Почему Elm?

@maxfarseer

Предыстория

- Компания из Мюнхена прислала 2 тестовых задания
  - первое на React
  - второе на Elm

Во время испытательного срока, я заметил что:

- Elm приложения
  - требуют меньше времени на поддержку, багов “кот наплакал” 😿
  - имеют “представления” для всех возможных вариантов
  - требуют от меня новых знаний, чтобы добавлять новый функционал

CutCut - сервис создания стикеров

- фон удаляется автоматически (используется сервис remove.bg)
- для работы с изображением используются библиотеки: cropper.js и fabric.js
- везде где это возможно, используется Elm

Что же такое Elm?

- Elm, это функциональный язык программирования.

```
sum : Number -> Number -> Number
sum a b =
  a + b

setNews : News -> State -> State
steNews data state =
  { state | news = data }
```

- Если ваш код “собрался”, он работает. В elm нет runtime ошибок
- В Elm нет null и undefined, а также возможности указать any

В Elm можно указать свой тип данных

```
type alias Model =
    { step : Step
    , removeBgApiKey : RemoveBgApiKey
    }


type Step
    = Add
    | Crop
    | RemoveBgOrNot UploadStatus Base64ImgUrl
    | Erase
```

Elm

```
exposed | 4 references
251  view : Model -> Html Msg
252  view model =
253      case model.step of
254          270|#>#            Ui.Modal.view
255          271|#>#                { title = "Add image: Background"
256          272|#>#                , open = True
257          273|#>#                , closeMsg = ClickedCloseModal
258          274|#>#                , confirmMsg = Nothing
259          275|#>#                , confirmText = Nothing
260          276|#>#                }
261          277|#>#                []
262          278|#>#                [ viewRemoveBgQuestion status imgUrl
263          279|#>#                ]
264
265          Missing possibilities include:
266
267              #Erase#
268
269          RemoveBgOrNot status imgUrl ->
270              Ui.Modal.view
271                  { title = "Add image: Background"
272                  , open = True
273                  , closeMsg = ClickedCloseModal
274                  , confirmMsg = Nothing
275                  , confirmText = Nothing
276                  }
277                  []
278                  [ viewRemoveBgQuestion status imgUrl
279                  ]
280
```

Elm +

- для р … ыдавать резул …
  - у …
  - о …
- если … риложение не уп …
- (BON … кэнде

```
int : Decoder Int
```

Decode a JSON number into an Elm `Int`.

```
decodeString int "true"                == Err ...
decodeString int "42"                  == Ok 42
decodeString int "3.14"                == Err ...
decodeString int "\"hello\""           == Err ...
decodeString int "{ \"hello\": 42 }" == Err ...
```

```
float : Decoder Float
```

Decode a JSON number into an Elm `Float`.

```
decodeString float "true"                == Err ..
decodeString float "42"                  == Ok 42
decodeString float "3.14"                == Ok 3.14
decodeString float "\"hello\""           == Err ...
decodeString float "{ \"hello\": 42 }" == Err ...
```

Компилятор помогает закончить “фичу”

- Типичное react-redux приложение, требует для чего-то нового:
  - action
  - reducer
  - изменение во view

- Если вы что-то забыли, компилятор в Elm подскажет

Пак

List

You can create a `List` in Elm with the `[1,2,3]` syntax, so lists are used all over the place. This module has a bunch of functions to help you work with them!

Create

```
singleton : a -> List a
```

Create a list with only one element:

```
singleton 1234 == [1234]
singleton "hi" == ["hi"]
```

```
repeat : Int -> a -> List a
```

Create a list with *n* copies of a value:

```
repeat 3 (0,0) == [(0,0),(0,0),(0,0)]
```

README
About
Source

Modules

Search

Array
Basics
Bitwise
Char
Debug
Dict
List
Maybe
Platform
Platform.Cmd
Platform.Sub
Process
Result
Set
String
Task
Tuple

Про документацию для пакета...

- Как оформлять документацию (link)

```
96  {-| Determine if a string is empty.
97
98      isEmpty "" == True
99      isEmpty "the world" == False
100 -}
101 isEmpty : String -> Bool
102 isEmpty string =
103   string == ""
```

```
isEmpty : String -> Bool
```

Determine if a string is empty.

```
isEmpty "" == True
isEmpty "the world" == False
```

Пак

elm packages — ChristophP / elm-i18next / 4.1.2

I18Next

This library provides a solution to load and display translations in your app. It allows you to load json translation files, display the text and interpolate placeholders. There is also support for fallback languages if needed.

Types and Data

```
type Translations
```

A type that represents your loaded translations

```
type Delims
    = Curly
    | Underscore
    | Custom ( String, String )
```

A union type for representing delimiters for placeholders. Most commonly those will be `{{...}}`, or `__...__`. You can also provide a set of custom delimiters(start and end) to account for different types of placeholders.

README
About
Source

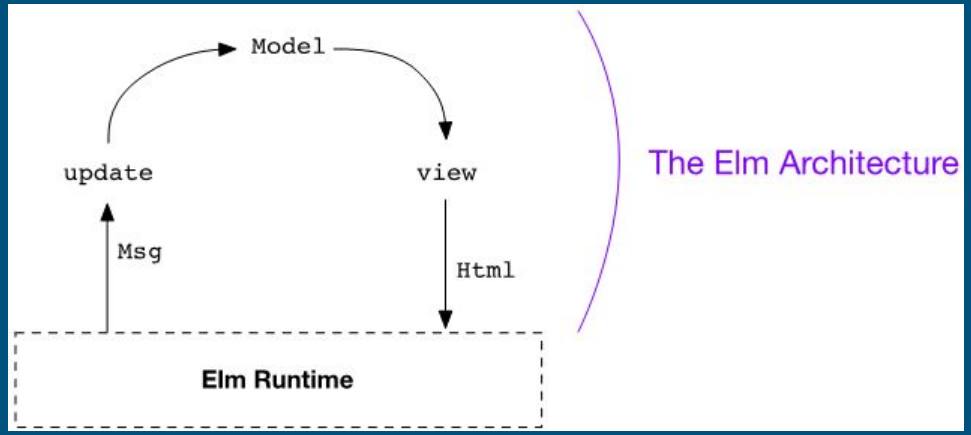
Modules

Search

I18Next

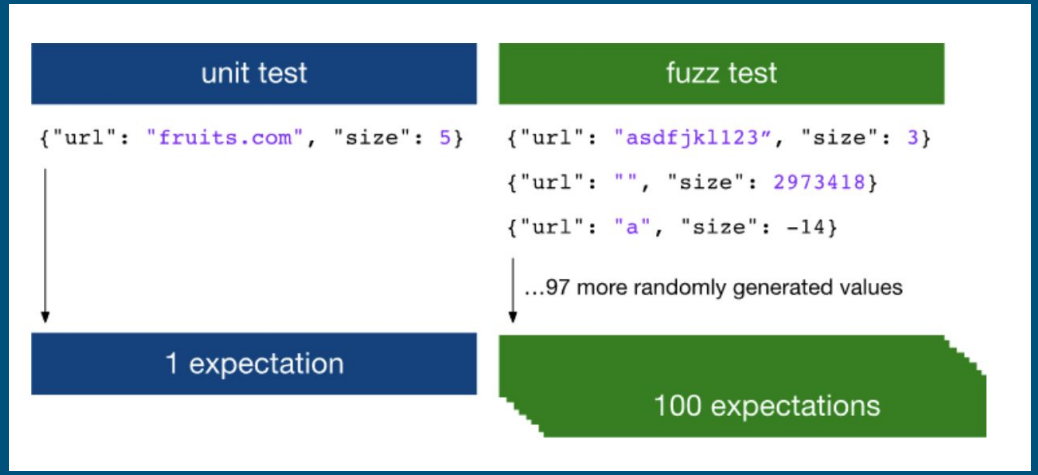
Elm - это язык и фреймворк

- У вас есть единый подход к обновлению данных в приложении



Тестирование

- Чистые функции без side effects легко тестируются
- Fuzz тесты



elm-css

```
{-| A logo image, with inline styles that change
-}
logo : Html msg
logo =
    img
        [ src "logo.png"
        , css
            [ display inlineBlock
            , padding (px 20)
            , border3 (px 5) solid (rgb 120 120 1
            , hover
                [ borderColor theme.primary
                , borderRadius (px 10)
                ]
            ]
        ]
        []
```

elm-ui

```
main =
    Element.layout []
        myRowOfStuff

myRowOfStuff : Element msg
myRowOfStuff =
    row [ width fill, centerY, spacing 30 ]
        [ myElement
        , myElement
        , el [ alignRight ] myElement
        ]


myElement : Element msg
myElement =
    el
        [ Background.color (rgb255 240 0 245)
        , Font.color (rgb255 255 255 255)
        , Border.rounded 3
        , padding 30
        ]
        (text "stylish!")
```

Инструменты

- песочница Ellie
- elm-live (также live-reload из коробки работает в parcel), для webpack нужно добавить loader
- поддерживается множество редакторов
- elm-format для автоформатирования кода
- html to elm
- json to elm

Поговорим о минусах 😕

- Нет возможности писать код для сервера
- Нет внятного roadmap (mobile? SSR? когда новая версия?)
- Гарантии для приложения уменьшаются на этапе интеграции с javascript
- “Начальник” языка (Evan Czaplicki)
- Количество вакансий

Плюсы и минусы

Плюсы:

- Гарантии в elm коде
- Легко рефакторить, богоподобный компилятор
- Интересно, новые знания
- Скорость сборки и размер бандла

Минусы:

- Ports (если общаемся с js, много кода для отображения UI, теряем некоторые гарантии)
- Нет roadmap
- Своеобразный “начальник” языка
- Количество вакансий

- Сообщество (community)

Что дальше?

- Посмотрите видео “Знакомимся с Elm” (конспект)
- Посетите официальный сайт elm-lang.org
- Попробуйте поиграться в песочнице Ellie
- Подписывайтесь на телеграм, где я рассказываю про Elm
- Посмотрите видео + презентацию сервиса CutCut (github)
- Посмотрите, кто использует Elm (github)
- Присоединяйтесь к коммьюнити (также есть ру чат в телеграм)

Разрабатывайте с наслаждением 😜

- Контакты
  - Github
  - Twitter
  - YouTube
  - Telegram @maxfarseer

Спасибо, за внимание!