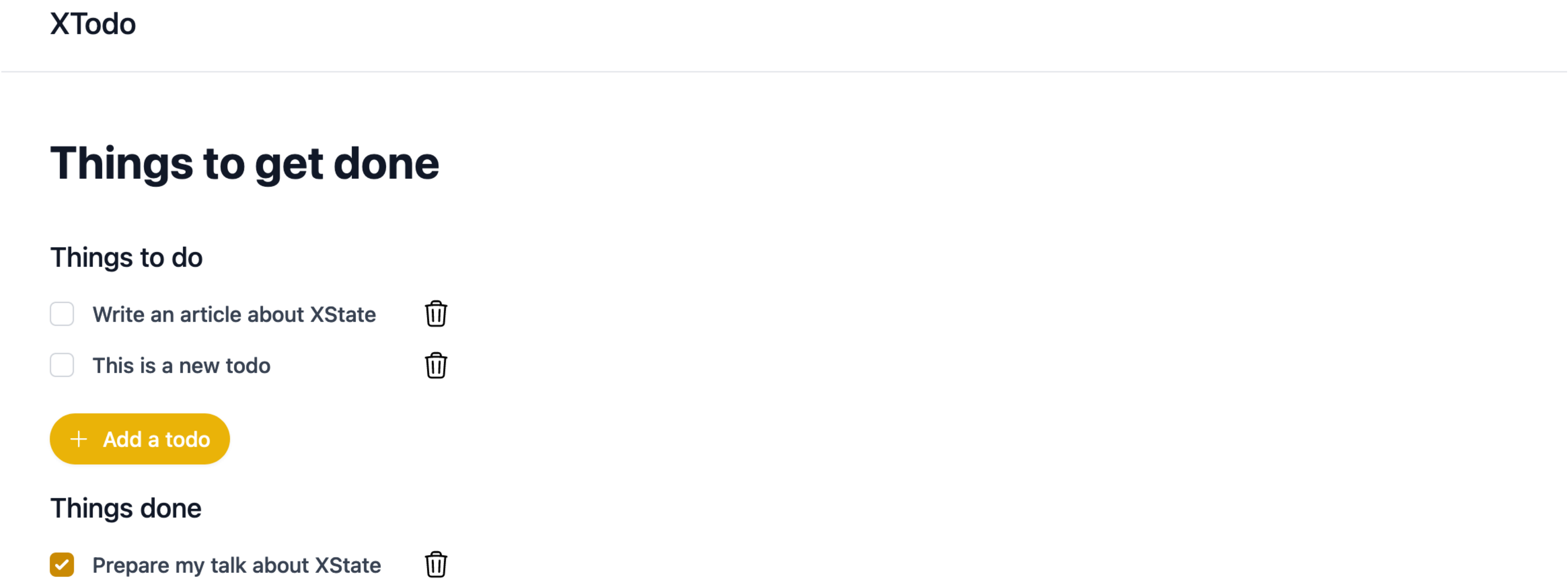
What is XState?

Is it a new version of X-Files or X-Men? 🤔

Baptiste Devessier - 2023

Today’s demo



XTodo

Things to get done

Things to do

- □ Write an article about XState
- □ This is a new todo

Things done

- ☑ Prepare my talk about XState

Structure of the app

- Data fetching (as a side effect) when page is first rendered
- Defensive code to prevent behaviors, based on booleans: Open the form if a condition is met when clicking on *Add a todo* button
- Booleans need to be synced from multiple places (error prone)
- Sometimes, a behavior is so complex to implement that the reason behind it fades out
    - Conditionally calling a debounced version of `` `synchronizeTodoList` ``

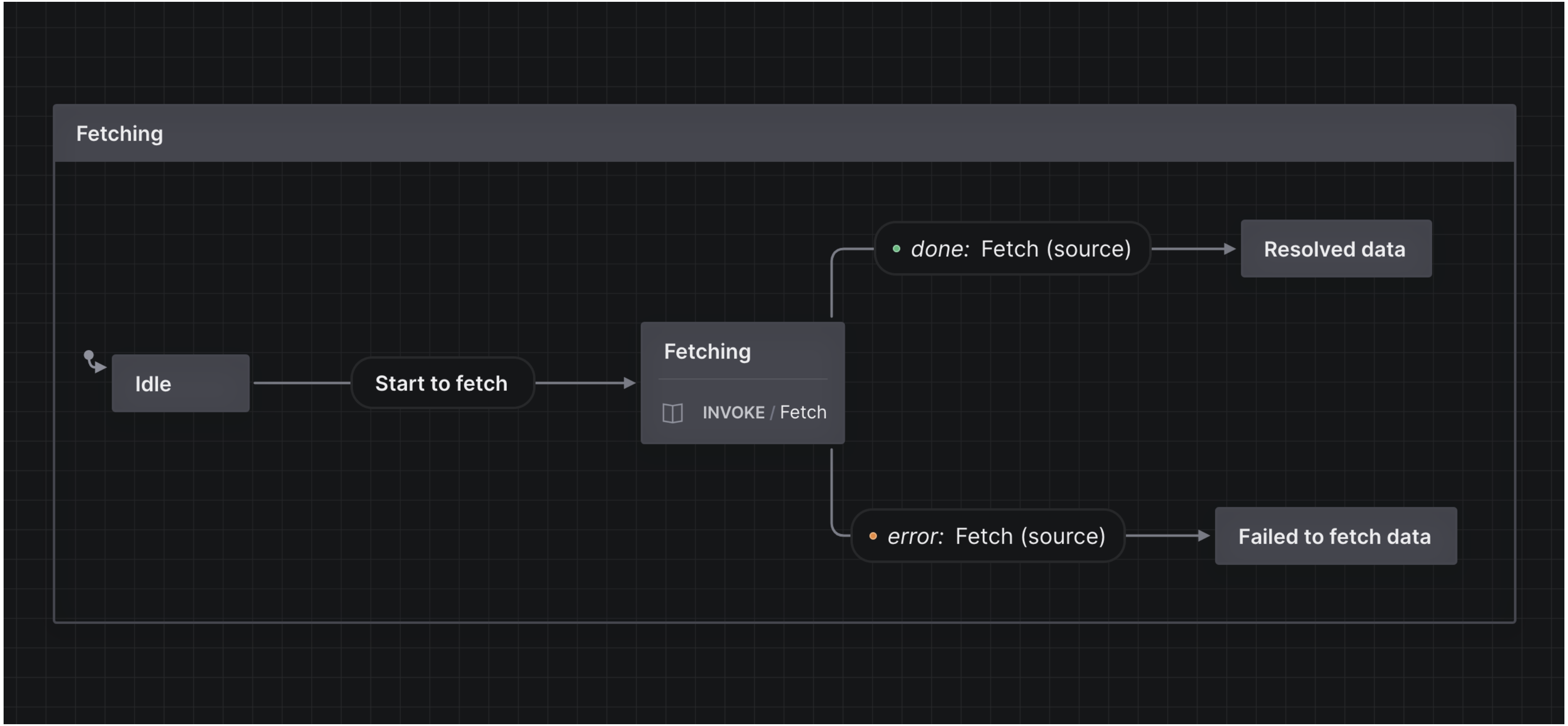
It’s time for a better world.

A world made of state machines.

What are state machines?

- Finite number of *states*
- In exactly one state at a time
- An *initial* state
- Wait for *events*
- Events trigger *transitions* from one state to another one
- Transitions are *deterministic* (same sequence of events always gives the same result)

State machine to fetch data



At the end… what is XState?

- https://stately.ai/docs (🆕 New documentation released last week!)
- Library to create state machines in JavaScript/TypeScript
- Can be used anywhere JavaScript can run (browser, Node.js, Temporal Workflows, …)
- In the frontend, it can be used with any framework, and even with no framework at all
- Stately develops visual tools for XState

Reimplement with XState

- Logic is centralized in the machine
- Event handlers only forward events – with a payload or not
- View *derives* state from the current state of the machine
- Context is used to store infinite data, like a list of todos

Pros and cons of XState

Pros

- Application logic is centralized and kept apart from implementation details
- Application logic is easier to understand, and even more with visual tools
- The logic is predictable; only what has been defined in the machine can occur
- Non-developers can understand the code – and even participate to it!

Cons

- Learning curve with statecharts, and with XState API
- Not a paradigm wildly adopted in front-end development; might take some efforts to convert a team to it

Other things XState can do

- Libraries for all front-end frameworks: @xstate/react, @xstate/vue, @xstate/svelte, @xstate/solid (coming soon)
- @xstate/inspect: Inspect a machine in real-time and interact with it
- @xstate/test: Generate tests from a state machine (aka. Model-Based Testing)
- Power a CLI built with Node.js
- In a Node.js server
- Power a Temporal Workflow
- Soon… with other languages than JavaScript

To learn more

- https://stately.ai
- https://statecharts.dev

Thank you!

Thank you Cédric and Loïck for inviting me.

- Full stack web and XState freelance consultant
- Devessier
- baptiste.devessier.fr
- BDevessier
- Code: https://github.com/Devessier/xstate-talk-demo
- Slides: https://xstate-talk.netlify.app

Link to GitHub repo 👇