Как я начал любить Vue

Минин Игорь, T-Systems 2018

LIFE IS FOR SHARING.

Короткий рекап статьи по Ангуляру

https://medium.com/@igogrek/how-i-stopped-loving-angular-c2935f7378c4

- Ужасный роутер
- Тяжеловесное и бесполезное Dependency Injection, модули
- Тонны лишних абстракций

Оговорка по DI - Юнит тестирование

- DI в Angular удобно для мокирования внешних зависомстей
- Во Vue менее удобно

Пару слов о Реакте и почему не

Все ОЧЕНЬ субъективно

- Реактивность во Vue просто работает: ~~иммутабельность~~, ~~чистые функции~~
- Концепция крутая, но многословная - долго/дорого
- JSX плох
  - Медленная миграция HTML - дизайнерам плохо
  - Преждевременная модулизация VS шаблон в Angular/Vue – выносишь по необходимости
  - Кондишены = боль
- Формы адов  отри далее
- create-react-a

Отличия

Документация

- В Angular кривая и непонятная

  Banana in a box

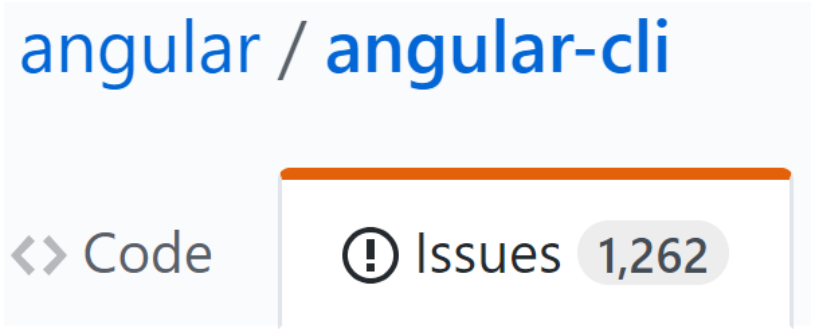
- В Vue простая и доходчивая

  На **6** языках, хотя хватило бы и английской

CLI

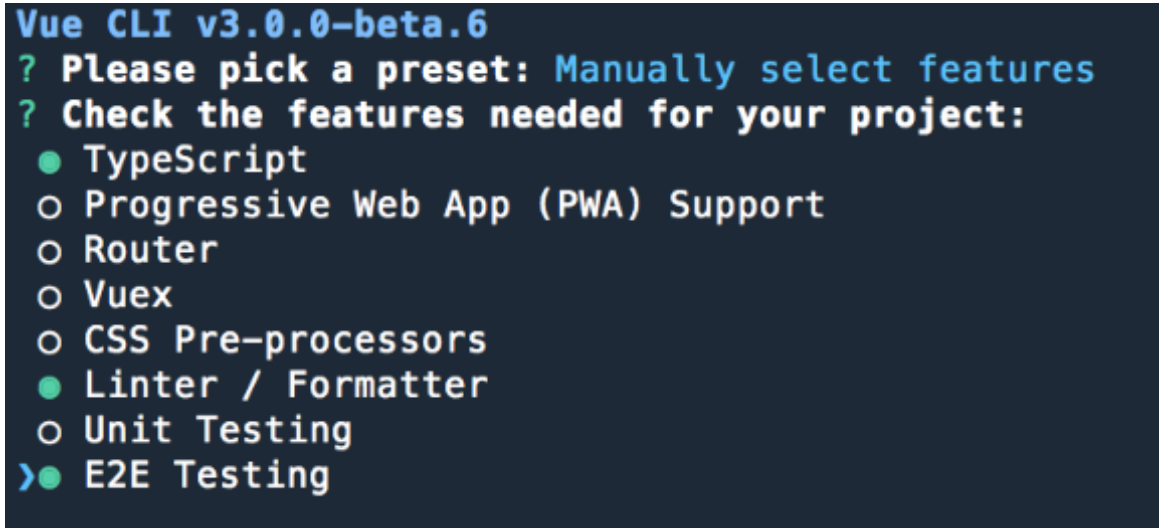
Раньше я думал что Angular CLI крутое, но:

- Сложно кастомизировать
- Множество пакетов и неконсистентно работает с версиями
- Боль при апгрейде



Vue CLI 3 – самое крутое CLI на сегодня

- Кастомизация
- Простота и гибкость



Zone.js vs Vue Reactivity

Angular - Zone.js

- Monkey-патчит стандартные API для отслеживания
- Сложности с нестандартными АПИ
- Ужасные стектрейсы
- Хак?

Vue не нужен Zone.js, отслеживание работатет благодаря Observer

- Нативный Object.defineProperty
- Есть подводные камни, но они описаны и легки для понимания
- 2.6 не будет их иметь – переход н

Во Vue нельзя динамически добавлять новые корневые реактивные свойства в уже существующий экземпляр. Тем не менее, можно добавить реактивное свойство во вложенные объекты, используя метод `Vue.set(object, key, value)`:

JS

```
Vue.set(vm.someObject, 'b', 2)
```

RxJS vs state MANAgement

Нужен ли RxJS?

Есть **vue-rx**

Есть Vuex + вагон альтернатив:

State Management

- vuex - Centralized State Management for Vue.js.
- vue-sync - Synchronize Vue State with the Browser URL, Server Backend, and other endpoints.
- vuelm - Lightweight state management inspired by Elm architecture.
- vue-duo - A tiny state management for Vue.js.
- vuez - A simple but powerful State Management for Vue.js, with only 2 APIs.
- vuet - Vue.js state management model for Agile Development
- vue-ya-stash - Yet Another stash storage with update/patch event emitters simillar with v-bind.sync
- vue-assign-model - Automatically assign elements value to model for Vue.js.
- vue-stash - Easily share reactive data between your Vue components.
- vue-mc - Models and Collections for Vue.js
- vue-model - Model component for Vue.js
- tuex - A mostly reasonable alternative to Vuex.
- revuejs - A tiny, light and handy state management for vuejs 2, writing less verbose code.
- lue - Vue and vuex based library, writing less verbose code.
- vuex-lite - A 1KB Vuex with just state and mutations.
- Vuenut - is a component to develop faster and more fluently.
- vuex-pathify - ridiculously simple vuex wiring + setup

Просто добавь ...

- ~~Воды~~
- RxJS
- Vuex
- TypeScript
- SCSS

  …

Роутер

В Angular **три** раза переписанный и все равно ужасен

- **Нет** именованных роутов (!!!)
- Странная система ивентов, параметры = Observable
- **Команды** для навигации
- Lazy Loading через строки

...

В Vue простой и рабочий

- Нет абстрактных роутов, плохо ли?
- Классная штука которая просто ра

```
path: `url/sub-url/:id',
component: MyComponent,
props: true
```

Расширение и переиспользование

В Angular компоненты – классы

НО сложно использоваать фишки TypeScript:

абстрактные классы, области видимости (DI, проблемы с AOT), …

Во Vue - объекты –просто работать и расширять/рефакторить

Для переиспользования кода есть Mixin’ы и Plugin’ы

UI компоненты

- Быстрые и простые в Vue
- **Огмроный** выбор – Element, Vuetify, Quasar, Vue-Material, Muse, iView, etc etc.

- Сложные и не всегда

быстрые в Angular

- Мало – материал, кларити

Реальный опыт

СЛИШКОМ МНОГО СВОБОДЫ

- Множество способов делать одно и то же
  - Объявление компонентов, в т.ч. глобальные
  - Примеси, в т.ч. глобальные
  - Plugin’ы

    ...
- Сложно заставить людей не использовать data() вместо state
- Нет четких паттернов для “сервисов”

TypeScript

TypeScript крут, но его надо уметь готовить

Angular – TS = тонны абстракций

Vue –TS = расширение возможностей JS

Проблемы с TypeScript в Vue

Так и не приняли мой Pull Request =(((

Два разных подхода к использованию TypeScript

Не нравтся class-component, мы используем Vue.extend:

- У class-component свои хитрости / недостатки
- Хочется меньше уходить от ES6 Vue компонентов
- Хочется чтобы команда понимала как работает Vue без TypeScript

  Особенно после Angular

Проблемы с TypeScript в Vue

- Tslint плохо работает с Vue
- Eslint для TS еще не супер, непонятные ошибки
- Vuex store не типизирован снаружи
- Vue.extend() имеет свои минусы:
  - Нельзя использовать как mixins, приватные методы и тд.
  - Боль с Vuex …mapWhatever()
  - Типизация data() неудобна
  - Типизация props практически невозможна

Формы с Vuex

Проблема не только Vuex а почти любого State

```
<input :value="message" @input="updateMessage">

...

computed: {
  ...mapState({
    message: state => state.obj.message
  })
},
methods: {
  updateMessage (e) {
    this.$store.commit('updateMessage', e.target.value)
  }
}
```

```
<input v-model="message">

...

computed: {
  message: {
    get () {
      return this.$store.state.obj.message
    },
    set (value) {
      this.$store.commit('updateMessage', value)
    }
  }
}
```

Формы с Vuex

+ Беда с mapGetters, мы решили:

```
static mapTwoWay<T>(getter: string, mutation: string) {
  return {
    get(this: Vue): T {
      return this.$store.getters[getter];
    },
    set(this: Vue, value: T) {
      this.$store.commit(mutation, value);
    }
  };
}
```

```
stringTags: Util.mapTwoWay<IDatasetExtra[]>(STRING_TAGS, UPDATE_STRING_TAGS)
```

```
v-model="stringTags"
```

Валидация форм

Из коробки - нет

Схожий механизм с template-driven form в Angular - **vee-validate**

Vuelidate супер!

```
<input v-validate="'required|email'">
```

```
<input v-model="name" @input="$v.name.$touch()">
```

```
import { required, email } from 'vuelidate/lib/validators'

export default {
  data () {
    return {
      name: ''
    }
  },
  validations: {
    name: {
      required,
      email
    }
  }
}
```

Основная фишка/проблема Vue

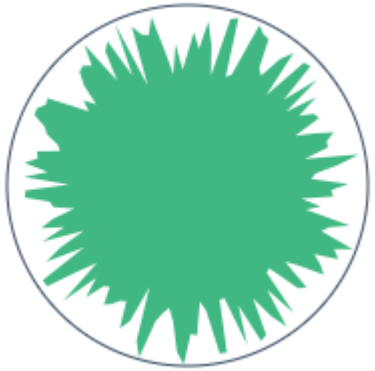
Это библиотека, нет из коробки:

- HTTP
- Валидации
- i18n
- ..?

Официальные:

- Router
- State

При этом есть такие крутые встроенные штуки как **Ar**
???



Nightwatch и e2e

Из коробки выбор между Nightwatch и Cypress

У Nightwatch проблема с URL’ами содердащими хеши

Решение:

```
.url('data:,')
.url(client.globals.devServerURL + `/#/my-hashed-url`)
```

Выводы

Скорость разработки

Куча лишнего boilerplate в Angular – долго и медленно

VS

Минимализм в Vue – быстро и просто

Гигантское комьюнити

Чего нет

- React Native крутой

VS

- Weex пока еще развивается
- NativeScript Vue - пока гораздо менее мощный тулсет

производительность

http://www.stefankrause.net/js-frameworks-benchmark7/table.html

Duration in milliseconds

Name	vue-v2.5.3-keyed	react-v16.1.0-keyed	angular-v5.0.0-keyed
remove row Duration to remove a row. (with 5 warmup iterations).	52.5 ± 1.8 (1.1)	51.5 ± 2.0 (1.1)	46.1 ± 2.6 (1.0)
create many rows Duration to create 10,000 rows	1521.4 ± 55.7 (1.0)	2033.7 ± 32.0 (1.3)	1682.0 ± 53.1 (1.1)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows.	338.4 ± 10.3 (1.3)	271.8 ± 9.9 (1.1)	257.6 ± 11.1 (1.0)
clear rows Duration to clear the table filled with 10.000 rows.	240.9 ± 11.4 (1.1)	224.4 ± 6.0 (1.0)	360.3 ± 16.4 (1.6)
startup time Time for loading, parsing and starting up	48.4 ± 2.4 (1.0)	49.4 ± 0.7 (1.0)	88.8 ± 2.9 (1.8)
slowdown geometric mean	1.10	1.10	1.27

Memory allocation in MBs

Name	vue-v2.5.3-keyed	react-v16.1.0-keyed	angular-v5.0.0-keyed
ready memory Memory usage after page load.	3.6 ± 0.1 (1.0)	3.7 ± 0.1 (1.0)	6.7 ± 0.1 (1.9)
run memory Memory usage after adding 1000 rows.	7.2 ± 0.0 (1.0)	7.6 ± 0.0 (1.0)	10.5 ± 0.0 (1.5)
update eatch 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	7.3 ± 0.0 (1.0)	8.5 ± 0.0 (1.2)	10.6 ± 0.0 (1.5)
replace 1k rows (5 cycles) Memory usage after clicking create 1000 rows 5 times	7.3 ± 0.0 (1.0)	9.0 ± 0.0 (1.2)	10.8 ± 0.0 (1.5)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	3.8 ± 0.0 (1.0)	4.7 ± 0.0 (1.2)	7.1 ± 0.0 (1.9)

Пару слов от более авторитетных товарищей

- GitLab - https://about.gitlab.com/2016/10/20/why-we-chose-vue/
- CodeShip - https://blog.codeship.com/consider-vuejs-next-web-project/
- Alibaba, Xiaomi - https://www.netguru.co/blog/13-top-companies-that-have-trusted-vue.js-examples-of-applications

Спасибо, вопросы?

Минин Игорь, T-Systems, 2018

igogrek@gmail.com