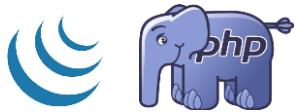
API Design

как сделать хорошо и не делать плохо

DINS®

О себе

В прошлом – PHP разработчик



Ушёл на фронт – с 2015

Vue.js → React

Senior Frontend Developer @ DINS

О чём речь

- API is …
- Application Programming Interface
- Interface or Communication Protocol
- Процедуры и методы, формат данных, взаимодействие

Интерфейс и порядок взаимодействия

```
const el = document.createElement('div');

el.innerText = 'Hello World';

document.body.appendChild(el);
```

HTMLElement

```
{
   innerText: …
   children: …
   attributes: …
   classList: …
}
```

С чем мы сталкиваемся

Browser API (DOM API, IndexedDB, WebGL)

Frameworks (React, Vue, Angular)

Libraries (Lodash, Redux, Std.library)

Own (project) utils

Own (project) components

Web Components

```
class PopUpInfo extends HTMLElement {
    constructor() {
        super();
        var shadow = this.attachShadow({mode: 'open'});
        var wrapper = document.createElement('span');
        wrapper.setAttribute('class','wrapper');
        var icon = document.createElement('span');
        icon.setAttribute('class','icon');
        icon.setAttribute('tabindex', 0);
        var info = document.createElement('span');
        info.setAttribute('class','info');
        var text = this.getAttribute('text');
        info.textContent = text;
        var imgUrl = this.hasAttribute('img')
            ? this.getAttribute('img')
            : 'img/default.png';
        var img = document.createElement('img');
        img.src = imgUrl;
        icon.appendChild(img);
        shadow.appendChild(wrapper);
        wrapper.appendChild(icon);
        wrapper.appendChild(info);
    }
}

customElements.define('popup-info', PopUpInfo);
/*
<popup-info img="img/alt.png" text="Your card validation
code"></popup-info>
 */
```

React

```
const PopupInfo = ({ img = 'img/default.png', text }) => (
  <span className='wrapper'>
    <span className="icon" tabIndex={0}>
      <img src={img}/>
    </span>
    <span className="info">{text}</span>
  </span>
);
```

API - различаются

Какой API хороший?

Простой

Гибкий

Консистентный

Уместный

С хорошим DX

Простой API

```
export const someApi = {
    subscribe(fn) {/* ... */},
    getState() {/* ... */},
    dispatch(a) {/* ... */},
};
```

Простой API

```
const someOtherApi = {
    commit(t, p = undefined, o = {}) {/* ... */},
    dispatch(t, p = undefined, o = {}) {/* ... */},
    watch(fn, cb, o = {}) {/* ... */},
    subscribe(fn) {/* ... */},
    subscribeAction(fn) {/* ... */},
    registerModule(p, m, o = {}) {/* ... */},
    unregisterModule(p) {/* ... */},
};
```

Простой API

Простой vs Лёгкий

“Simple Made Easy” Rich Hickey

https://github.com/matthiasn/talk-transcripts/blob/master/Hickey_Rich/SimpleMadeEasy.md

Callback vs Promise

Простой API

KISS

YAGNI

Гибкий API

REST API

GET /users/1/dashboard

POST /users/1/dashboard

gRPC

```
const dashboard = client.getUserDashboard({ user: 1 }, (err, resp) => {/* … */})
client.updateUserDashboard({ user: 1 });
```

WebGL (lovely shaders)

(отступление) Протечка абстракций

WebGL (shaders)

WebRTC (SDP)

WebSQL (abandoned, SQL)

Node.js (file API)

Консистентный API

String.fromCharCode(65)

'abc'.toUpperCase()

atob('abc')

decodeURIComponent('abc')

parseInt('abc')

Консистентный API

```
const MyList = ({ items, className = '', fwRef }) => (
    <ul ref={fwRef} className={className}>{items}</ul>
);

const MyItem = forwardRef(({ text, classes }, ref) => (
    <ul ref={ref} className={classes()}>{text}</ul>
));
```

Уместный API

```
const someAPI = new SomeAPIFactory({ options });

someAPI.getInstance().getData(data => { /* … */ })


const stats: Maybe<newStructure> =
                                someAPI.getInstance().brandNewCall();
```

Уместный API

Lodash

Ramda

Хороший Developer eXperience

Документация – текст, код, примеры, туториалы

```
/**
 * Creates instance of MAIN Backend API
 * @param {string} url - base server address
 * @param {number} timeout - ms for request timeouts
 */
function BackendAPI(url, timeout, { options: { retryCount, ...opts } }) {
    // code goes here
}
```

Хороший Developer eXperience

TTFHW – Time to first hello world

TTFPA - Time to first profitable application

1. Take API
2. Read the docs
3. ???
4. Profit!

Документируем компоненты

https://www.docz.site/

https://storybook.js.org/

https://react-styleguidist.js.org/

https://vue-styleguidist.github.io/

Документируем API

Swagger

JsDoc

TsDoc

Sphinx doc generator

Хороший Developer eXperience

Поддержка

- Отзывчивый автор
- Коммьюнити (чаты, форумы, SO)
- Актуальные туториалы (образцы использования) и/или доки

Стабильность между версиями

Поддержка миграции

Адекватный порог входа

Баги (!)

Развитие

Как сделать хорошо

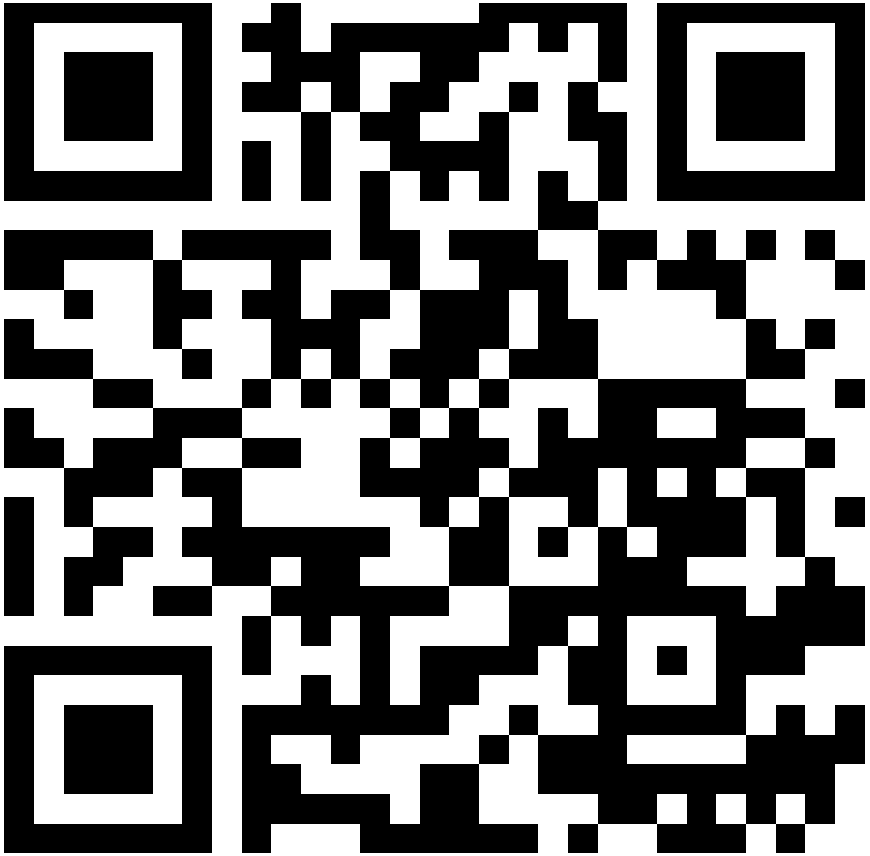
Для кого пишем (?)

Какую проблему решаем

Какие сценарии использования

Знать, где будем (и будем ли) расширять

KISS



Ссылка на ссылки

https://git.io/JvY7X

Спасибо!

DINS®