

## 选择一个合适的UI库

	Mint-ui	vux	Cube-ui	vant
维护团队	饿了么团队	个人	滴滴	有赞
更新迭代	1年没提交	三个月	持续更新	持续更新
Github	13K	14k	5K	7K
测试	基本没啥覆盖	基本没覆盖	测试完备	测试完备 80%
缺点	基本停止维护	个人项目 更新迭代不及时, 没适配vue-cli3		
特点			后编译, create-api, 滴滴线上业务	有赞线上项目, 支持SSR, TS

## 准备工作

```
vue add cube-ui
vue add router
vue add vuex
```

登录表单: [cube-ui的表单文档](#)

```
// Login.vue
<div>
  <div class="logo">
    
  </div>
  <!-- <cube-button>登录</cube-button> -->
  <cube-form
    :model="model"
    :schema="schema"
    @submit.prevent="handleLogin"
    @validate="handleValidate"
  >
  </cube-form>
</div>
```

分别设置model和schema

```
model: { username: "", passwd: "" },
schema: {
  fields: [
    {
      type: "input",
      modelKey: "username",
      label: "用户名",
      props: {
        placeholder: "请输入用户名"
      },
      rules: {
        // 校验规则
        required: true
      },
      trigger: "blur"
    },
  ],
}
```

发起登录请求, Login.vue

```
this.$store
  .dispatch("login", this.model)
  .then(success => {})
  .catch(error => {});
```

登录动作, store.js

```
import us from "../service/user";

login({ commit }, model) {
  return us.login(model).then(res => {
    const {token} = res.data;
    if (token) {
      localStorage.setItem("token", token);
      commit("setLoginState", true);
      return true;
    }
    return false;
  });
}
```

接口服务, service/user.js

```
import axios from "axios";

export default {
  login(user) {
    return axios.get("/api/login", {params: user});
  }
};
```

vue.config.js中, 模拟接口

```
configureWebpack:{
  devServer:{
    before(app){
      app.get("/api/login", function(req, res) {
        const { username, passwd } = req.query;
        console.log(username, passwd);

        if (username == "kaikeba" && passwd == "123") {
          res.json({ code: 1, token: "jilei" });
        } else {
          res
            .status(401)
            .json({ code: 0, message: "用户名或者密码错误" });
        }
      });
    }
  }
}
```

## http拦截器

有了token之后, 每次http请求发出, 都要加载header上

```
// interceptor.js
const axios = require("axios");

export default function() {
  axios.interceptors.request.use(config => {
    const token = localStorage.getItem('token')
    if (token) {
      config.headers.token = token;
    }
    return config;
  });
}

// 启用, main.js
import interceptor from './interceptor'
interceptor();
```

## 接口验证

```
// mock接口, vue.config.js
function auth(req, res, next) {
  if (req.headers.token) {
    // 已认证
    next()
  } else {
    res.sendStatus(401)
  }
}

app.get("/api/userinfo", auth, function(req, res) {
  res.json({ code: 1, data: { name: "tom", age: 20 } });
});
```

## http拦截响应

```
// interceptor.js
export default function(vm) { // 传入vue实例
  // ...
  // 响应拦截
  axios.interceptors.response.use(null, err => {
    if (err.response.status === 401) {
      // 清空vuex和localStorage
      vm.$store.dispatch("logout");
      // 跳转login
      vm.$router.push("/login");
    }
    return Promise.reject(err);
  });
}
```

## 令牌机制

- 服务端, ~/server/server.js

```
const Koa = require("koa");
const Router = require("koa-router");
const jwt = require("jsonwebtoken");
const jwtAuth = require("koa-jwt");

const secret = "it's a secret";

const app = new Koa();
const router = new Router();
```

```

router.get("/api/login", async ctx => {
  const { username, passwd } = ctx.query;
  console.log(username, passwd);

  if (username == "kaikeba" && passwd == "123") {
    // 生成令牌
    const token = jwt.sign(
      {
        data: { name: "kaikeba" }, // 用户信息数据
        exp: Math.floor(Date.now() / 1000) + 60 * 60 // 过期时间
      },
      secret
    );
    ctx.body = { code: 1, token };
  } else {
    ctx.status = 401;
    ctx.body = { code: 0, message: "用户名或者密码错误" };
  }
});

router.get(
  "/api/userinfo",
  jwtAuth({ secret }),
  async ctx => {
    ctx.body = { code: 1, data: { name: "jerry", age: 20 } };
  }
);
app.use(router.routes());
app.listen(3000);

```

- 服务器代理, vue.config.js

```

devServer: {
  //代理配置
  proxy: {
    "/api": {
      target: "http://127.0.0.1:3000/",
      changOrigin: true
    }
  },
  // before(app) { ... }
}

```

- 拦截器处的修改, interceptor.js

```

config.headers.Authorization = 'Bearer ' + token;

```

# 作业

---

令牌过期的实现

