

个人介绍

基本情况

- 姓名：刘洋
- 联系电话：18284335838
- 坐标：成都市天府新区华阳
- GitHub：[web-liuyang](#)
- Blog：[博客主页](#)
- [网页版简历链接](#)

技术栈

```
{  
  var base = ['HTML5+', 'CSS3', 'JS', 'TS'];  
  var frame = ['Vue', 'React', 'Umi3', 'Uniapp'];  
  var build = ['Webpack', 'Rollup'];  
  var components = ['Antd', 'Element', 'Echarts'];  
  var map = ['BMap', 'AMap'];  
  var node = ['Express', 'Mysql', 'Sequelize'];  
  var extra = ['WebSocket', 'Sass', 'Less', 'SEO'];  
}
```

技术介绍

1. 熟练掌握上面技术栈Web前端技术及相关技术
2. 具有至少2年半以上web前端开发工作经验，能独立承担和推进开发工作，对前端代码优化、代码规范及安全开发有丰富的经验
3. 熟练掌握主流前端框架或者UI库，根据需求实现控件设计、交互实现
4. 熟悉es2015+，能使用rollup等构建工具优化打包组件与SDK，处理打包过程发生的各种问题
5. 了解比较主流的设计模式并实践到工作中
6. 能够独立负责前端项目、业务、以及各种比较复杂的tree型结构与逻辑
7. 项目中实行各种前端组件化，具有良好的OOP编程思想
8. 有自己的SDK、组件库、blogs

轮子

- 基于原生TS编写的实用工具库
[点我链接文档](#) — [点我链接源码](#)
- 基于原生TS编写的浏览器存储解决方案
[点我链接文档](#) — [点我链接源码](#)
- 基于VuePress编写的自动生成侧边栏plugin

[点我链接文档](#) — [点我链接源码](#)

- 基于Uniapp编写的组件库

[点我链接文档](#) — [点我链接源码](#)

- 基于Uniapp编写的SDK

[点我链接文档](#) — [点我链接源码](#)

- 基于React编写的组件库

[点我链接文档](#) — [点我链接源码](#)

- 基于React编写的hooks

[点我链接文档](#) — [点我链接源码](#)

工作状况

千一科技有限公司

工作时间：2018.7 - 2019.12

所在部门：

- 软件研发部

工作描述：

- 开发前端项目。
- 原生JQ编写页面。
- 负责一些网站的重构。
- 测试并解决所有项目遗漏 BUG。
- 维护以及优化老项目。

项目实践：

乐山城东蔬菜批发市场

- 项目技术：[原生JS](#)、[JQ](#)、[bootcss库](#)
- 我的职责：

```
[  
  "负责登陆注册与一些小页面与部分组件的编写",  
  "测试网页中各种BUG",  
  "考虑向下兼容性问题",  
]
```

- 技术使用：

```
[  
  "使用JQ库进行开发",  
  "使用bootcss库进行页面美化",  
  "使用animate动画库进行组件过度",  
  "与合作方进行数据交互对接",  
]
```

成都鑫红江汽车租赁有限公司

工作时间：2019-12 - 2021-3

所在部门：

- 软件研发部

工作描述：

- 开发前端项目。
- 负责公司前端项目的组件的编写。
- 负责公司前端项目的所有封装插件的编写。
- 测试并解决所有项目遗漏 BUG。
- 维护以及优化老项目。
- 负责公司每个项目的代码编写方式，技术框架等。

项目实践：

瑞佰快跑-客户端 | 司机端

- 项目技术：Uniapp, Java
- 我的职责：

```
[  
  "负责整个APP项目的代码编写",  
  "测试APP中各种BUG",  
  "考虑向下兼容性问题",  
  "保证APP的安全性",  
  "兼容小程序"  
]
```

- 技术使用：

```
[
  "当部分功能实现不了的时候调用HTML5+ API Reference接口进行原生开发",
  "高德地图集成的SDK与服务端API",
  "地图搜索、地图导航、地图拖拽地图等操作",
  "APP内嵌套H5页面，唤起其他APP，如：QQ，微信，淘宝，支付宝等",
  "与合作方进行数据交互",
  "支付功能",
  "使用WebSocket进行实时通讯",
  "整个项目进行防抖、懒加载等优化",
  "使用WebSocket进行实时通讯",
  "修改原生代码进行代码重写完成更高需求",
  "页面组件化"
]
```

汽车商城

- 项目技术：`element ui`、`Vue nuxt`、`SEO`、`TS`、`Echarts`
- 我的职责：

```
[
  "独立负责商城的前端代码的编写",
  "测试并处理项目中的各种奇葩BUG",
  "对不同浏览器做兼容性的处理",
  "通过高德地图提供的SDK与API进行地图管理",
  "项目进行SEO搜索引擎优化，进行服务端渲染"
]
```

- 技术使用：

```
[
  "项目使用Vue全家桶与nuxt服务端渲染进行编写",
  "项目使用TypeScript进行静态检查并进行代码编写",
  "使用echarts进行部分数据图形化",
  "在编写的途中抽离各种组件进行复用",
  "代码编写的过程中实行OOP的编程思想",
  "封装SDK进行原型链扩展",
  "使用构造设计模式，进行代码的封装",
  "通过各种中间价进行代码健全与完善"
]
```

租车商城

- 项目技术：`React`、`umi3`、`Antd`
- 我的职责：

```
[
  "负责H5在APP中的实现",
  "调用原生能力进行功能调用",
  "用户信息统计"
]
```

- 技术使用：

```
[  
  "基于TS编写",  
  "基于React的umi3框架",  
  "dva数据管理",  
  "UI框架Antd",  
  "Webpack拓展",  
  "模块化管理"  
]
```

教育背景

1. 学校：西华大学
2. 专业：计算机信息管理
3. 荣誉证书
 - a. 2016 年：考取全国计算机信息高新技术 office 证书。编号 201600240147346
 - b. 2017 年：考取全国计算机信息高新技术 Dreamweaver 证书。编号 201712140195498
 - c. 全国联网可查

自我评价

1. 如果公司要求本科，但是我是专科，希望能够通融一下，不想因为这个原因而错失作为同事的机会
2. 如果公司要求3年的前端经验，我现在恰好是2年半多，两者相差几个月的时间，这点也希望能够通融一下，不想因为这几个月而错失机会
3. 本人性格开朗，善于与人沟通
4. 有很强的团队合作意识，能够很快的适应到新的团队合作组中
5. 在工作上有上进心，喜欢不断汲取新的技术与知识，对上级交代的任务认真负责并能细心完成
6. 有一定的自我钻研能力，平时的业余时间喜欢看一些前端大佬的博客，购买书籍观看
7. 自己写写博客积累知识
8. 如果觉得我不错请添加我微信 18284335838 或者是给我发邮箱 984584014@qq.com
9. 期待您的回复

没啥才艺，给大伙手写几个Demo

New 关键字的实现

```
/**  
 * @description 模拟New关键字的实现  
 * @template T  
 * @param { Function } ctor - 构造函数  
 * @param { *[] } args - 参数列表  
 * @return { T } 构造实例  
 */  
function create(ctor, ...args) {
```

```

// 判断是否是函数
if (ctor instanceof Function === false) {
  throw new Error(`Type Error: ${typeof ctor}`);
}
// 创建空对象
var target = {};

// 创造以构造函数prototype为原型的对象赋值到目标对象target
target = Object.create(ctor.prototype);

// 执行构造函数并把this指向更改为目标对象target
var ret = ctor.apply(target, args);

// 如果构造函数存在返回值那么返回该值，否则返回目标对象target
return ret ? ret : target;
}

/**
 * @description 加减值构造函数
 * @constructor
 * @param { number } count - 初始值
 */
function Test(count) {
  this.count = count;

  this.add = function() {
    return ++this.count;
  };

  this.minus = function() {
    return --this.count;
  };
}

// 实例对象
const test = create(Test, 1);

```

Vuex 简易版

```

/**
 * 实现Vuex 用了一下几种设计模式
 * 观察者模式
 * 发布订阅者模式
 * 单例模式
 * 构造者模式
 */

type IState = {
  [key: string]: any;
};

type IAction = {

```

```

    [key: string]: (commit: ICommit, payload: any) => void;
};

type IMutation = {
    [key: string]: (state: IState, payload: any) => void;
};

type IWatchListen = (key: keyof IState, oVal: any, nVal: any) => void;

type IListen = (callback: IWatchListen) => void;

type IDispatch = (key: keyof IAction, payload: any) => void;

type ICommit = (key: keyof IMutation, payload: any) => void;

type IInit<T> = (store: T) => InstanceType<new () => Store<T>>;

interface IStore {
    state?: IState;
    actions?: IAction;
    mutations?: IMutation;
}

class Store<T extends IStore> {
    private state?: IState;
    private actions?: IAction;
    private mutations?: IMutation;
    private _watchListen?: IWatchListen[];
    private static instance: InstanceType<new () => Store<IStore>>;
    constructor({ state, actions, mutations }: T) {
        // 通过Proxy代理器监听state数据变化
        this.state = new Proxy<typeof state>(state, { set: setState });
        this.actions = actions;
        this.mutations = mutations;
        // 监听state变化后触发的函数列表
        this._watchListen = [];

        // 设置state属性值
        function setState(target: T, key: keyof typeof state, value: any) {
            // 触发监听事件
            for (const item of Store.instance._watchListen) {
                item(key, target[key], value);
            }
            target[key] = value;
            return true;
        }
    }

    // 初始化仓库
    public static init: IInit<IStore> = (store) => {
        if (Store.instance) {
            return Store.instance;
        }
    }
}

```

```

    } else {
        Store.instance = new Store<IStore>(store);
        return Store.instance;
    }
};

// 触发action
public dispatch: IDispatch = (key, payload) => {
    for (const [item, value] of Object.entries(this.actions)) {
        if (item === key) {
            value(this.commit.bind(this), payload);
            return;
        }
    }
};

// 触发mutation
public commit: ICommit = (key, payload) => {
    console.log(this);
    for (const [item, value] of Object.entries(this.mutations)) {
        if (item === key) {
            value(this.state, payload);
            return;
        }
    }
};

// 监听state
public listen: IListen = (callback) => {
    this._watchListen.push(callback);
};
}

const store = Store.init({
    state: {
        count: 0,
    },
    actions: {
        add(commit, data) {
            console.log(`触发了actions中的add值为:${data}`);
            commit('ADD', data);
        },
    },
    mutations: {
        ADD(state, data) {
            console.log(`触发了mutations中的ADD值为:${data}`);
            state.count = data;
        },
    },
});

store.listen((key, oVal, nVal) => {

```



```
    console.log(`${key}发生了变化从${oVal}到${nVal}`);  
  });  
  
  store.dispatch('add', 1);
```