Resources

> A community driven list of useful Fable tutorials, libraries and software.

Examples

Some good apps written in Fable.

- Elmish - Find web app samples in elmish repository list.
- SAFE Bookstore - Full stack SAFE example with support for deploying in a Docker container.
- SAFE-Chat - IRC-style chat demo featuring full-stack F#, Akka.Streams, Akkling, Fable, Elmish, Websockets and .NET Core.
- SAFE Nightwatch - A Demo application for React Native development in F# using Fable and the SAFE stack.
- SAFE Confplanner - A Demo application showcasing shared behaviour of CQRS/Event-Sourcing on the backend and the Elm architecture on the frontend. Both systems communication with push-notifications via websockets.
- tabula-rasa - Minimalistic real-worldish blogging platform, written entirely in F#, made as a learning reference for building large Elmish apps
- SAFE TodoList - The simplest Todo app: a client-server application written entirely in F# using Elmish on the client, Suave on the server and Fable.Remoting for type-safe communication between the two.
- Fable-Elmish-Electron-Material-UI demo - Complete boilerplate for Electron apps using Fable 2 and Elmish with hot module reloading, time-travel debugging, etc. Also demoes how to implement some non-trivial UX patterns in Elmish, as well as how to use Material-UI with JSS (styles as code).
- Volca FM editor - A Patch editor for the Korg Volca FM made with Fable-Elmish-React which uses Web MIDI
- fable-webmidi-sample - A simple sample for making a Web MIDI application with fable
- fable-uploadcare - A simple React sample to use UploadCare widget
- bulma-timepicker - A simple timepicker using Bulma in an F# React component + how to use this React component in Elmish.
- dexter - A minimal Pokemon search made with Fable, Feliz and Elmish. It demoes React components designed with Typesafe styling.
- FSharp React Starter - A starter application with examples of common architecture and testing patterns.
- F# trivia game - A trivia game written in Fable + F# Azure Functions, deployed to Azure Static Web Apps

Learn

Learn what this awesome thing is.

- Official Docs — General information and in-depth guide with examples.
- Official SAFE-Stack Docs - Official SAFE-Stack docs with nice samples and getting started tutorials.
- Official Elmish Docs - Official Elmish docs with nice samples and explanation of concepts

- Minimalistic Live Testing Fable Apps With QUnit
- F# Interop with Javascript in Fable: The Complete Guide - A comprehensive guide to Fable's interop capabilities
- Introducing Fable.Remoting: Automated Type-Safe Client-Server Communication for Fable Apps

- Statically Typed Client-Server Communication with F#: Proof of Concept - Joining F# Server and Client (outdated).
- Fable and Fable-Elmish Step-by-Step - Creating a Calculator using Fable-Elmish (**Fable 0.7**).
- Getting Started with Fable Elmish - Learn Elmish by working up to the Counter sample app from scratch (**Fable 0.7**).
- Fablelous Enterprise Tic-Tac-Toe - Web-based tic-tac-toe game written in F# and transpiled to JavaScript using Fable.
- Fable from Scratch Series - Bootstrap a Fable application from an empty directory to learn more about the stack.
- Creating Visual Planetary Systems using Fable and F#
- FableConf 2017, Elmish & Canvas based presentation - Learn how to create gorgeous Perlin based canvas animations and texts with Elmish and JS Events through a very simple example.
- Learning about the F# SAFE stack - Suave.io, Azure, Fable, Elmish - High level introduction to the SAFE stack by Scott Hanselman.
- A fable of Web MIDI - An article about how to create Fable bindings for Web MIDI
- Opinionated Fable - Architecture & Performance - Architecture & Performance tips and tricks for Fable 1.x
- Fable in React land - Fable for React: creating components and optimizing them
- Create WebComponents with Fable + Elmish + React - How to create WebComponents with Fable + Elmish + React
- Even more interop with Fable - How to use Bcrypt Js library with Fable - Dec 2018
- Fable interop 101 : generate-password - How to use generate-password Js library with Fable - Jan 2019
- Getting Started with Elmish - Walking through Elmish Counter app and explaining the MVU architecture.
- Starting with Fable (F#) - From scratch, no React, no Elmish, no Paket. Interop with DOM, p5.js explained. Also published on dev.to.

Videos and podcasts

Watch great talks about Fable

- Official Fable youtube channel
  - FableConf 2018 videos playlist - All FableConf 2018 videos
  - FableConf 2017 videos playlist - All FableConf 2017 videos
  - Fable conference talks videos playlist - Other Fable talks on Youtube
- Conquer the JavaScript Ecosystem with F# and Fable - A gentle introduction of Fable from creator *Alfonso Garcia-Caro*
- From F# to JavaScript and beyond with Fable- with Alfonso Garcia-Caro on Scott Hanselman's "Hanselminutes" podcast
- F# and the SAFE stack - with Krzysztof Cieślak on Scott Hanselman's "Hanselminutes" podcast
- Introduction to Web-Development with F# and Fable - @zaid-ajaj and @rommsen will talk about what Fable is, why it is an awesome tool to use for web development in 2019 and how you can start using it today.
- Fable + Azure Functions + Static Web Apps - A session for On .NET Live

Libraries

Useful helpers to build apps.

- Elmish - Elm-like abstractions for F# apps
- Elmish.Bridge - Create client-server Fable-Elmish apps keeping a single mindset
- Fable.Fetch - Fable bindings for Browsers' Fetch API.
- Fable.Promise - Fable bindings for JS promise.
- Fable.Date - Fable bindings for working with Dates.
- Fable.Remoting - Typed RPC client-server communication for Fable and .NET

- Fable.SignalR - A functional type-safe wrapper for SignalR and Fable.
- Fable.Extras - A more functional construct on-top of Fable.Core.
- Fable.Aether - Optics library build for Fable
- Fable.Mqtt - Fable bindings for MqttJS
- Fable.Mocha - Fable testing library with mocha. Works in browser without any dependency.
- Fable.Jester - Testing Fable apps with Jest
- Fable.Ava - Testing Fable apps with Ava
- Elmish.SweetAlert - sweetalert2 integration in Fable, implmeneted as Elmish commands, see live docs.
- Elmish.Toastr - Toastr (notification library) integration with Fable, implemented as Elmish commands
- Elmish.AnimatedTree - A tree component built on top of react-animated-tree ready to use from Elmish applications.
- Fable.SqlClient - Fable Node client for Microsoft SQL Server, built around a node-mssql binding
- Fable.React.Flatpickr - Fable binding for react-flatpickr (datetime picker component) that is ready to use within Elmish applications
- Fable.ReactAgGrid - Fable Binding for ReactAgGrid
- Fable.Parsimmon - Fable bindings for the Parsimmon parser combinator library.
- Fable.SimpleJson A library for easily working with JSON in Fable projects.
- Fable.SimpleXml A library for easily working with XML in Fable projects.
- Fable.DateFunctions - binding for the date-fns library, implemented as 120+ extension methods for DateTime.
- fable-moment-range - Fable bindings for momentjs range
- fable-react-grid-system - Fable bindings for React Grid System
- Fulma - Fable-React like DSL for Bulma + Bulma extension
- Fulma.Elmish - Ready to use *elmish components*
- fable-validation - A isomorphic validation library for F#/Fable
- Thoth.Json - Json encoder/decoder library inspire by Elm
- fable-material-ui - Fable bindings for MaterialUI
- Fable Linq - QueryBuilder for Fable
- Fable Cyclsfs - A Cycle-like Fable library for build reactive webapp
- Fable.Reaction - Extends Elmish with reactive (AsyncRx) query capabilities
- Fable.Reactstrap - Fable bindings for reactstrap.
- Feliz - A fresh retake of the base React DSL and a collection of high-quality components used to build React applications, optimized for happiness.
- Feliz.Bulma - Bulma UI wrapper for the amazing Feliz DSL.
- Feliz.Delay - Adds easy to use delayed rendering.
- Feliz.MaterialUI - Feliz-style Fable bindings for Material-UI.
- Feliz.MaterialUI.MaterialTable - Fable bindings written in the Feliz-style for `material-table`.
- Feliz.MaterialUI.Pickers - Fable bindings written in the Feliz-style for `material-ui-pickers`.
- Feliz.PigeonMaps - `pigeon-maps` bindings based on the Feliz API.
- Feliz.Plotly - Fable bindings written in the Feliz-style for plotly.js.
- Feliz.Popover - `react-popover` bindings based on the Feliz API.
- Feliz.ReactFlow - `ReactFlow` bindings based on the Feliz API.
- Feliz.Recharts - `Recharts` bindings based on the Feliz API.
- Feliz.Recoil - Fable bindings in Feliz style for Facebook's experimental state management library recoil.
- Feliz.RoughViz - Feliz binding for the `rough-viz` library.
- Feliz.Router - An Elmish router that is focused, powerful and extremely easy to use.
- Feliz.SweetAlert - Fable bindings written in the Feliz-style for sweetalert2.
- Feliz.UseDeferred - Hooks for dead-simple data fetching with Feliz.
- Feliz.UseElmish - Hooks to build Elmish components as React components.
- Feliz.UseListener - React hooks for easy event listener management.
- Feliz.UseMediaQuery - Hooks to build responsive websites.

- Feliz.UseWorker - Web workers in Fable made easy, exposed as React hooks and Elmish commands.
- Fable.GroupingPanel - A computation expression for grouping data into collapsible panels.
- Fable.AntD - Ant Design bindings for Fable
- Fss - Type-safe CSS styling library.
- office-fable - Fable bindings for office-js.

Tools

Tools around Fable platform.

- fable-loader - Fable loader for Webpack
- ts2fable - Fable parser for Typescript declaration files
- Online REPL - The Fable Online REPL
- HTML to Elmish - Convert HTML snippets into code ready to be used in Elmish apps
- JSON to Thoth - Convert JSON to F# code with Thoth decoders.
- Femto - Femto is a CLI tool that automatically resolves npm packages used by Fable bindings

Editors

Editors to code with F#.

- Ionide - A wonderful Visual Studio Code extension for F# language.
- Visual Studio
- JetBrains Rider

Templates

Fable templates to get up and running

- Elmish templates - Templates to kick start a new Elmish application. Install them like `dotnet new -i "Fable.Template.Elmish.React::*"` and create a project with `dotnet new fable-elmish-react -n myproject`
- SAFE template - Dotnet CLI template to bootstrap SAFE projects, including Suave.IOon server side
- Fulma minimal template - The quickest way to get started an Elmish + Fulma application from scratch
- Fable.Library.Template - F# Template for create and publishing Fable Libraries
- Semuserable.Fable.Templates - Minimal Fable templates
- Fable for Azure Static Web Apps - GitHub repo template for Fable + Functions for Azure Static Web Apps
- Fable + Feliz for Azure Static Web Apps - GitHub repo template for Feliz + Functions for Azure Static Web Apps
- Elmish for Azure Static Web Apps - GitHub repo template for Elmish + Functions for Azure Static Web Apps

Old (working only before Fable3)

- Fable.Jest - Testing Fable apps with Jest
- fable-signalr - Fable bindings for SignalR
- fable-websockets - Archived - Fable bindings for WS
- fable-react-toolbox - UI components for fable-react
- fable-momentjs - Fable bindings for momentjs
- Fable.Import.WebMIDI - Web MIDI bindings for Fable
- rollup-plugin-fable - Fable plugin for Rollup

Support

Where to find help.

- Gitter - Ask questions on fable gitter.
- Slack - Join Official FSharp.org and Slack channel.

Built with Fable

Production application that built with Fable

- ionide - VS Code and Atom extension for F# development
- The Gamma - Tools for open data-driven storytelling
- Casque Noir - Web documentary about Haïti environmental issues
- Fable-of-the-Day - Catch of the day by @wesbos ported to Fable
- Czech Republic 2018 Election Analytics
- Metadata Excel-Add-in: Swate