx`]
9    },
10   setup({ el, App, props }) {
11     createRoot(el).render(<App { ... props} />)
12   },
13 })
```

Setting up inertia in rails

2. Client-side setup

```
# Install Inertia's React adapter
yarn add @inertiajs/react react react-dom
```

3. Initialize the inertia app

```
1  // app/frontend/entrypoints/application.js
2  import { createInertiaApp } from '@inertiajs/react'
3  import { createRoot } from 'react-dom/client'
4
5  createInertiaApp({
6    resolve: (name) => {
7      const pages = import.meta.glob('../pages/**/*.jsx', { eager: true })
8      return pages[`../pages/${name}.jsx`]
9    },
10   setup({ el, App, props }) {
11     createRoot(el).render(<App { ... props} />)
12   },
13 })
```


Setting up inertia in rails

2. Client-side setup

```
# Install Inertia's React adapter
yarn add @inertiajs/react react react-dom
```

3. Initialize the inertia app

```
1  // app/frontend/entrypoints/application.js
2  import { createInertiaApp } from '@inertiajs/react'
3  import { createRoot } from 'react-dom/client'
4
5  createInertiaApp({
6    resolve: (name) => {
7      const pages = import.meta.glob('../pages/**/*.jsx', { eager: true })
8      return pages[`../pages/${name}.jsx`]
9    },
10   setup({ el, App, props }) {
11     createRoot(el).render(<App {... props} />)
12   },
13 })
```

Setting up inertia in rails

2. Client-side setup

```
# Install Inertia's React adapter
yarn add @inertiajs/react react react-dom
```

3. Initialize the inertia app

```
1  // app/frontend/entrypoints/application.js
2  import { createInertiaApp } from '@inertiajs/react'
3  import { createRoot } from 'react-dom/client'
4
5  createInertiaApp({
6    resolve: (name) => {
7      const pages = import.meta.glob('../pages/**/*.jsx', { eager: true })
8      return pages[`../pages/${name}.jsx`]
9    },
10   setup({ el, App, props }) {
11     createRoot(el).render(<App { ... props} />)
12   },
13 })
```

Setting up inertia in rails

2. Client-side setup

```
# Install Inertia's React adapter
yarn add @inertiajs/react react react-dom
```

3. Initialize the inertia app

```
1  // app/frontend/entrypoints/application.js
2  import { createInertiaApp } from '@inertiajs/react'
3  import { createRoot } from 'react-dom/client'
4
5  createInertiaApp({
6    resolve: (name) => {
7      const pages = import.meta.glob('../pages/**/*.jsx', { eager: true })
8      return pages[`../pages/${name}.jsx`]
9    },
10   setup({ el, App, props }) {
11     createRoot(el).render(<App { ... props} />)
12   },
13 })
```

Layout & controllers

Layout & controllers

```
1 <%# app/views/layouts/application.html.erb %>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Startups showcase</title>
6     <%# If you want to use React add
7       `vite_react_refresh_tag` %>
8     <%= vite_client_tag %>
9     <%= vite_javascript_tag 'application' %>
10  </head>
11  <body>
12    <%= yield %>
13  </body>
14 </html>
```

Layout & controllers

```
1 <%# app/views/layouts/application.html.erb %>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Startups showcase</title>
6     <%# If you want to use React add
7       `vite_react_refresh_tag` %>
8     <%= vite_client_tag %>
9     <%= vite_javascript_tag 'application' %>
10  </head>
11  <body>
12    <%= yield %>
13  </body>
14 </html>
```

```
3 startups = Startup.
4   featured.
5   with_attached_preview_image
6
7   render inertia: 'Startups/Index', props: {
8     featured_startups: startups.as_json(
9       include: [:id, :name, :description],
10      methods: [:preview_image_url]
11    )
12  }
13 end
14
15 def show
16   startup = Startup.
17     with_attached_preview_image.
18     find(params[:id])
19
20   render inertia: 'Startups/Show', props: {
21     startup: startup.as_json(
22       include: [:team_members],
23       methods: [:preview_image_url]
24     )
25   }
```

Layout & controllers

```
1 <%= app/views/layouts/application.html.erb %>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Startups showcase</title>
6     <%= If you want to use React add
7       `vite_react_refresh_tag` %>
8     <%= vite_client_tag %>
9     <%= vite_javascript_tag 'application' %>
10  </head>
11  <body>
12    <%= yield %>
13  </body>
14 </html>
```

```
1 class StartupsController < ApplicationController
2   def index
3     startups = Startup.
4       featured.
5       with_attached_preview_image
6
7     render inertia: 'Startups/Index', props: {
8       featured_startups: startups.as_json(
9         include: [:id, :name, :description],
10        methods: [:preview_image_url]
11      )
12    }
13  end
14
15  def show
16    startup = Startup.
17      with_attached_preview_image.
18      find(params[:id])
19
20    render inertia: 'Startups/Show', props: {
21      startup: startup.as_json(
22        include: [:team_members],
23        methods: [:preview_image_url]
24      )
25    }
```

Layout & controllers

```
1 <%= app/views/layouts/application.html.erb %>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Startups showcase</title>
6     <%= If you want to use React add
7       `vite_react_refresh_tag` %>
8     <%= vite_client_tag %>
9     <%= vite_javascript_tag 'application' %>
10  </head>
11  <body>
12    <%= yield %>
13  </body>
14 </html>
```

```
1 class StartupsController < ApplicationController
2   def index
3     startups = Startup.
4       featured.
5       with_attached_preview_image
6
7     render inertia: 'Startups/Index', props: {
8       featured_startups: startups.as_json(
9         include: [:id, :name, :description],
10        methods: [:preview_image_url]
11      )
12    }
13  end
14
15  def show
16    startup = Startup.
17      with_attached_preview_image.
18      find(params[:id])
19
20    render inertia: 'Startups/Show', props: {
21      startup: startup.as_json(
22        include: [:team_members],
23        methods: [:preview_image_url]
24      )
25    }
```


Layout & controllers

```
1 <%= app/views/layouts/application.html.erb %>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Startups showcase</title>
6     <%= If you want to use React add
7       `vite_react_refresh_tag` %>
8     <%= vite_client_tag %>
9     <%= vite_javascript_tag 'application' %>
10  </head>
11  <body>
12    <%= yield %>
13  </body>
14 </html>
```

```
1 class StartupsController < ApplicationController
2   def index
3     startups = Startup.
4       featured.
5       with_attached_preview_image
6
7     render inertia: 'Startups/Index', props: {
8       featured_startups: startups.as_json(
9         include: [:id, :name, :description],
10        methods: [:preview_image_url]
11      )
12    }
13  end
14
15  def show
16    startup = Startup.
17      with_attached_preview_image.
18      find(params[:id])
19
20    render inertia: 'Startups/Show', props: {
21      startup: startup.as_json(
22        include: [:team_members],
23        methods: [:preview_image_url]
24      )
25    }
```

Layout & controllers

```
1 <%= app/views/layouts/application.html.erb %>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Startups showcase</title>
6     <%= If you want to use React add
7       `vite_react_refresh_tag` %>
8     <%= vite_client_tag %>
9     <%= vite_javascript_tag 'application' %>
10  </head>
11  <body>
12    <%= yield %>
13  </body>
14 </html>
```

```
3   startups = Startup.
4     featured.
5     with_attached_preview_image
6
7     render inertia: 'Startups/Index', props: {
8       featured_startups: startups.as_json(
9         include: [:id, :name, :description],
10        methods: [:preview_image_url]
11      )
12    }
13  end
14
15  def show
16    startup = Startup.
17      with_attached_preview_image.
18      find(params[:id])
19
20    render inertia: 'Startups/Show', props: {
21      startup: startup.as_json(
22        include: [:team_members],
23        methods: [:preview_image_url]
24      )
25    }
```

Layout & controllers

```
1 <%= app/views/layouts/application.html.erb %>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Startups showcase</title>
6     <%= If you want to use React add
7       `vite_react_refresh_tag` %>
8     <%= vite_client_tag %>
9     <%= vite_javascript_tag 'application' %>
10  </head>
11  <body>
12    <%= yield %>
13  </body>
14 </html>
```

```
3 startups = Startup.
4   featured.
5   with_attached_preview_image
6
7   render inertia: 'Startups/Index', props: {
8     featured_startups: startups.as_json(
9       include: [:id, :name, :description],
10      methods: [:preview_image_url]
11    )
12  }
13 end
14
15 def show
16   startup = Startup.
17     with_attached_preview_image.
18     find(params[:id])
19
20   render inertia: 'Startups/Show', props: {
21     startup: startup.as_json(
22       include: [:team_members],
23       methods: [:preview_image_url]
24     )
25   }
```

Layout & controllers

```
1 <%= app/views/layouts/application.html.erb %>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Startups showcase</title>
6     <%= If you want to use React add
7       `vite_react_refresh_tag` %>
8     <%= vite_client_tag %>
9     <%= vite_javascript_tag 'application' %>
10  </head>
11  <body>
12    <%= yield %>
13  </body>
14 </html>
```

```
3 startups = Startup.
4   featured.
5   with_attached_preview_image
6
7   render inertia: 'Startups/Index', props: {
8     featured_startups: startups.as_json(
9       include: [:id, :name, :description],
10      methods: [:preview_image_url]
11    )
12  }
13 end
14
15 def show
16   startup = Startup.
17     with_attached_preview_image.
18     find(params[:id])
19
20   render inertia: 'Startups/Show', props: {
21     startup: startup.as_json(
22       include: [:team_members],
23       methods: [:preview_image_url]
24     )
25   }
```

Layout & controllers

```
1 <%# app/views/layouts/application.html.erb %>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Startups showcase</title>
6     <%# If you want to use React add
7       `vite_react_refresh_tag` %>
8     <%= vite_client_tag %>
9     <%= vite_javascript_tag 'application' %>
10  </head>
11  <body>
12    <%= yield %>
13  </body>
14 </html>
```

```
3 startups = Startup.
4   featured.
5   with_attached_preview_image
6
7   render inertia: 'Startups/Index', props: {
8     featured_startups: startups.as_json(
9       include: [:id, :name, :description],
10      methods: [:preview_image_url]
11    )
12  }
13 end
14
15 def show
16   startup = Startup.
17     with_attached_preview_image.
18     find(params[:id])
19
20   render inertia: 'Startups/Show', props: {
21     startup: startup.as_json(
22       include: [:team_members],
23       methods: [:preview_image_url]
24     )
25   }
```

Frontend components

Frontend components

w/ react

Frontend components

w/ react

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { Head, Link } from '@inertiajs/react'
3
4  export default function Index({ featured_startups }) {
5    return (
6      <
7        <Head title="Our startups" />
8        <div>
9          <h1>Our startups</h1>
10
11          <div className="grid ...">
12            {featured_startups.map(startup => (
13              <Link
14                key={startup.id}
15                href={`\startups/${startup.id}`}
16              >
17                <div>
18                  <img
19                    src={startup.preview_image_url}
20                    alt={startup.title}
21
22                />
23              </div>

```


Frontend components

w/ react

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { Head, Link } from '@inertiajs/react'
3
4  export default function Index({ featured_startups }) {
5    return (
6      <
7        <Head title="Our startups" />
8        <div>
9          <h1>Our startups</h1>
10
11          <div className="grid ...">
12            {featured_startups.map(startup => (
13              <Link
14                key={startup.id}
15                href={`~/startups/${startup.id}`}
16              >
17                <div>
18                  <img
19                    src={startup.preview_image_url}
20                    alt={startup.title}
21
22                  />
23                </div>
```

Frontend components

w/ react

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { Head, Link } from '@inertiajs/react'
3
4  export default function Index({ featured_startups }) {
5    return (
6      <
7        <Head title="Our startups" />
8        <div>
9          <h1>Our startups</h1>
10
11          <div className="grid ...">
12            {featured_startups.map(startup => (
13              <Link
14                key={startup.id}
15                href={`~/startups/${startup.id}`}
16              >
17                <div>
18                  <img
19                    src={startup.preview_image_url}
20                    alt={startup.title}
21                  />
22                </div>
23              </div>

```

Frontend components

w/ react

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { Head, Link } from '@inertiajs/react'
3
4  export default function Index({ featured_startups }) {
5    return (
6      <
7        <Head title="Our startups" />
8        <div>
9          <h1>Our startups</h1>
10
11          <div className="grid ...">
12            {featured_startups.map(startup => (
13              <Link
14                key={startup.id}
15                href={`~/startups/${startup.id}`}
16              >
17                <div>
18                  <img
19                    src={startup.preview_image_url}
20                    alt={startup.title}
21                  />
22                </div>
23            )}
```

Frontend components

w/ react

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { Head, Link } from '@inertiajs/react'
3
4  export default function Index({ featured_startups }) {
5    return (
6      <
7        <Head title="Our startups" />
8        <div>
9          <h1>Our startups</h1>
10
11          <div className="grid ...">
12            {featured_startups.map(startup => (
13              <Link
14                key={startup.id}
15                href={`/${startups}/${startup.id}`}
16              >
17                <div>
18                  <img
19                    src={startup.preview_image_url}
20                    alt={startup.title}
21
22                  />
23                </div>
```

Frontend components

w/ react

```
9      <h1>Our startups</h1>
10
11      <div className="grid ...">
12        {featured_startups.map(startup => (
13          <Link
14            key={startup.id}
15            href={`/${startups}/${startup.id}`}
16          >
17            <div>
18              <img
19                src={startup.preview_image_url}
20                alt={startup.title}
21              />
22            </div>
23            <h2>{startup.title}</h2>
24          </Link>
25        ))}
26      </div>
27    </div>
28  </>
29 )
30 }
31 }
```

Frontend components

w/ react

```
9      <h1>Our startups</h1>
10
11      <div className="grid ...">
12        {featured_startups.map(startup => (
13          <Link
14            key={startup.id}
15            href={`/${startups}/${startup.id}`}
16          >
17            <div>
18              <img
19                src={startup.preview_image_url}
20                alt={startup.title}
21              />
22            </div>
23            <h2>{startup.title}</h2>
24          </Link>
25        ))}
26      </div>
27    </div>
28  </div>
29 </>
30 )
31 }
```

Frontend components

w/ react

```
9      <h1>Our startups</h1>
10
11      <div className="grid ..." >
12        {featured_startups.map(startup => (
13          <Link
14            key={startup.id}
15            href={`/${startups}/${startup.id}`}
16          >
17            <div>
18              <img
19                src={startup.preview_image_url}
20                alt={startup.title}
21              />
22            </div>
23            <h2>{startup.title}</h2>
24          </Link>
25        ))}
26      </div>
27    </div>
28  </>
29 )
30 }
31 }
```

Frontend components

w/ react

```
9      <h1>Our startups</h1>
10
11      <div className="grid ..." >
12        {featured_startups.map(startup => (
13          <Link
14            key={startup.id}
15            href={`/${startups}/${startup.id}`}
16          >
17            <div>
18              <img
19                src={startup.preview_image_url}
20                alt={startup.title}
21              />
22            </div>
23            <h2>{startup.title}</h2>
24          </Link>
25        ))}
26      </div>
27    </div>
28  </>
29 )
30 }
31 }
```


Frontend components

w/ react

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { Head, Link } from '@inertiajs/react'
3
4  export default function Index({ featured_startups }) {
5    return (
6      <
7        <Head title="Our startups" />
8        <div>
9          <h1>Our startups</h1>
10
11          <div className="grid ...">
12            {featured_startups.map(startup => (
13              <Link
14                key={startup.id}
15                href={`\startups/${startup.id}`}
16              >
17                <div>
18                  <img
19                    src={startup.preview_image_url}
20                    alt={startup.title}
21
22                />
23              </div>

```

Batteries included

Batteries included

Smart **asset versioning**

Batteries included

Smart **asset versioning**

REQUEST:

GET: /startups/1

X-Inertia: true

X-Inertia-Version: [old-version]

RESPONSE:

409 Conflict

X-Inertia-Location: /startups/1

Batteries included

Smart **asset versioning**

```
REQUEST:
GET: /startups/1
X-Inertia: true
X-Inertia-Version: [old-version]

RESPONSE:
409 Conflict
X-Inertia-Location: /startups/1
```

[Documentation](#)

Batteries included

Batteries included

Performance: **partial reloads**

Batteries included

Performance: **partial reloads**

```
REQUEST
```

```
GET: /startups
```

```
X-Inertia: true
```

```
X-Inertia-Partial-Data: startups
```

```
X-Inertia-Partial-Component: Startups
```


Batteries included

Performance: **partial reloads**

REQUEST

GET: /startups

X-Inertia: true

X-Inertia-Partial-Data: startups

X-Inertia-Partial-Component: Startups

RESPONSE

Content-Type: application/json

Batteries included

Performance: **partial reloads**

REQUEST

GET: /startups

X-Inertia: true

X-Inertia-Partial-Data: startups

X-Inertia-Partial-Component: Startups

RESPONSE

Content-Type: application/json

```
{
  "component": "Startups",
  "props": {
    "filters": [ ... ], // NOT included
    "startups": [ ... ] // included
  }
}
```

Batteries included

Performance: **partial reloads**

REQUEST

GET: /startups

X-Inertia: true

X-Inertia-Partial-Data: startups

X-Inertia-Partial-Component: Startups

RESPONSE

Content-Type: application/json

```
{
  "component": "Startups",
  "props": {
    "filters": [ ... ], // NOT included
    "startups": [ ... ] // included
  }
}
```

[Documentation](#)

Batteries included

Batteries included

More **goodies**

Batteries included

More **goodies**

- Shared data across components

Batteries included

More **goodies**

- Shared data across components
- Deferred props loading

Batteries included

More **goodies**

- Shared data across components
- Deferred props loading
- Link prefetching

Batteries included

More **goodies**

- Shared data across components
- Deferred props loading
- Link prefetching
- Classic rails html.erb views still work (perfect for devise out of the box setup)

Additional batteries

Additionnal batteries

Remember this ugly thing..?

Additional batteries

Remember this **ugly thing..?**

```
1  // app/frontend/Pages/Startups/Index.jsx
2  // ...
3  <div className="grid ..." >
4    {featured_startups.map(startup => (
5      <Link
6        key={startup.id}
7        href={`/${startups}/${startup.id}`}
8      >
9        <div>
10          <img
11            src={startup.preview_image_url}
12            alt={startup.title}
13          />
14        </div>
15        <h2>{startup.title}</h2>
16      </Link>
17    ))}
18  </div>
19  // ...
```

Additional batteries

Remember this **ugly thing..?**

```
1  // app/frontend/Pages/Startups/Index.jsx
2  // ...
3  <div className="grid ...">
4    {featured_startups.map(startup => (
5      <Link
6        key={startup.id}
7        href={`/${startups}/${startup.id}`}
8      >
9        <div>
10          <img
11            src={startup.preview_image_url}
12            alt={startup.title}
13          />
14        </div>
15        <h2>{startup.title}</h2>
16      </Link>
17    ))}
18  </div>
19  // ...
```

Additional batteries

Remember this **ugly thing..?**

```
1  // app/frontend/Pages/Startups/Index.jsx
2  // ...
3  <div className="grid ...">
4    {featured_startups.map(startup => (
5      <Link
6        key={startup.id}
7        href={`/${startups}/${startup.id}`}
8      >
9        <div>
10          <img
11            src={startup.preview_image_url}
12            alt={startup.title}
13          />
14        </div>
15        <h2>{startup.title}</h2>
16      </Link>
17    ))}
18 </div>
19 // ...
```

Additional batteries

Remember this **ugly thing..?**

```
1  // app/frontend/Pages/Startups/Index.jsx
2  // ...
3  <div className="grid ..." >
4    {featured_startups.map(startup => (
5      <Link
6        key={startup.id}
7        href={`/${startups}/${startup.id}`}
8      >
9        <div>
10          <img
11            src={startup.preview_image_url}
12            alt={startup.title}
13          />
14        </div>
15        <h2>{startup.title}</h2>
16      </Link>
17    ))}
18  </div>
19  // ...
```



Additional batteries

Additional batteries

There's a neat solution for this:

Additional batteries

There's a neat solution for this: [JS from routes](#)

Additional batteries

There's a neat solution for this: [JS from routes](#)

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { startups } from "@path_helpers";
3
4  // ...
5  <div className="grid ... ">
6    {featured_startups.map(startup => (
7      <Link
8        key={startup.id}
9        href={startups.show.path(startup)}
10      >
11        <div>
12          <img
13            src={startup.preview_image_url}
14            alt={startup.title}
15          />
16        </div>
17        <h2>{startup.title}</h2>
18      </Link>
19    )}}
20  )}}
```

Additional batteries

There's a neat solution for this: [JS from routes](#)

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { startups } from "@path_helpers";
3
4  // ...
5  <div className="grid ... ">
6    {featured_startups.map(startup => (
7      <Link
8        key={startup.id}
9        href={startups.show.path(startup)}
10      >
11        <div>
12          <img
13            src={startup.preview_image_url}
14            alt={startup.title}
15          />
16        </div>
17        <h2>{startup.title}</h2>
18      </Link>
19    ))}
20  </div>
```

Additional batteries

There's a neat solution for this: [JS from routes](#)

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { startups } from "@path_helpers";
3
4  // ...
5  <div className="grid ... ">
6    {featured_startups.map(startup => (
7      <Link
8        key={startup.id}
9        href={startups.show.path(startup)}
10      >
11        <div>
12          <img
13            src={startup.preview_image_url}
14            alt={startup.title}
15          />
16        </div>
17        <h2>{startup.title}</h2>
18      </Link>
19    ))}
20  </div>
```

Additional batteries

There's a neat solution for this: [JS from routes](#)

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { startups } from "@path_helpers";
3
4  // ...
5  <div className="grid ... ">
6    {featured_startups.map(startup => (
7      <Link
8        key={startup.id}
9        href={startups.show.path(startup)}
10      >
11        <div>
12          <img
13            src={startup.preview_image_url}
14            alt={startup.title}
15          />
16        </div>
17        <h2>{startup.title}</h2>
18      </Link>
19    ))}
20  </div>
```

Additional batteries

There's a neat solution for this: [JS from routes](#)

```
1  // app/frontend/Pages/Startups/Index.jsx
2  import { startups } from "@path_helpers";
3
4  // ...
5  <div className="grid ...">
6    {featured_startups.map(startup => (
7      <Link
8        key={startup.id}
9        href={startups.show.path(startup)}> We're home now 🏠
10     >
11       <div>
12         <img
13           src={startup.preview_image_url}
14           alt={startup.title}
15         />
16       </div>
17       <h2>{startup.title}</h2>
18     </Link>
19   )]}
20 }
```

Start living **your best frontend life**

Start living **your best frontend life**

Now you can focus on building great UIs:

Start living **your best frontend life**

Now you can focus on building great UIs:

- Leverage [shadcn/ui](#) slick & modern components

Start living **your best frontend life**

Now you can focus on building great UIs:

- Leverage [shadcn/ui](#) slick & modern components
- Setup complex frontend interactions (slidesheet, drawer, etc)

Start living **your best frontend life**

Now you can focus on building great UIs:

- Leverage [shadcn/ui](#) slick & modern components
- Setup complex frontend interactions (slidesheet, drawer, etc)
- Never write boilerplate JS yourself ever again

Start building apps **at record speed**

Start building apps **at record speed**

Focus on what matters most.

Start building apps **at record speed**

Focus on what matters most.

Building your product features,

Start building apps **at record speed**

Focus on what matters most.

Building your product features,
leveraging the full **frontend native DX & UX,**

Start building apps **at record speed**

Focus on what matters most.

Building your product features,
leveraging the full **frontend native DX & UX,**
right there within the **comfy & efficient** monolith 🚀

Start building apps **at record speed**

Focus on what matters most.

Building your product features,
leveraging the full **frontend native DX & UX,**
right there within the **comfy & efficient** monolith 🚀

Now, let's go and **build something great** 🚀

About the speaker

About the speaker

I've been a programmer by trade for nearly 10 years, but above all **I'm a tech & product builder**

About the speaker

I've been a programmer by trade for nearly 10 years, but above all **I'm a tech & product builder**
As such, when I build new things, I always aim for **speed & efficiency**.

About the speaker

I've been a programmer by trade for nearly 10 years, but above all **I'm a tech & product builder**
As such, when I build new things, I always aim for **speed & efficiency**.

Because winning with a new thing on the market always depends on how fast and efficiently the team behind it can ship it to users.

About the speaker

I've been a programmer by trade for nearly 10 years, but above all **I'm a tech & product builder**
As such, when I build new things, I always aim for **speed & efficiency**.

Because winning with a new thing on the market always depends on how fast and efficiently the team behind it can ship it to users.

I'm currently splitting my time between:

About the speaker

I've been a programmer by trade for nearly 10 years, but above all **I'm a tech & product builder**.
As such, when I build new things, I always aim for **speed & efficiency**.

Because winning with a new thing on the market always depends on how fast and efficiently the team behind it can ship it to users.

I'm currently splitting my time between:

- Building [stellaire.studio](#), a startup studio specialized in building SaaS products

About the speaker

I've been a programmer by trade for nearly 10 years, but above all **I'm a tech & product builder**.
As such, when I build new things, I always aim for **speed & efficiency**.

Because winning with a new thing on the market always depends on how fast and efficiently the team behind it can ship it to users.

I'm currently splitting my time between:

- Building [stellaire.studio](#), a startup studio specialized in building SaaS products
- Working as a freelance Product Developer / Founding Engineer

About the speaker

I've been a programmer by trade for nearly 10 years, but above all **I'm a tech & product builder**.
As such, when I build new things, I always aim for **speed & efficiency**.

Because winning with a new thing on the market always depends on how fast and efficiently the team behind it can ship it to users.

I'm currently splitting my time between:

- Building [stellaire.studio](#), a startup studio specialized in building SaaS products
- Working as a freelance Product Developer / Founding Engineer

And btw, I'm currently on the lookout for a new part-time long-term freelancing mission 🙋

About the speaker

I've been a programmer by trade for nearly 10 years, but above all **I'm a tech & product builder**.
As such, when I build new things, I always aim for **speed & efficiency**.

Because winning with a new thing on the market always depends on how fast and efficiently the team behind it can ship it to users.

I'm currently splitting my time between:

- Building [stellaire.studio](#), a startup studio specialized in building SaaS products
- Working as a freelance Product Developer / Founding Engineer

And btw, I'm currently on the lookout for a new part-time long-term freelancing mission 🙋



[about.nfilzi.com](#)

Questions?