

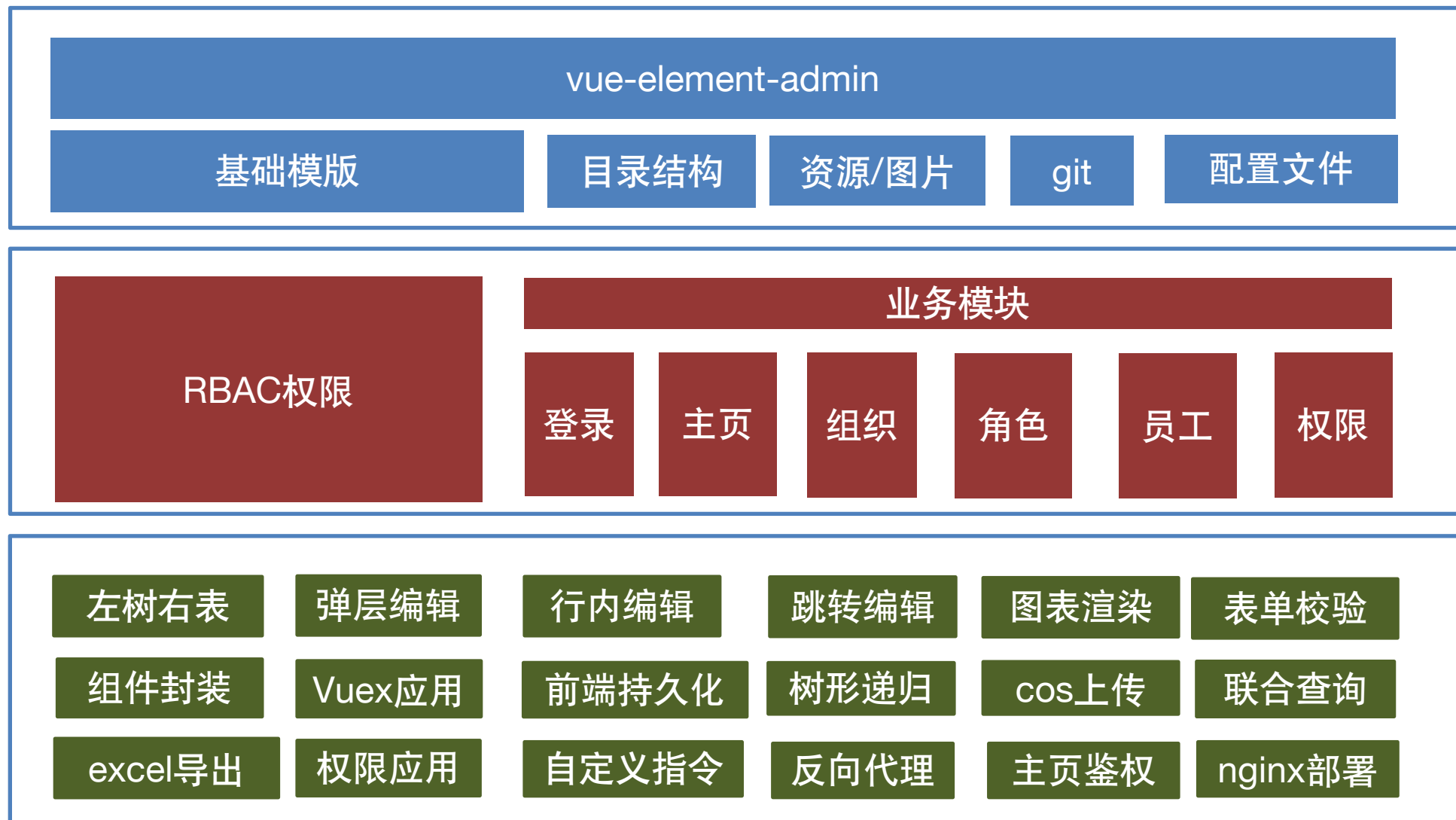
# 人力资源后台项目



黑马程序员  
[www.itheima.com](http://www.itheima.com)

传智教育旗下  
高端IT教育品牌

## 项目架构



底层框架

业务模块

解决方案

## 课程安排

基础环境+登录 (day01)

登录 + 主页鉴权 (day02)

主页模块 (day03)

组织架构 (day04)

角色管理 (day05)

左树右表 (day06)

员工新增修改 (day07)

cos上传+权限数据(day08)

权限应用(day09)

首页 + 部署(day10)

## 学完该课程具备

基于vue-element-admin二次开发

前端项目CRUD(增删改查)

封装组件适配产品需求

三种编辑模式(弹层, 编辑, 跳转)

反向代理解决跨域

表单验证及自定义校验

树形组件及树形数据处理

自定义指令解决权限技术方案

Cos第三方上传

左树右表-联合查询

封装请求工具的项目应用

Vuex技术在项目中的应用

RBAC动态路由

echarts图表应用

请求工具封装

主页鉴权

nginx项目部署上线

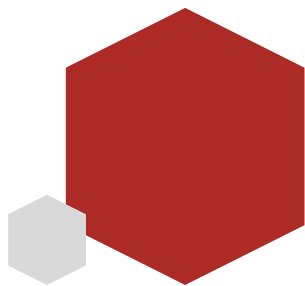
excel的导入导出

## 课程资料

vue-element-admin文档地址: <https://panjiachen.gitee.io/vue-element-admin-site/zh/>

vue-element-admin演示地址: <https://panjiachen.github.io/vue-element-admin/>

项目演示地址: <https://heimahr.itheima.net>



## 拉取项目基础代码

## 拉取模版代码

```
git clone https://github.com/PanJiaChen/vue-admin-template.git heimahr
```

其他：core-js版本处理

- 项目模版中的core-js的版本号有些滞后，需要将其版本号改为“3.25.5”再安装依赖



# 总结

1. 使用模版是更快更好更贴近企业的开发模式

```
https://github.com/PanJiaChen/vue-admin-template.git
```

2. 修改core-js版本为3.25.5并安装依赖

```
core-js: 3.6.5 ➔ 3.25.5
```

```
yarn
```

3. 启动命令在package.json中查看

```
yarn dev
```

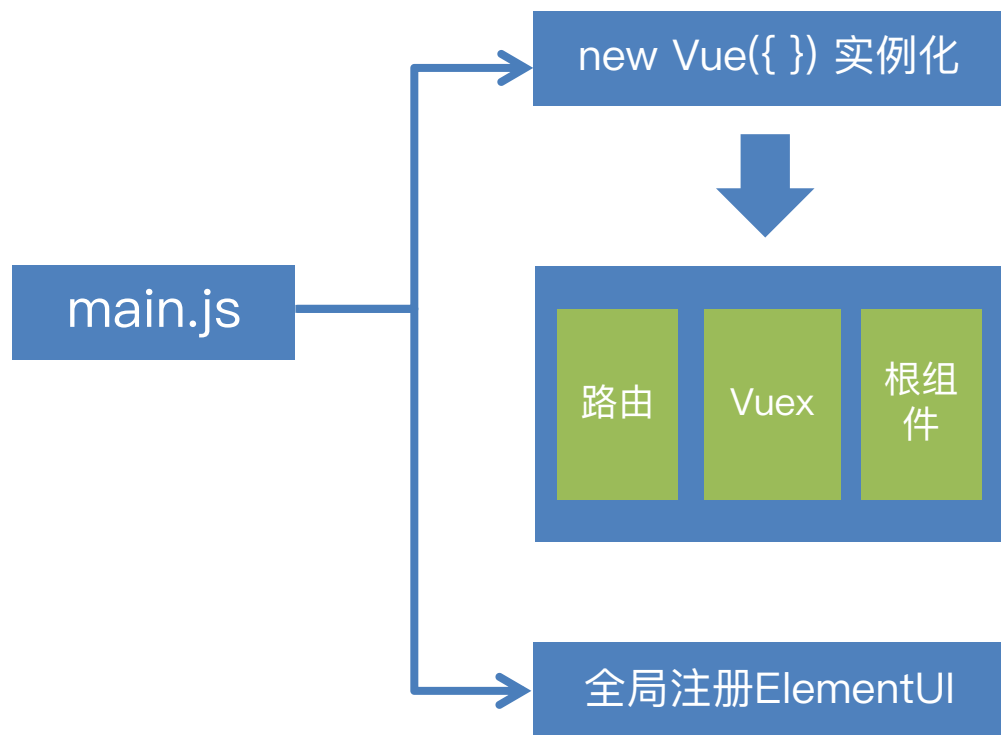
4. 插件是提升开发效率的利器：ESLint & Vetur



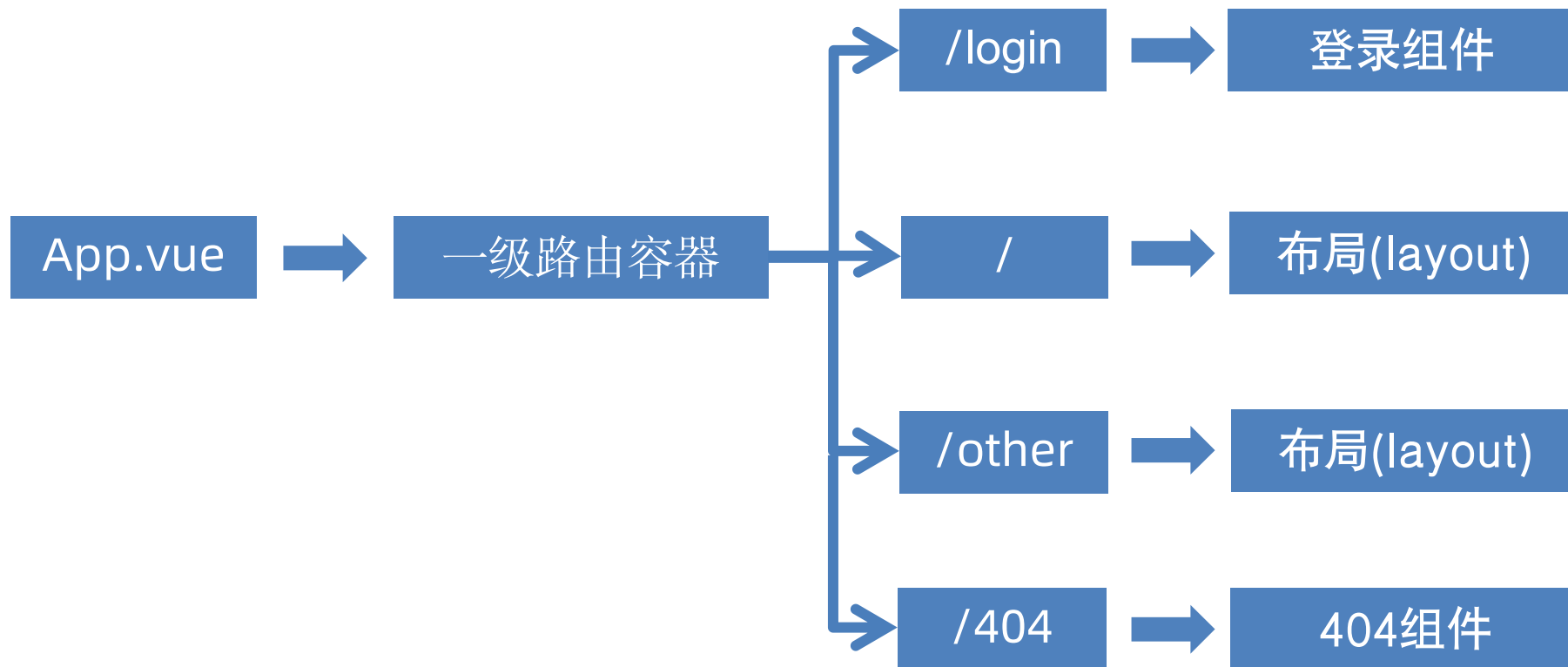
## 项目目录和文件介绍

```
├─ src                # 源代码目录
│   ├─ api            # 所有请求
│   ├─ assets         # 主题 字体等静态资源
│   ├─ components    # 全局公用组件
│   ├─ icons          # 项目所有 svg icons
│   ├─ layout         # 全局 layout
│   ├─ router         # 路由
│   ├─ store          # 全局 store管理
│   ├─ styles         # 全局样式
│   ├─ utils          # 全局公用方法
│   ├─ views          # views 所有页面
│   ├─ App.vue        # 入口页面
│   ├─ main.js        # 入口文件 加载组件 初始化等
│   └─ permission.js  # 权限管理
│   └─ settings.js    # 配置文件
```

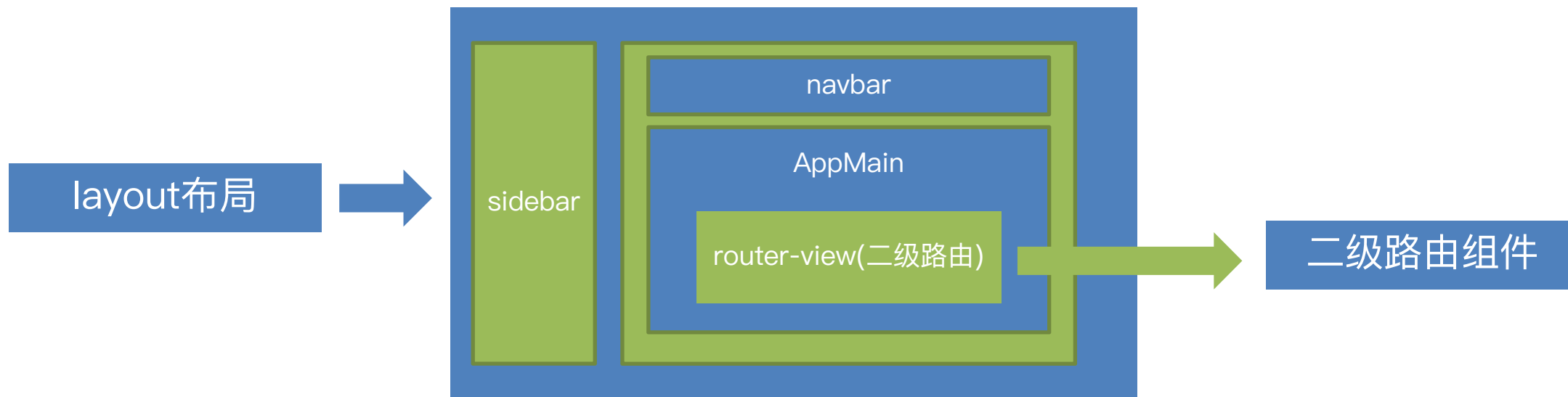
## 入口文件说明



## 根组件App.vue



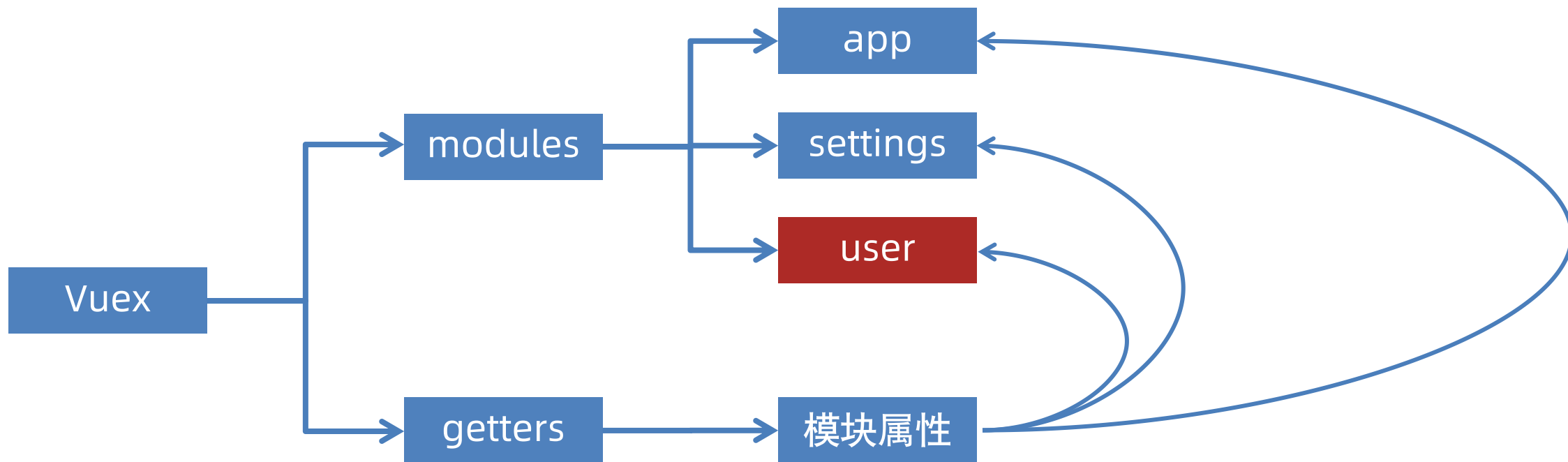
## layout布局组件



## 基础设置settings.js和导航守卫permission.js

- settings.js导出网站基础配置，包括：网站标题、固定header、显示logo
- permission.js(权限)，主要负责路由导航守卫

## Vuex结构



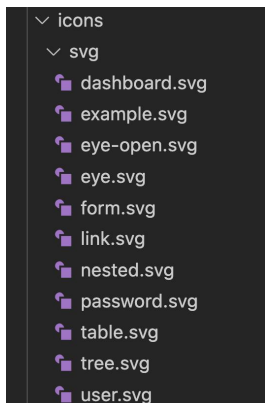
人力资源项目针对user模块重写，其他模块进行复用

## 模板中的Vuex的设计思想

- 页面交互状态（**折叠侧边栏-固定头部**）使用全局状态Vuex
- 根据功能拆分成不同的模块（**modules**）进行状态管理
- 通过getters建立对于模块中属性的快捷访问

## 使用模板中的icon图标

- `src/icons/svg`目录下的图标都可以使用

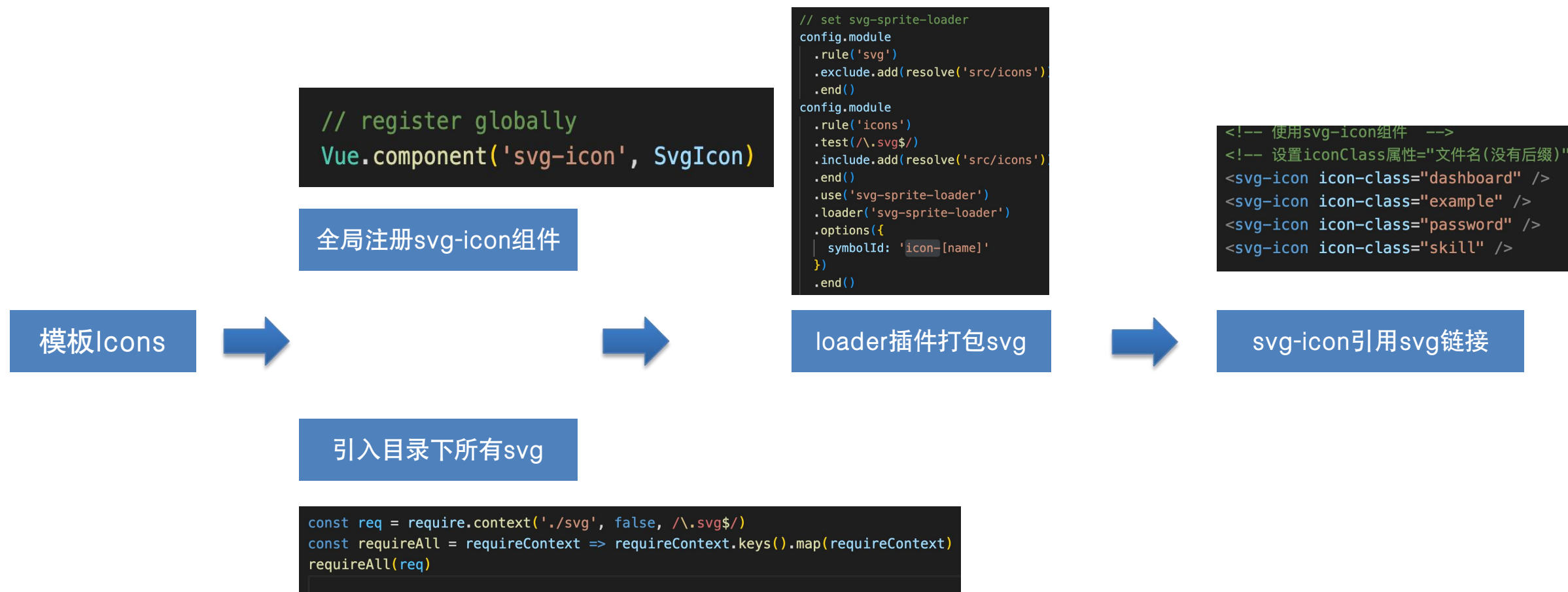


- 图标文件名直接设置为`svg-icon`组件的`iconClass`属性

```
<svg-icon icon-class="dashboard" />
<svg-icon icon-class="example" />
<svg-icon icon-class="eye-open" />
```



## 模板icons图标实现思路



## 知识扩展

- 了解：require.context(路径, 是否扫描子目录, 正则匹配) 可以引入某个目录下的内容
- 了解：svg-sprite-loader打包了所有svg到一个svg标签上，将svg名称作为symbol标签的id属性
- 了解：svg-icon使用iconClass属性引用了symbol的id

## 导入样式和资源，使用git管理项目

- 将准备好的 图片，样式，svg导入到项目src中



- 删除原有git文件，重新初始化仓库，建立远程仓库，并同步推送

```
git init # 初始化仓库
```

```
git add . # 添加到暂存区
```

```
git commit -m "日志消息" # 提交本地仓库
```

本地仓库

```
git remote add origin 远程仓库地址 # 本地仓库配置远程仓库地址
```

```
git push -u origin master # 推送到远程仓库
```

推送远程

## 登录页的结构和表单

- 准备登录页面的样式和结构
- 实现登录表单的结构

### 登录



```
<el-form autocomplete="off">
  <el-form-item>
    <el-input placeholder="请输入用户名" />
  </el-form-item>
  <el-form-item>
    <el-input show-password placeholder="请输入密码" />
  </el-form-item>
  <el-form-item>
    <el-checkbox>用户平台使用协议</el-checkbox>
  </el-form-item>
  <el-form-item>
    <el-button style="width: 350px" type="primary" block>登录</el-button>
  </el-form-item>
</el-form>
```

- 提交代码

## 登录表单校验-实现

☐ 用户平台使用协议

登录

```
<el-card shadow="never" class="login-card">
  <el-form ref="form" autocomplete="off" :rules="loginRules" :model="loginForm">
    <el-form-item prop="mobile">
      <el-input v-model="loginForm.mobile" placeholder="请输入用户名" />
    </el-form-item>
    <el-form-item prop="password">
      <el-input
        v-model="loginForm.password"
        show-password
        placeholder="请输入密码"
      />
    </el-form-item>
    <el-form-item prop="isAgree">
      <el-checkbox v-model="loginForm.isAgree">用户平台使用协议</el-checkbox>
    </el-form-item>
    <el-form-item>
      <el-button style="width: 350px" type="primary">登录</el-button>
    </el-form-item>
  </el-form>
```

## 登录表单的规则

- 手机号 必填，手机号规则

```
{ required: true, trigger: 'blur', message: '手机号不能为空' },  
{ trigger: 'blur', pattern: /^1[3-9]\d{9}$/, message: '手机号格式不正确' }
```

- 密码 必填，6-16位长度

```
{ required: true, trigger: 'blur', message: '密码不能为空' },  
{ min: 6, max: 16, message: '密码的长度在6-16位之间', trigger: 'blur' }
```

- 用户协议 必须勾选

```
isAgree: [{  
  validator: (rule, value, callback) =>  
    value ? callback() : callback(new Error('您还未勾选用户协议'))  
}]
```

- 表单整体校验

☐ 用户平台使用协议

```
login() {  
  this.$refs.form.validate(async(isOK) => {  
    if (isOK) {  
      alert('校验通过!')  
    }  
  })  
}
```



传智教育旗下高端IT教育品牌