Elm + PostgREST

The declarative duo

Elm

- Very simple functional language that compiles to JS
- Haskell heritage
- Language + Libraries + Architecture
- Focus on usability

PostgREST

- Declarative API
- Maps a database schema to a “REST-like” API.
- Also very simple tool, little room for things to go wrong.

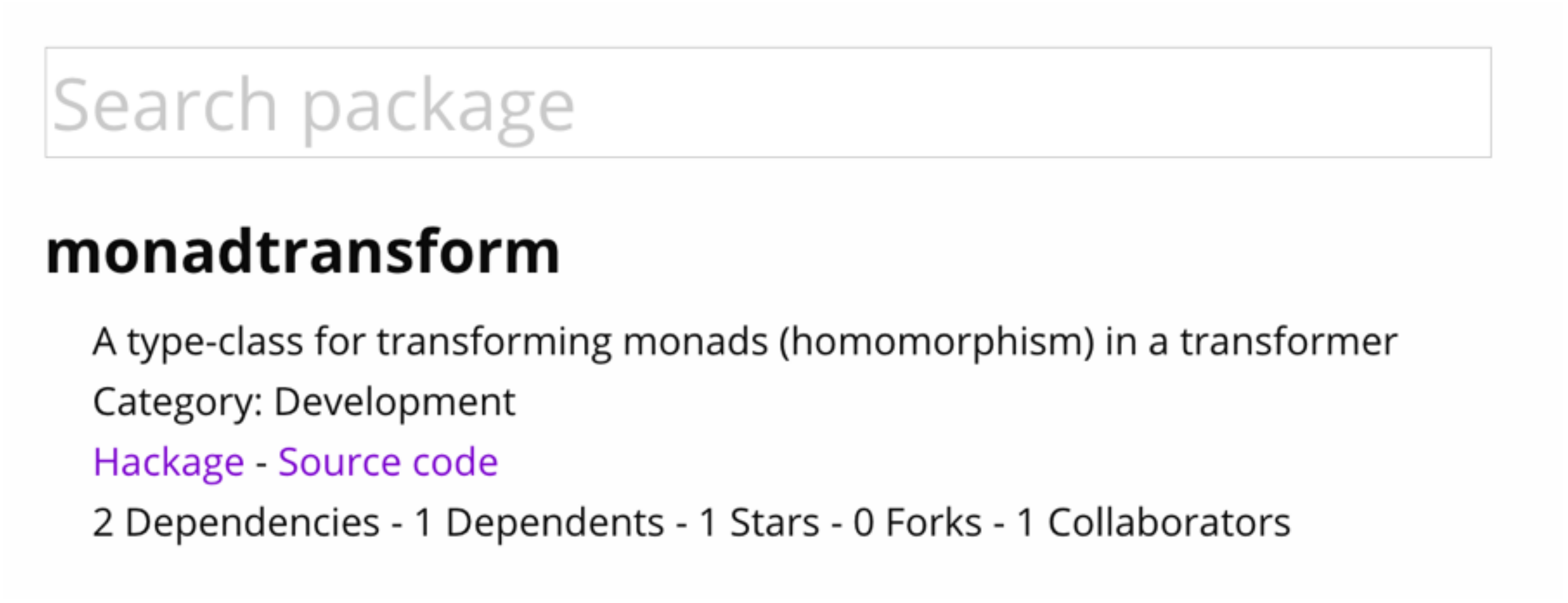
A good match

Our application

- Package search
  - We will use package data in a PostgreSQL schema.
  - The visitor wants to find a Haskell package by typing parts of it’s name or description.
  - The application will show all packages that match the keywords ordering by relevance

Planning

- Package search
  - We will use package data in a PostgreSQL schema.
  - The visitor wants to find a Haskell package by typing parts of it’s name or description.
  - The application will show all packages that match the keywords ordering by relevance



Search package

monadtransform

A type-class for transforming monads (homomorphism) in a transformer

Category: Development

Hackage - Source code

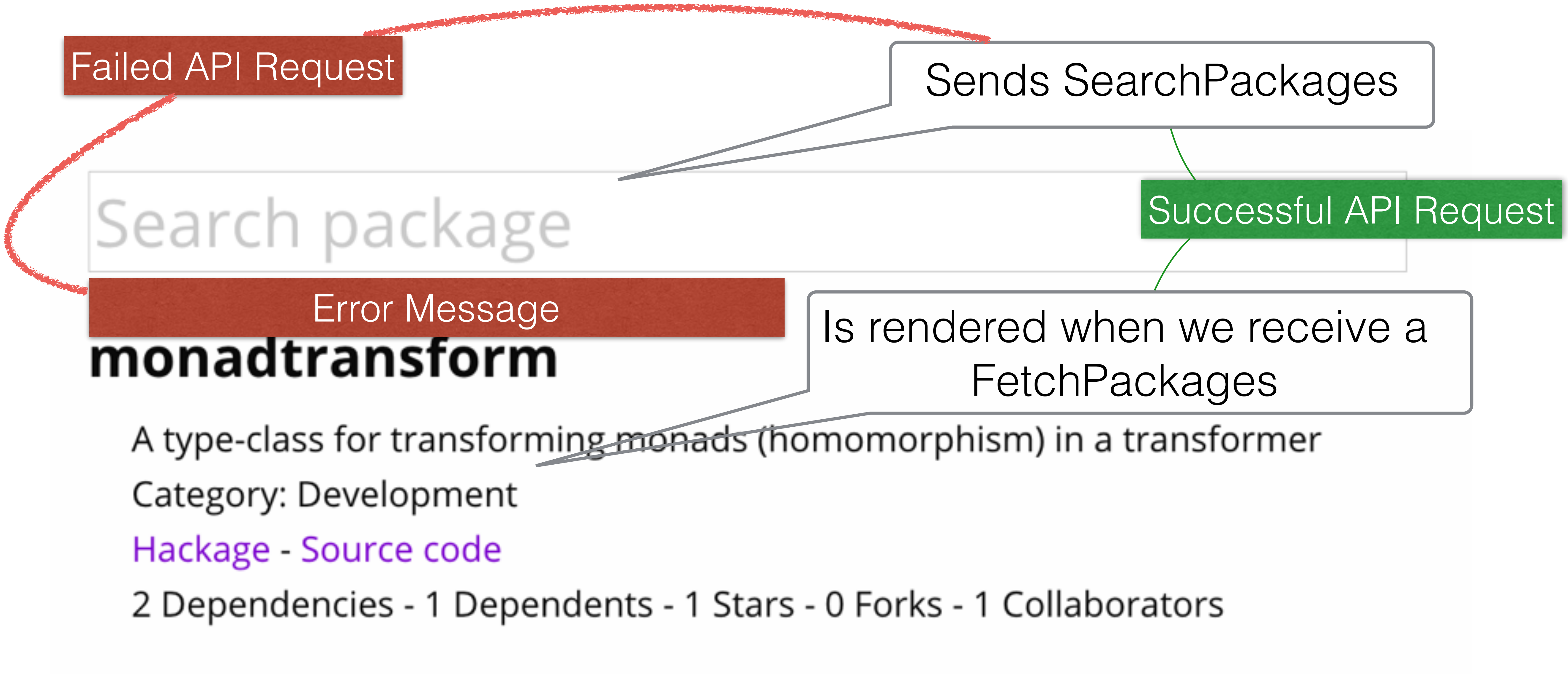
2 Dependencies - 1 Dependents - 1 Stars - 0 Forks - 1 Collaborators

Wireframe

1 view with 3 components: search box, list of packages and a package entry.

The Elm Architecture

- Init - Initial state (Model and Side effects)
- View - Template to render Model
- Update - Reducer that will take a Msg and a Model and produce a new Model (and possibly some side-effects)
- Subscriptions - Events triggered outside of my code



Search package

monadtransform

A type-class for transforming monads (homomorphism) in a transformer

Category: Development

Hackage - Source code

2 Dependencies - 1 Dependents - 1 Stars - 0 Forks - 1 Collaborators

Wiring the view

Search box -> SearchPackages String -> Api Request -> FetchPackages Result