前端报表开发培训--Vue

2019-12-30

1.开发框架

1.1 Vue

Vue.js 是构建 Web 界面的 JavaScript 库，提供数据驱动的组件，还有简单灵活的 API，使得 MVVM 更简单。

主要特性：

- 可扩展的数据绑定
- 将普通的 JS 对象作为 model
- 简洁明了的 API
- 组件化 UI 构建
- 配合别的库使用

其他：

1.Vue中文官网

2.教程

3.对比其他框架

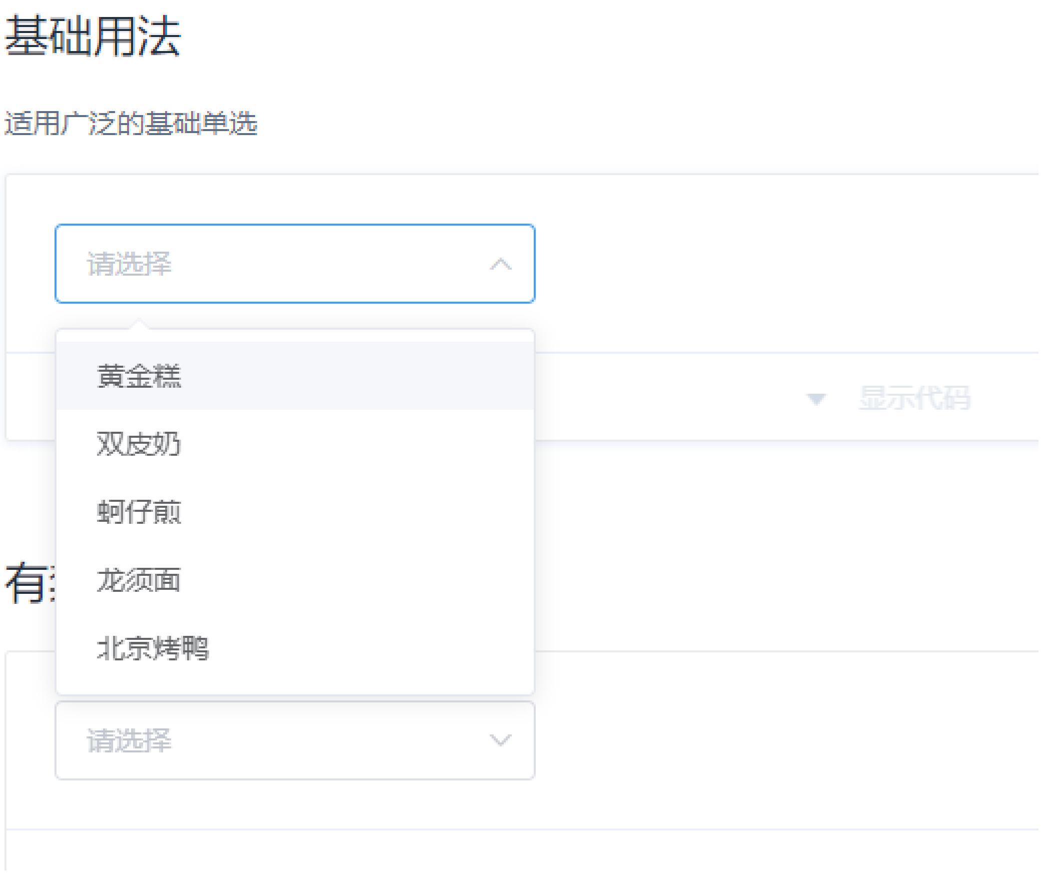
1.2 ElementUI

Element，一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的组件库，提供了配套设计资源，帮助你的网站快速成型。由饿了么公司前端团队开源。

官网

常用组件：

- 选择器

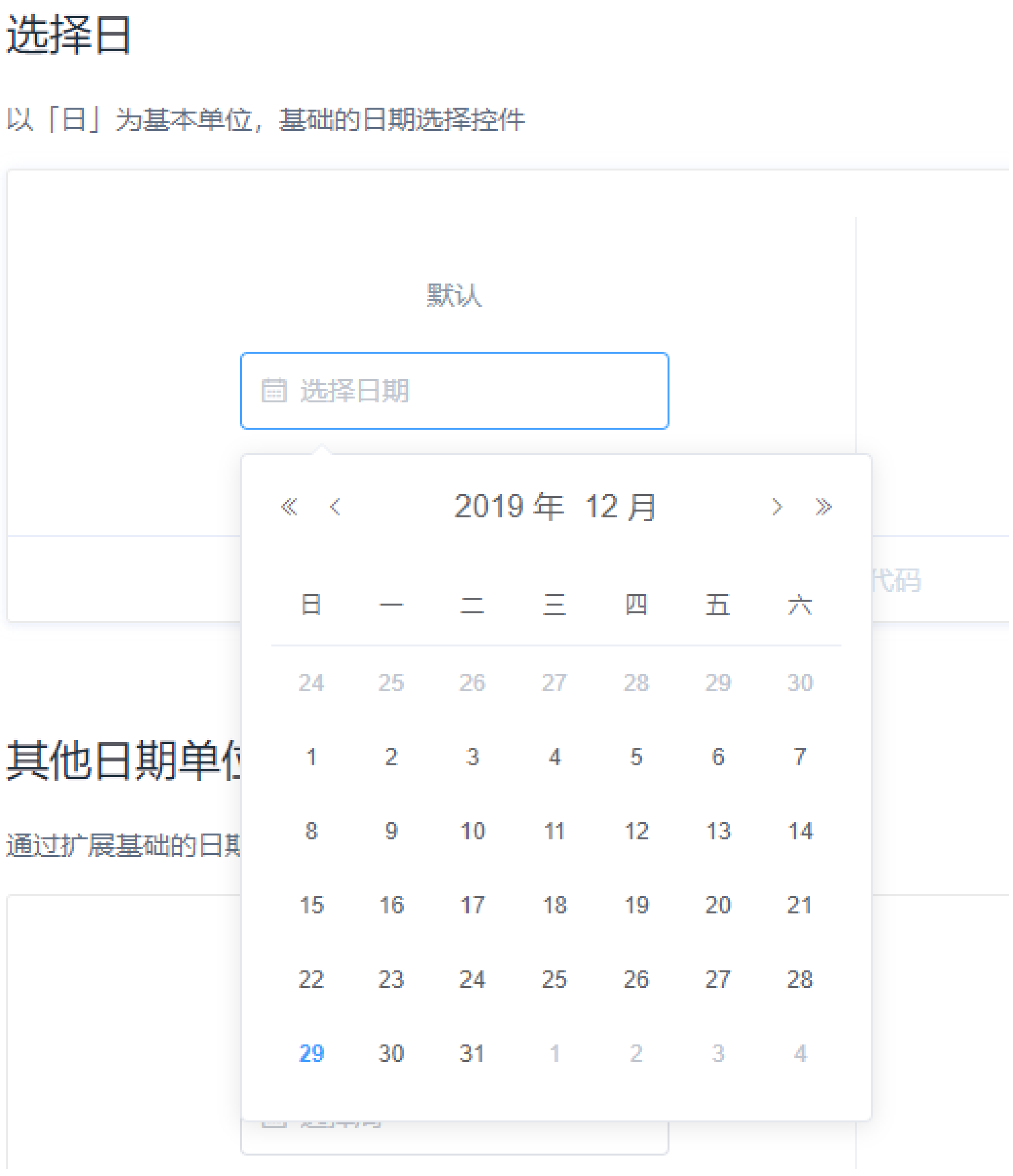


基础用法

适用广泛的基础单选

请选择

- 日期选择器

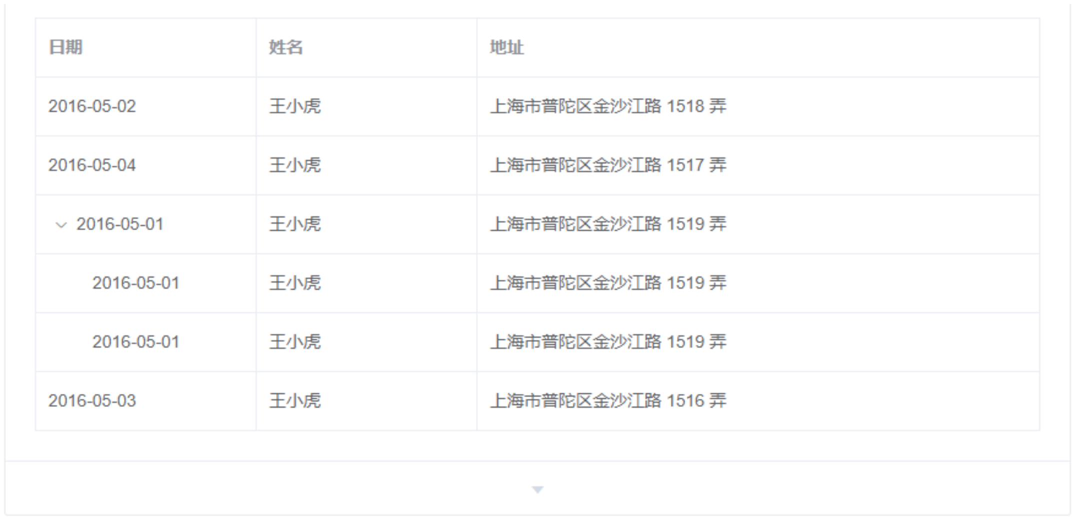


选择日

以「日」为基本单位，基础的日期选择控件

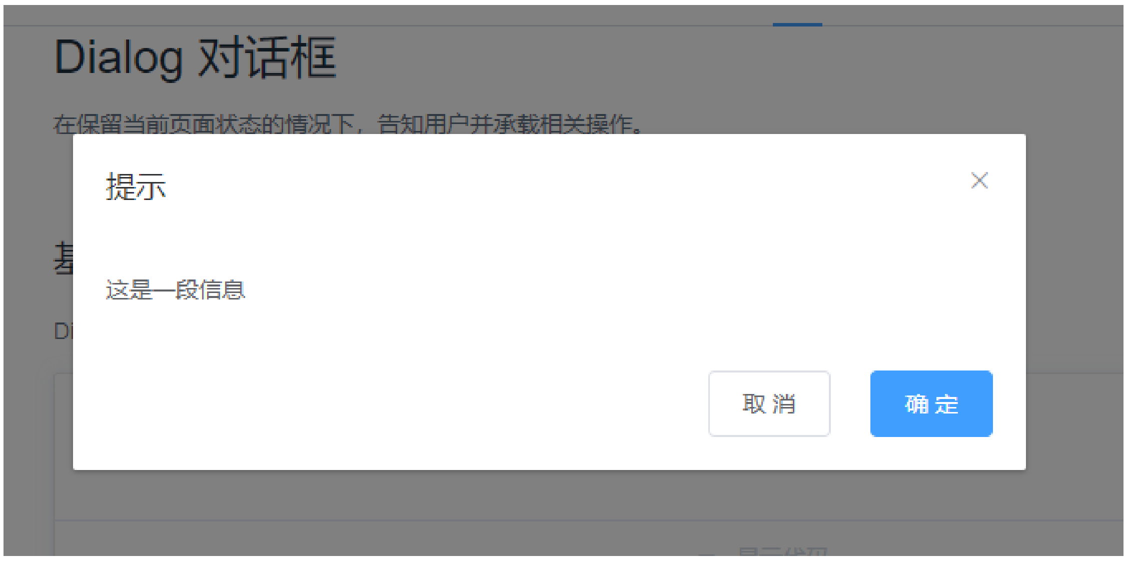
日	一	二	三	四	五	六
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

- 表格



日期	姓名	地址
2016-05-02	王小虎	上海市普陀区金沙江路 1518 弄
2016-05-04	王小虎	上海市普陀区金沙江路 1517 弄
2016-05-01	王小虎	上海市普陀区金沙江路 1519 弄
2016-05-01	王小虎	上海市普陀区金沙江路 1519 弄
2016-05-01	王小虎	上海市普陀区金沙江路 1519 弄
2016-05-03	王小虎	上海市普陀区金沙江路 1516 弄

- 对话框


Dialog 对话框

在保留当前页面状态的情况下，告知用户并承载相关操作。

提示

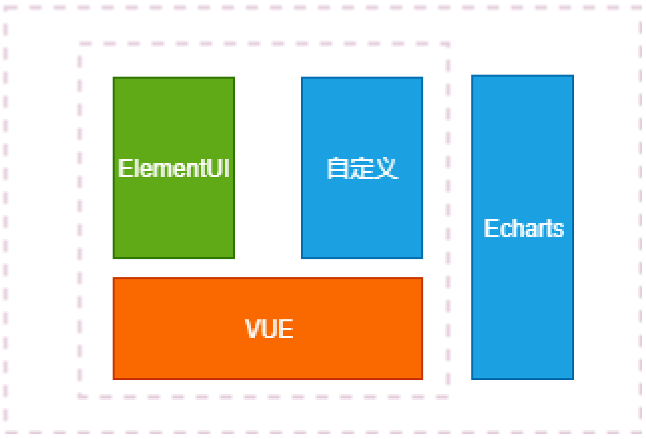
这是一段信息

1.3 Echarts

ECharts是一款由百度前端技术部开发的，基于Javascript的数据可视化图表库，提供直观，生动，可交互，可个性化定制的数据可视化图表。

官方文档

1.4 总体结构：



2. 环境搭建

2.1 Node.js安装

按照安装包提示安装。

2.1 YARN安装

官网 按照安装包提示安装，yarn依赖于Node.js。

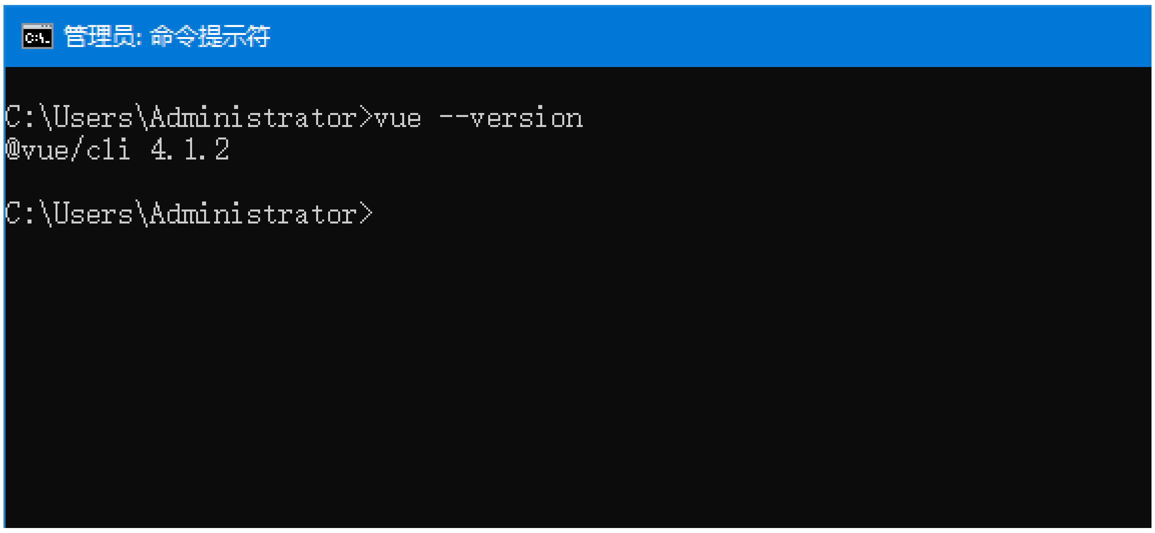
2.2 Vue安装

通过Vue CLI安装Vue。

```
npm install -g @vue/cli
# 或者
yarn global add @vue/cli
```

此处选择yarn的安装方式。安装之后可以通过version检查是否安装成功。

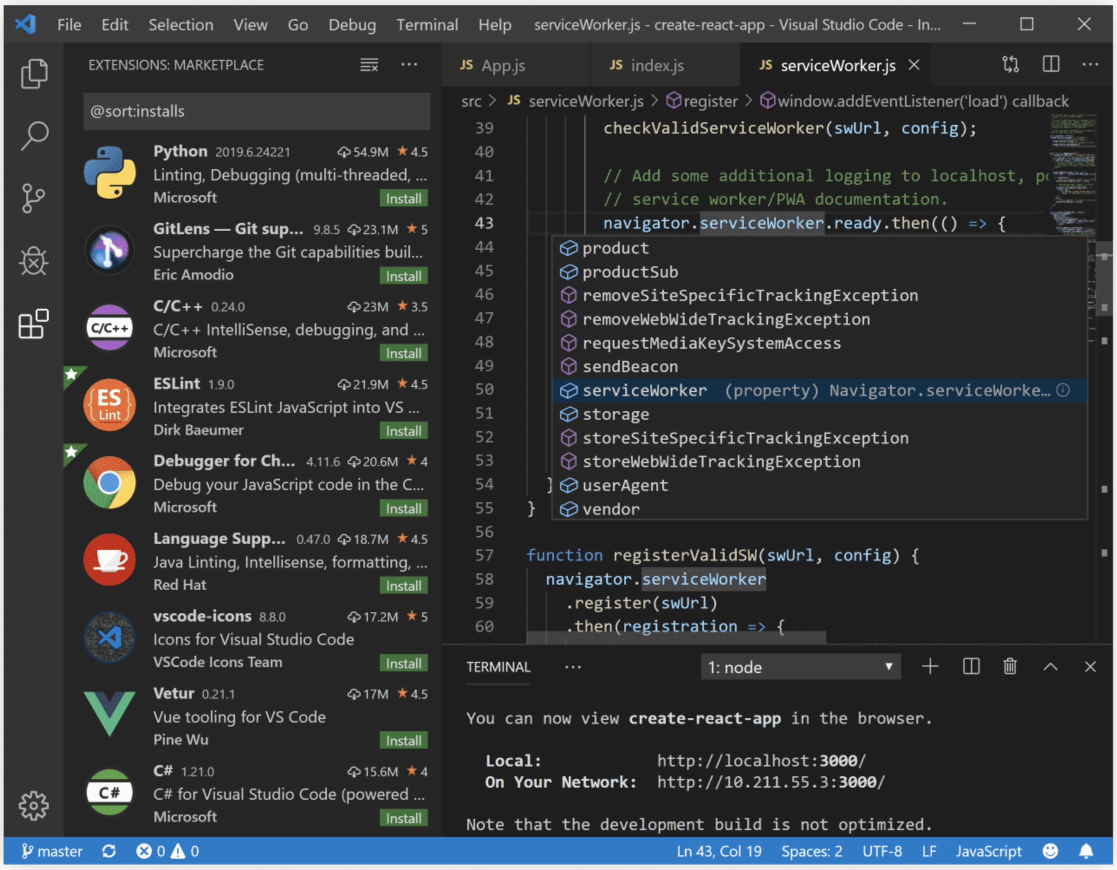
```
vue --version
```



```
C:\Users\Administrator>vue --version
@vue/cli 4.1.2

C:\Users\Administrator>
```

2.3 开发工具

可以选择任意文本编辑工具，此处选择Visual Studio Code。


File
Edit
Selection
View
Go
Debug
Terminal
Help
serviceWorker.js - create-react-app - Visual Studio Code - In...
EXTENSIONS: MARKETPLACE
@sort:installs
Python 2019.6.24221 54.9M 4.5 Linting, Debugging (multi-threaded, ... Microsoft Install
GitLens — Git sup... 9.8.5 23.1M 5 Supercharge the Git capabilities buil... Eric Amodio Install
C/C++ 0.24.0 23M 3.5 C/C++ IntelliSense, debugging, and ... Microsoft Install
ESLint 1.9.0 21.9M 4.5 Integrates ESLint JavaScript into VS ... Dirk Baeumer Install
Debugger for Ch... 4.11.6 20.6M 4 Debug your JavaScript code in the C... Microsoft Install
Language Supp... 0.47.0 18.7M 4.5 Java Linting, Intellisense, formatting, ... Red Hat Install
vscode-icons 8.8.0 17.2M 5 Icons for Visual Studio Code VSCode Icons Team Install
Vetur 0.21.1 17M 4.5 Vue tooling for VS Code Pine Wu Install
C# 1.21.0 15.6M 4 C# for Visual Studio Code (powered ... Microsoft Install
App.js
index.js
serviceWorker.js
src > serviceWorker.js > register > window.addEventListener('load') callback
product
productSub
removeSiteSpecificTrackingException
removeWebWideTrackingException
requestMediaKeySystemAccess
sendBeacon
serviceWorker (property) Navigator.serviceWorke...
storage
storeSiteSpecificTrackingException
storeWebWideTrackingException
userAgent
vendor
TERMINAL
1: node
master
0 0
Ln 43, Col 19
Spaces: 2
UTF-8
LF
JavaScript

```
        checkValidServiceWorker(swUrl, config);

        // Add some additional logging to localhost, p
        // service worker/PWA documentation.
        navigator.serviceWorker.ready.then(() => {
      }
    }

function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then(registration => {
```

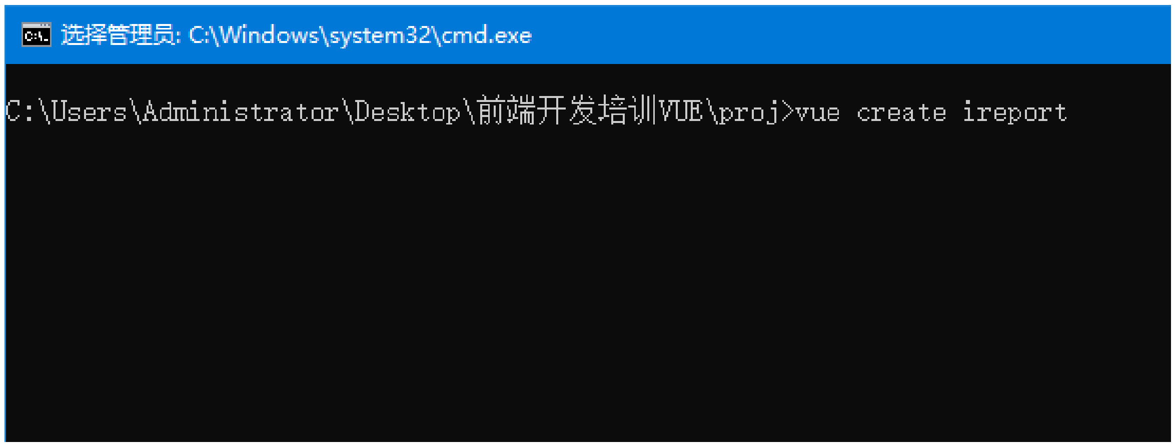
```
You can now view create-react-app in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://10.211.55.3:3000/

Note that the development build is not optimized.
```

3. 工程创建

3.1 通过Vue CLI创建工程

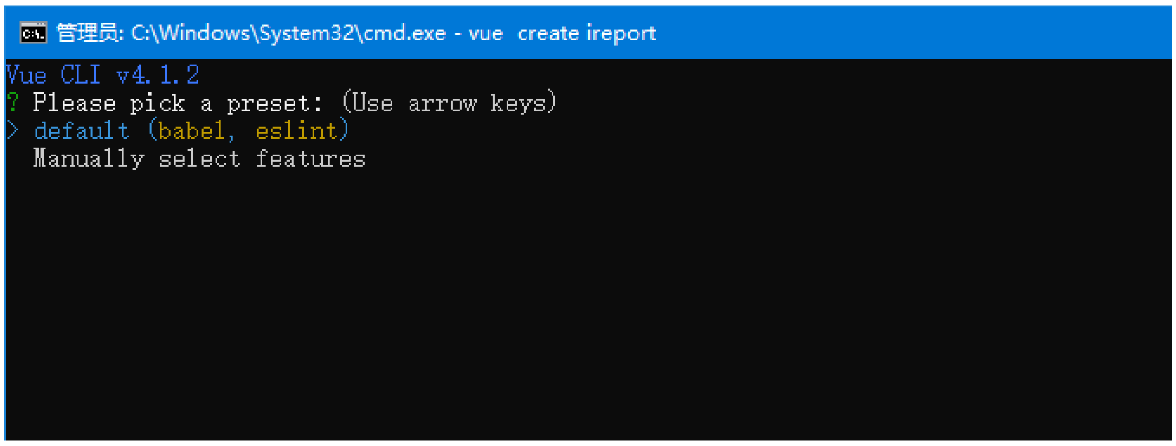
```
vue create ireport
```

创建工程：



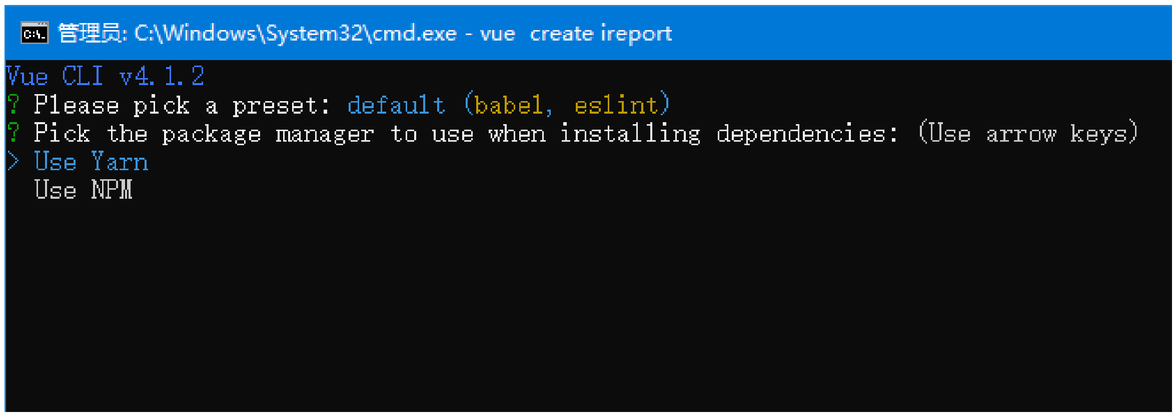
```
选择管理员: C:\Windows\system32\cmd.exe

C:\Users\Administrator\Desktop\前端开发培训VUE\proj>vue create ireport
```

选择默认：



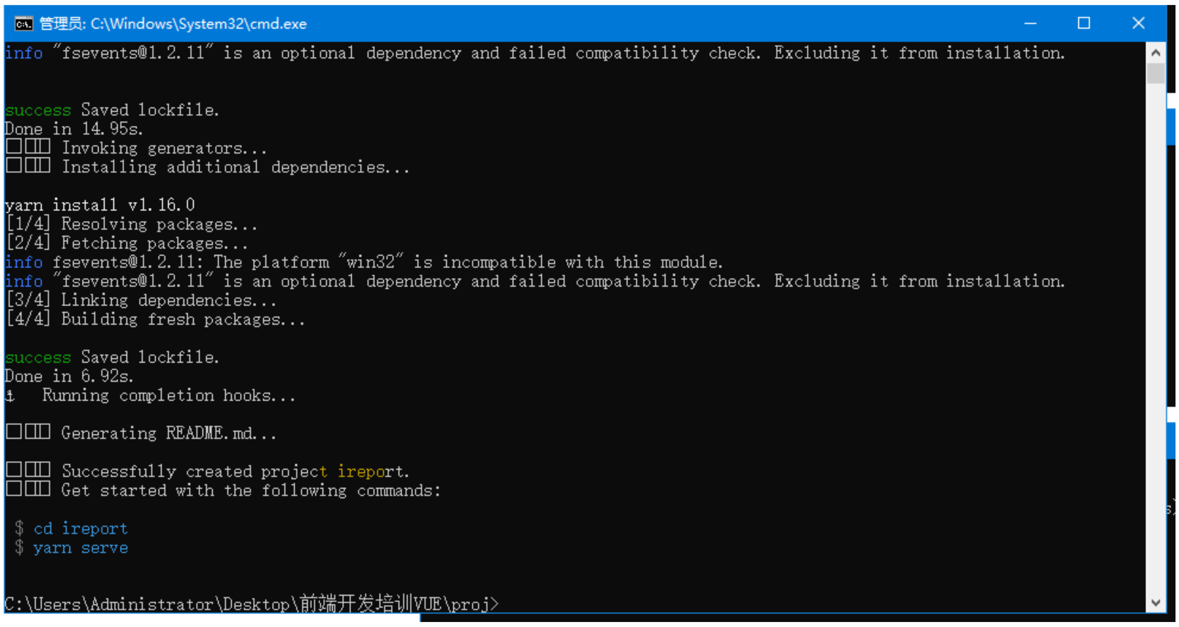
```
管理员: C:\Windows\System32\cmd.exe - vue  create ireport
Vue CLI v4.1.2
? Please pick a preset: (Use arrow keys)
> default (babel, eslint)
  Manually select features
```

选择Yarn：



```
管理员: C:\Windows\System32\cmd.exe - vue  create ireport
Vue CLI v4.1.2
? Please pick a preset: default (babel, eslint)
? Pick the package manager to use when installing dependencies: (Use arrow keys)
> Use Yarn
  Use NPM
```

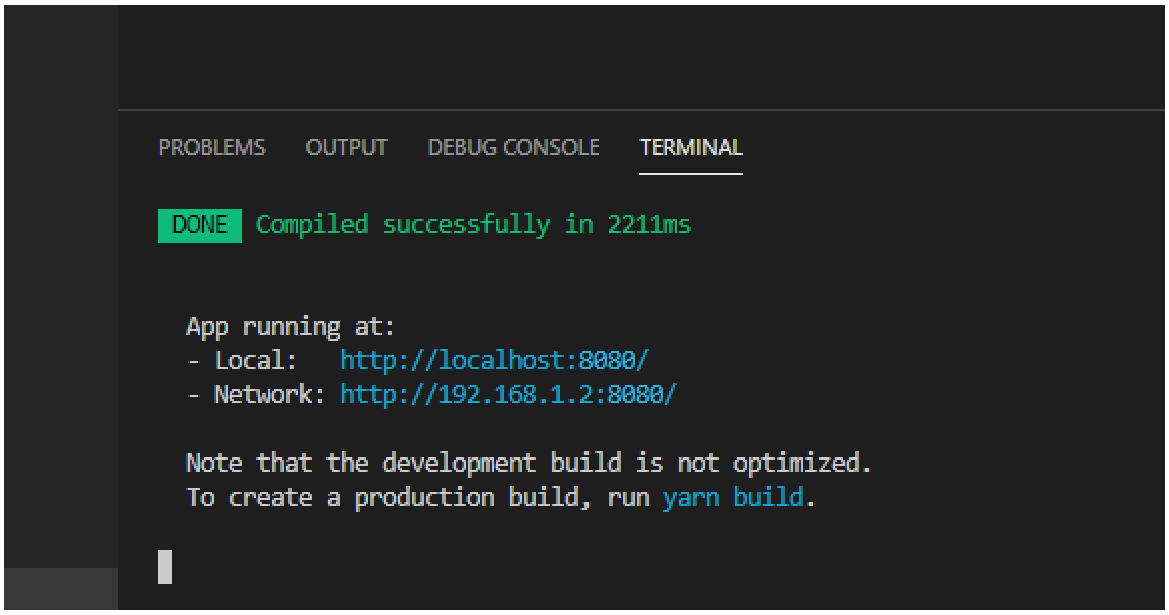
创建完成：



```
info "fsevents@1.2.11" is an optional dependency and failed compatibility check. Excluding it from installation.

success Saved lockfile.
Done in 14.95s.
□□□ Invoking generators...
□□□ Installing additional dependencies...

yarn install v1.16.0
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@1.2.11: The platform "win32" is incompatible with this module.
info "fsevents@1.2.11" is an optional dependency and failed compatibility check. Excluding it from installation.
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
Done in 6.92s.
⚓  Running completion hooks...

□□□ Generating README.md...

□□□ Successfully created project ireport.
□□□ Get started with the following commands:

 $ cd ireport
 $ yarn serve

C:\Users\Administrator\Desktop\前端开发培训VUE\proj>
```

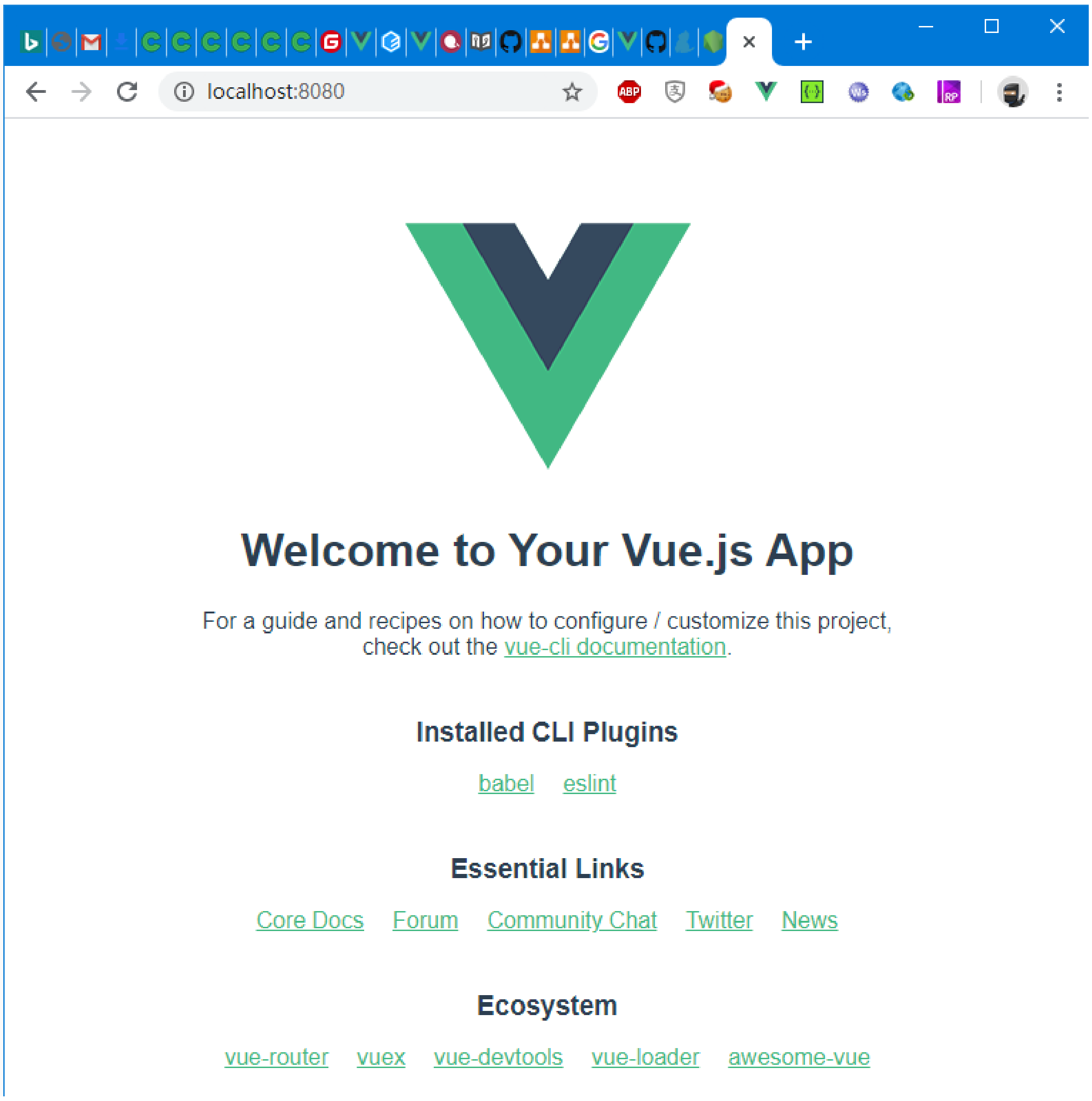
3.2 运行工程

```
yarn serve
```



```
 DONE  Compiled successfully in 2211ms


  App running at:
  - Local:   http://localhost:8080/
  - Network: http://192.168.1.2:8080/

  Note that the development build is not optimized.
  To create a production build, run yarn build.
```

工程运行成功：


localhost:8080

Welcome to Your Vue.js App

For a guide and recipes on how to configure / customize this project, check out the vue-cli documentation.

Installed CLI Plugins

babel eslint

Essential Links

Core Docs Forum Community Chat Twitter News

Ecosystem

vue-router vuex vue-devtools vue-loader awesome-vue

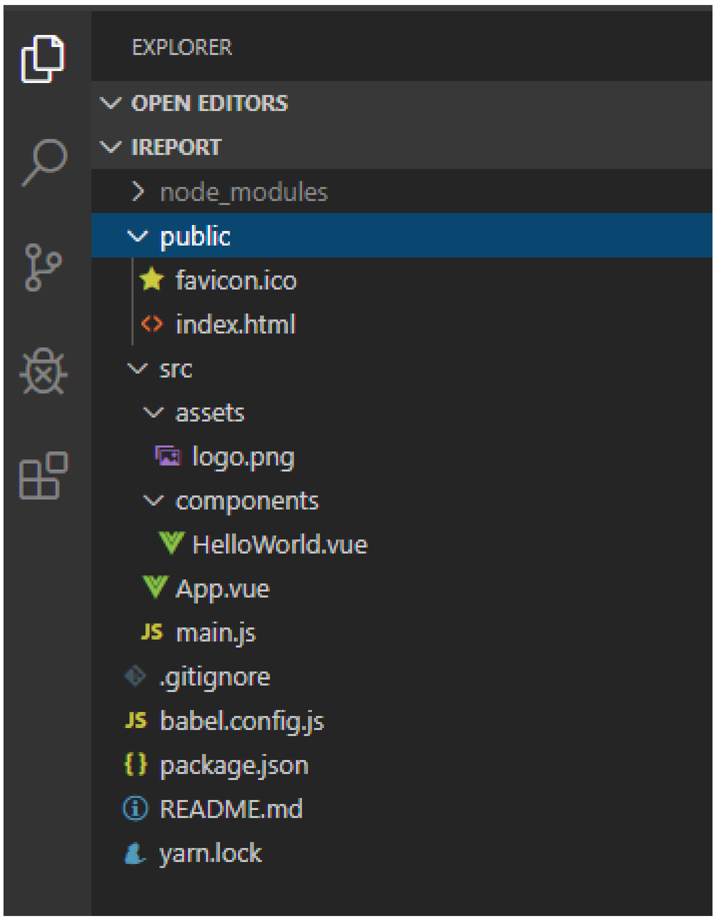
3.3 多页面应用

pages，Vue CLI有关于pages的介绍。

```
module.exports = {
  pages: {
    index: {
      // entry for the page
      entry: 'src/index/main.js',
      // the source template
      template: 'public/index.html',
      // output as dist/index.html
      filename: 'index.html',
      // when using title option,
      // template title tag needs to be <title><%=
htmlWebpackPlugin.options.title %></title>
      title: 'Index Page',
      // chunks to include on this page, by default includes
      // extracted common chunks and vendor chunks.
      chunks: ['chunk-vendors', 'chunk-common', 'index']
    },
    // when using the entry-only string format,
    // template is inferred to be `public/subpage.html`
    // and falls back to `public/index.html` if not found.
    // Output filename is inferred to be `subpage.html`.
    subpage: 'src/subpage/main.js'
  }
}
```

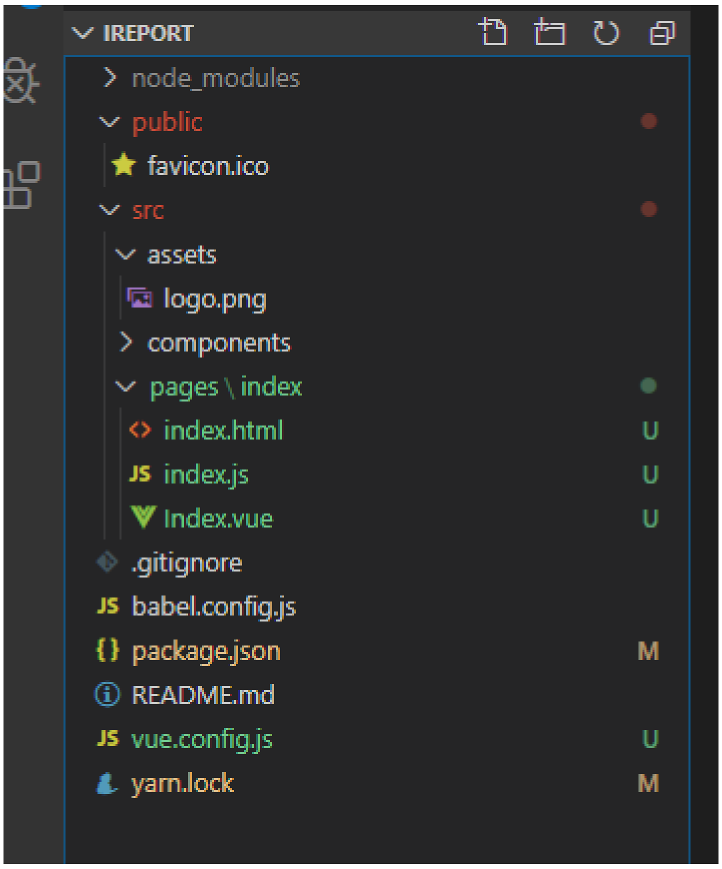
修改工程目录:

- 修改之前

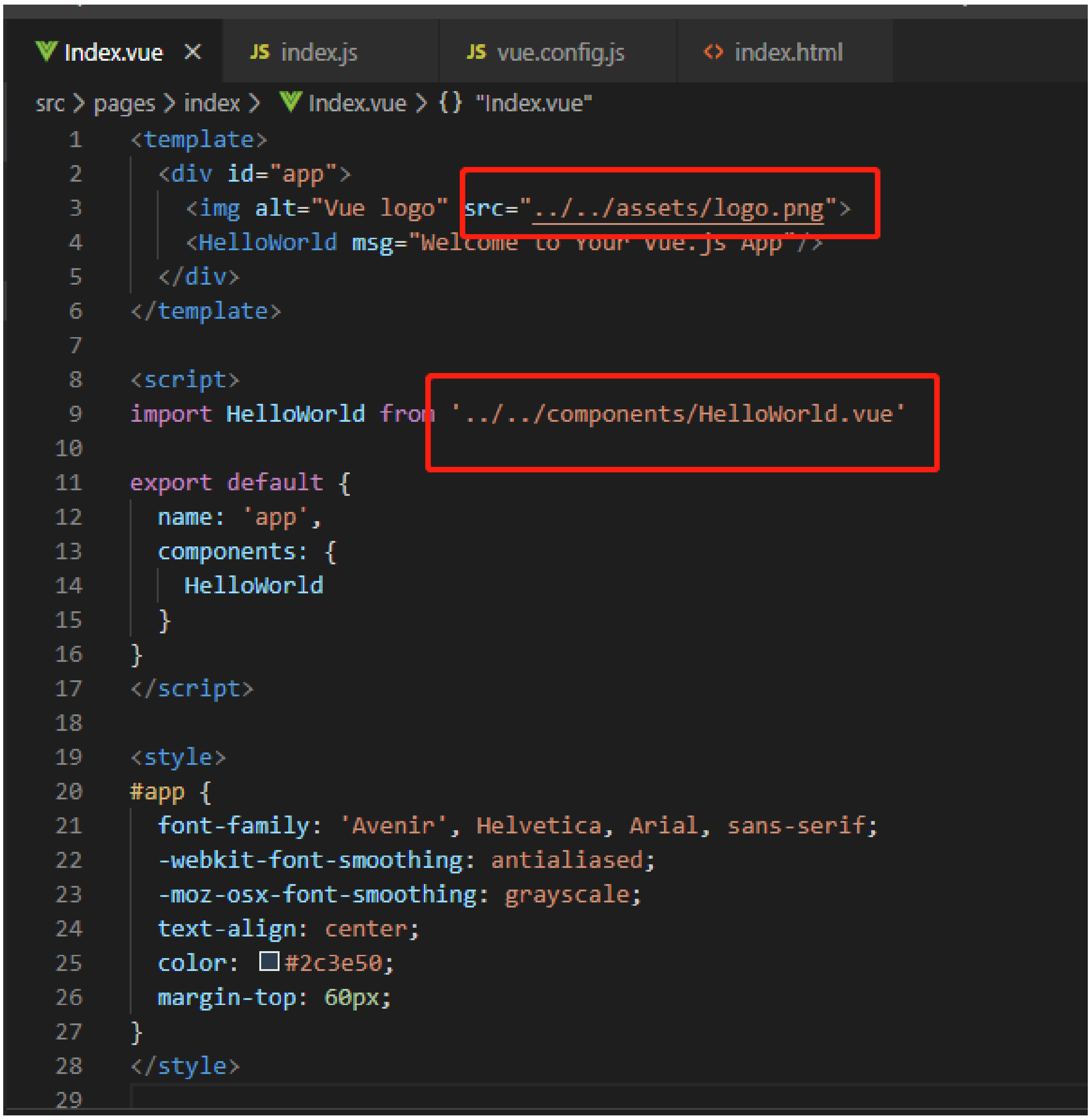


```
EXPLORER
OPEN EDITORS
IREPORT
  node_modules
  public
    favicon.ico
    index.html
  src
    assets
      logo.png
    components
      HelloWorld.vue
    App.vue
    main.js
  .gitignore
  babel.config.js
  package.json
  README.md
  yarn.lock
```

- 修改之后



- 修改相对路径Index.vue

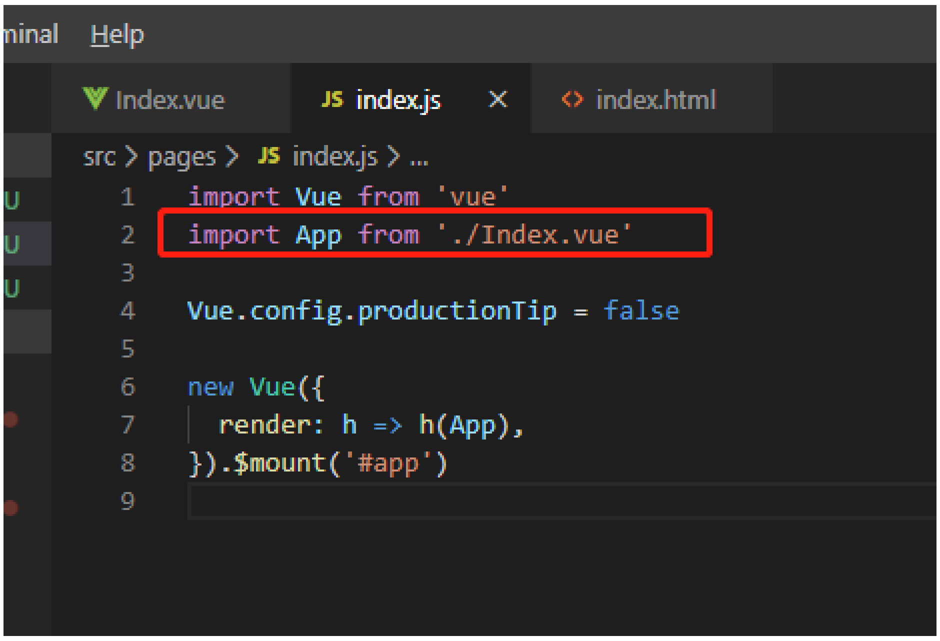


Index.vue × | index.js | vue.config.js | index.html

src > pages > index > Index.vue > {} "Index.vue"

```
<template>
  <div id="app">
    <img alt="Vue logo" src="../../assets/logo.png">
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>

<script>
import HelloWorld from '../../components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>

<style>
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

- 修改相对路径index.js



Index.vue | index.js × | index.html

src > pages > index.js > ...

```
import Vue from 'vue'
import App from './Index.vue'

Vue.config.productionTip = false

new Vue({
  render: h => h(App),
}).$mount('#app')
```

- **根目录添加vue.config.js文件**

```
module.exports = {
    pages: {
        index: {
          entry: './src/pages/index/index.js',
          template: './src/pages/index/index.html',
          filename: 'index.html',
          title: '首页'
        }
    },
    devServer: {
      host: '0.0.0.0',
      port: 9000
    },
    publicPath: process.env.NODE_ENV === 'production' ? '/ireport/dist/':
'/'
}
```

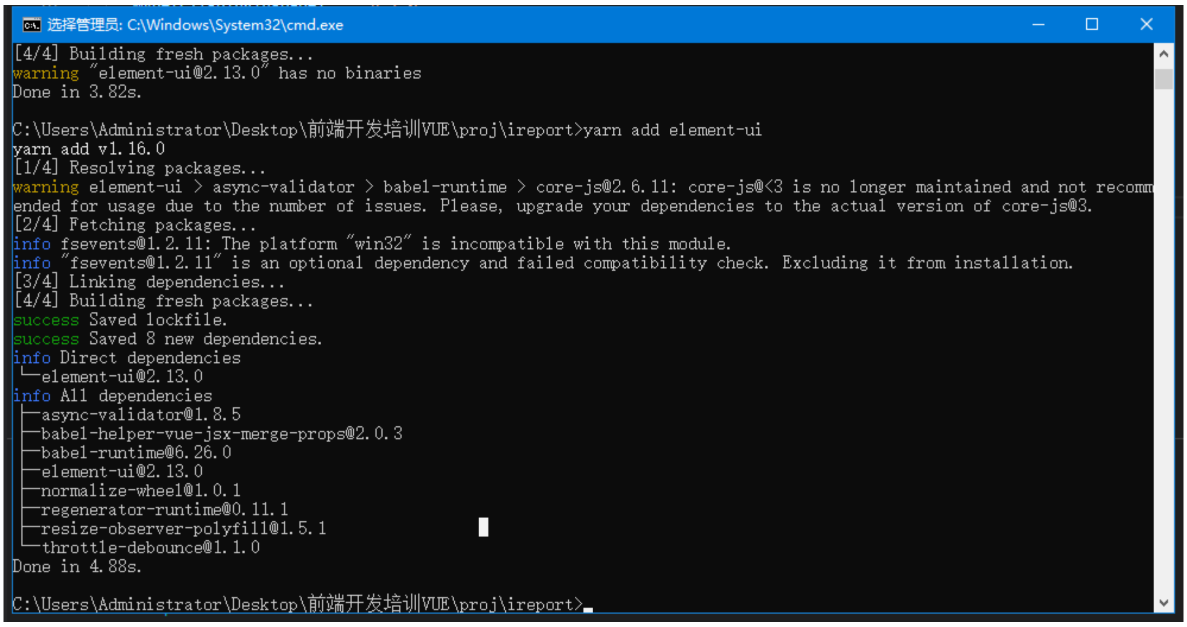
重启工程查看工程是否正常运行。

3.3.1 【练习】再添加一个页面?

3.4 添加ElementUI

- 添加依赖

```
yarn add element-ui
```



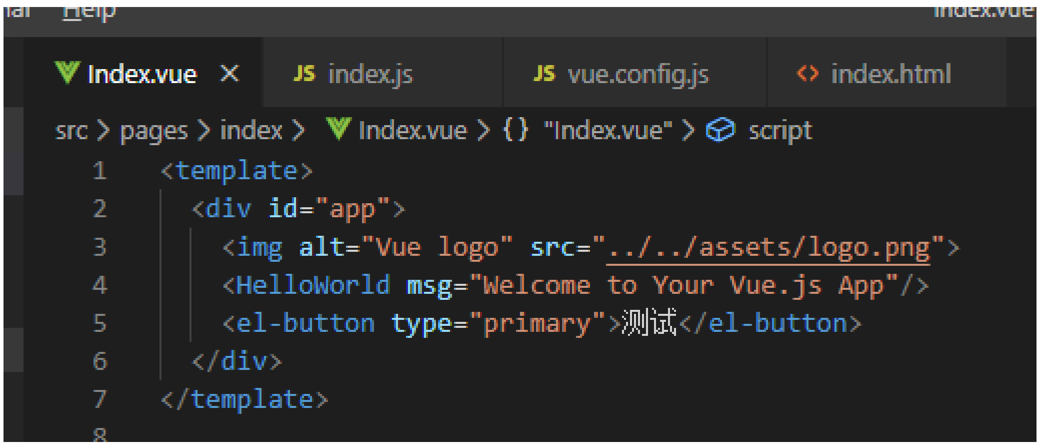
```
[4/4] Building fresh packages...
warning "element-ui@2.13.0" has no binaries
Done in 3.82s.

C:\Users\Administrator\Desktop\前端开发培训VUE\proj\ireport>yarn add element-ui
yarn add v1.16.0
[1/4] Resolving packages...
warning element-ui > async-validator > babel-runtime > core-js@2.6.11: core-js@<3 is no longer maintained and not recomm
ended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
[2/4] Fetching packages...
info fsevents@1.2.11: The platform "win32" is incompatible with this module.
info "fsevents@1.2.11" is an optional dependency and failed compatibility check. Excluding it from installation.
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 8 new dependencies.
info Direct dependencies
└─element-ui@2.13.0
info All dependencies
├─async-validator@1.8.5
├─babel-helper-vue-jsx-merge-props@2.0.3
├─babel-runtime@6.26.0
├─element-ui@2.13.0
├─normalize-wheel@1.0.1
├─regenerator-runtime@0.11.1
├─resize-observer-polyfill@1.5.1
└─throttle-debounce@1.1.0
Done in 4.88s.

C:\Users\Administrator\Desktop\前端开发培训VUE\proj\ireport>_
```

- 代码添加

  参照ElementUI的快速上手文档在工程代码中添加代码。

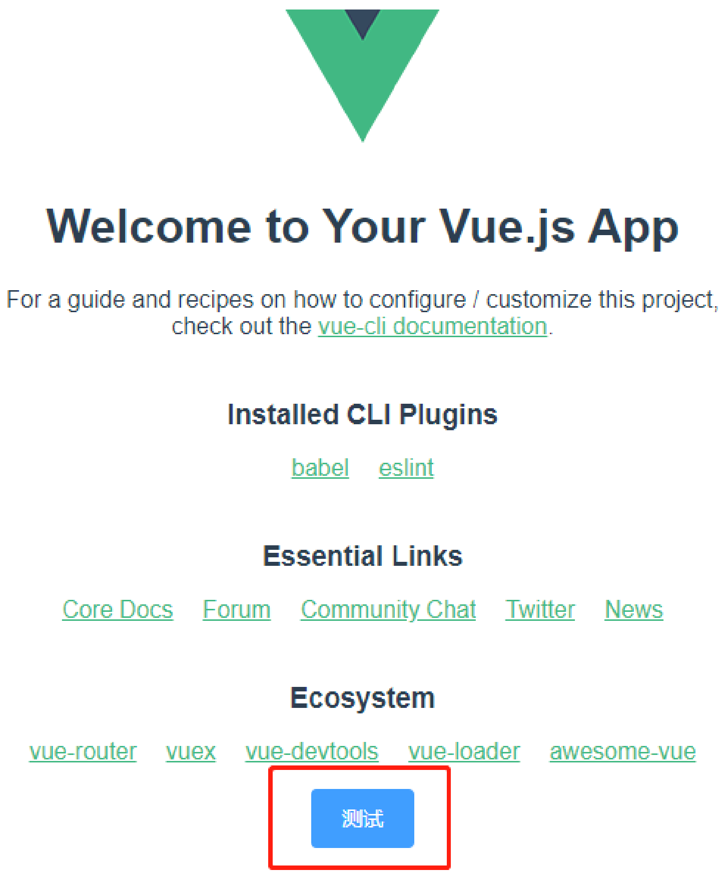
```
import Vue from 'vue';
import ElementUI from 'element-ui';
import 'element-ui/lib/theme-chalk/index.css';
import App from './App.vue';

Vue.use(ElementUI);

new Vue({
  el: '#app',
  render: h => h(App)
});
```

- 添加ElementUI组件验证是否成功



```
<template>
  <div id="app">
    <img alt="Vue logo" src="../../assets/logo.png">
    <HelloWorld msg="Welcome to Your Vue.js App"/>
    <el-button type="primary">测试</el-button>
  </div>
</template>
```

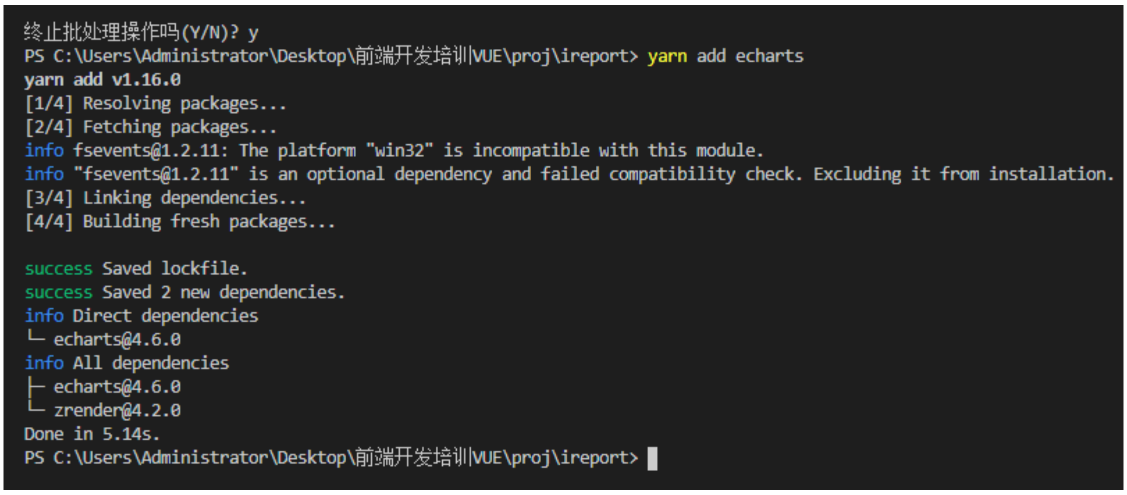
- 正常的执行结果



3.5 添加Echarts

- 添加依赖

```
yarn add echarts
```



```
终止批处理操作吗(Y/N)? y
PS C:\Users\Administrator\Desktop\前端开发培训\VUE\proj\ireport> yarn add echarts
yarn add v1.16.0
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@1.2.11: The platform "win32" is incompatible with this module.
info "fsevents@1.2.11" is an optional dependency and failed compatibility check. Excluding it from installation.
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
success Saved 2 new dependencies.
info Direct dependencies
└─ echarts@4.6.0
info All dependencies
├─ echarts@4.6.0
└─ zrender@4.2.0
Done in 5.14s.
PS C:\Users\Administrator\Desktop\前端开发培训\VUE\proj\ireport>
```

- 修改app.js文件，此处为index.js文件：

```
import Vue from 'vue'
import ElementUI from 'element-ui';
import 'element-ui/lib/theme-chalk/index.css';
import App from './Index.vue'
import echarts from 'echarts'

Vue.prototype.$echarts = echarts

Vue.config.productionTip = false

Vue.use(ElementUI);

new Vue({
  render: h => h(App),
}).$mount('#app')
```

- echarts的添加参考官方例子：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>ECharts</title>
    <!-- 引入 echarts.js -->
    <script src="echarts.min.js"></script>
</head>
<body>
    <!-- 为ECharts准备一个具备大小（宽高）的Dom -->
    <div id="main" style="width: 600px;height:400px;"></div>
    <script type="text/javascript">
        // 基于准备好的dom，初始化echarts实例
        var myChart = echarts.init(document.getElementById('main'));

        // 指定图表的配置项和数据
        var option = {
            title: {
                text: 'ECharts 入门示例'
            },
            tooltip: {},
            legend: {
                data:['销量']
            },
            xAxis: {
                data: ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
            },
            yAxis: {},
            series: [{
                name: '销量',
                type: 'bar',
                data: [5, 20, 36, 10, 10, 20]
            }]
        };

        // 使用刚指定的配置项和数据显示图表。
        myChart.setOption(option);
    </script>
</body>
</html>
```

4. 组件

4.1 简单组件添加

- components文件中添加一个文件CompOne.vue

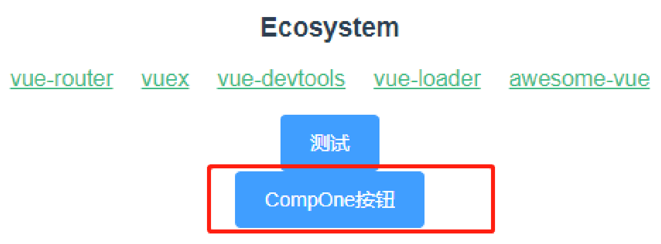
```
<template>
    <div>
        <el-button type="primary">CompOne按钮</el-button>
    </div>
</template>
<script>
export default {
    name: 'CompOne'
}
</script>
```

- 在Index.vue引用组件:

```
<template>
  <div id="app">
    <img alt="Vue logo" src="../../assets/logo.png">
    <HelloWorld msg="Welcome to Your Vue.js App"/>
    <el-button type="primary">测试</el-button>
    <CompOne />
  </div>
</template>

<script>
import HelloWorld from '../../components/HelloWorld.vue'
import CompOne from '../../components/CompOne.vue'
```

```
export default {
  name: 'app',
  components: {
    HelloWorld,
    CompOne
  }
}
</script>
```

- 检查组件是否安装成功：

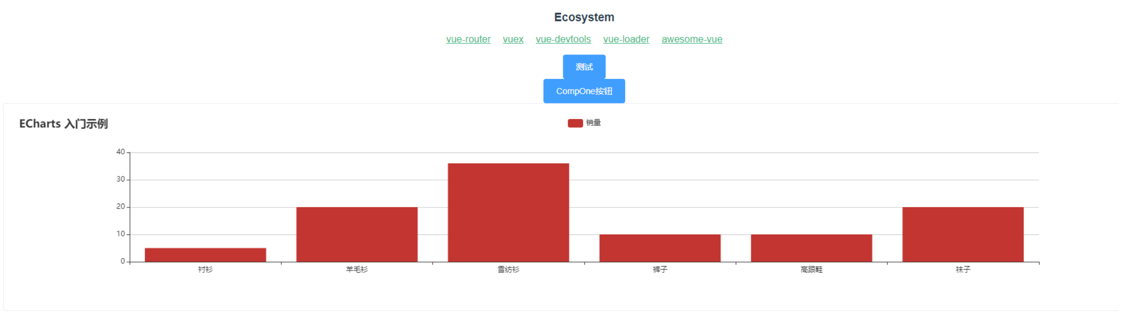


4.2 在component中添加Echarts图表

- 在components文件夹中添加组件SimpleBarChart.vue

```
<template>
    <el-card shadow="never">
        <div id="main" ref="refMainChart"></div>
    </el-card>
</template>
<script>
export default {
    name: "SimpleBarChart",
    mounted() {
        this.drawLine()
    },
    methods: {
        drawLine(){
            // 基于准备好的dom，初始化echarts实例
            let myChart = this.$echarts.init(this.$refs.refMainChart)
            myChart.clear()
            // 绘制图表
            let option = {
                title: {
                    text: 'ECharts 入门示例'
                },
                tooltip: {},
                legend: {
                    data:['销量']
                },
                xAxis: {
                    data: ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
                },
                yAxis: {},
                series: [{
                    name: '销量',
                    type: 'bar',
                    data: [5, 20, 36, 10, 10, 20]
                }]
            };
            myChart.setOption(option);
        }
    },
}
</script>

<style scoped>
#main {
    width: 100%;
    height: 300px;
}
</style>
```

- 查看bar图标是否添加成功



4.2.1 【练习】在页面中添加Echarts图表?

4.3 向子组件中传递数据

通过props的方式向component传递数据。

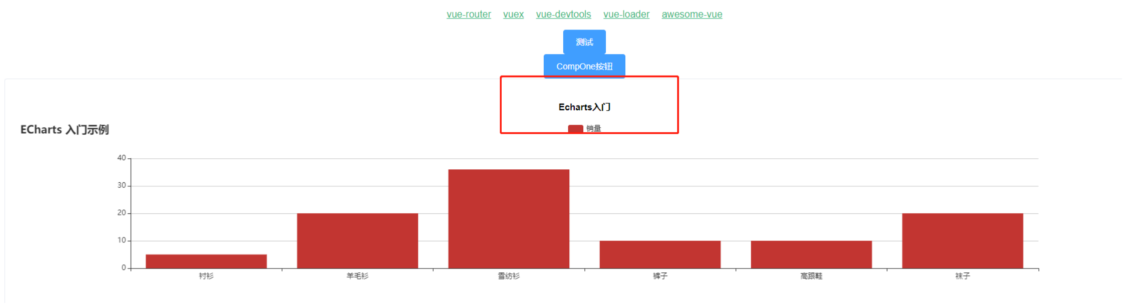
- 修改SimpleBarChart.vue

```
<template>
    <el-card shadow="never">
        <p class="chart-title">{{chartTitle}}</p>
        <div id="main" ref="refMainChart"></div>
    </el-card>
</template>
<script>
export default {
    name: "SimpleBarChart",
    props: ['chartTitle'],
    mounted() {
        this.drawLine()
    },
    // ...
```

- 修改Index.vue，向SimpleBarChart.vue传递数据

```
<SimpleBarChart chartTitle="Echarts入门"/>
```

- 检查属性设置结果



4.4 emit向父级组件传递消息

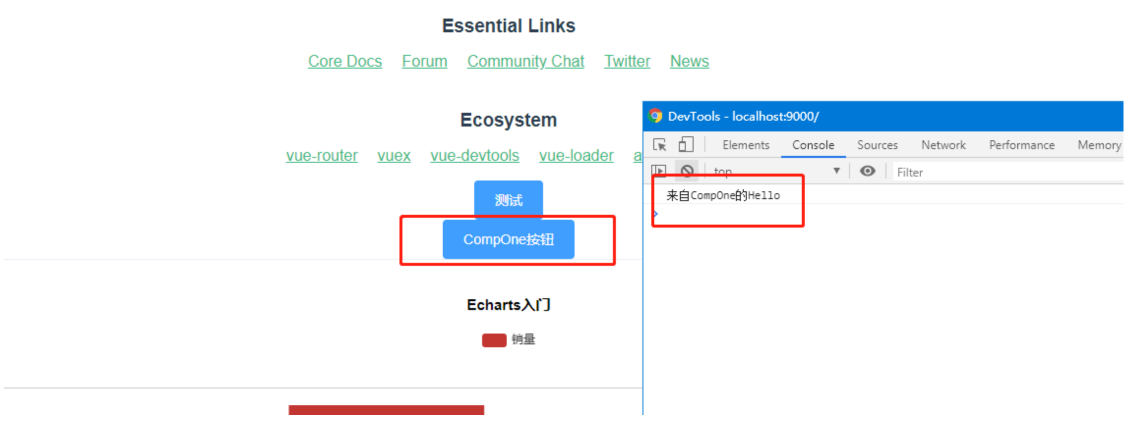
- 修改CompOne.vue，添加按钮点击事件函数，发出消息

```
<template>
    <div>
        <el-button type="primary" @click="onBtnClick">CompOne按钮</el-
button>
    </div>
</template>

<script>
export default {
    name: 'CompOne',
    methods: {
        onBtnClick() {
            this.$emit('sayHello', '来自CompOne的Hello')
        }
    }
}
</script>
```

- 修改Index.vue，绑定消息，添加函数，打印参数内容

```
<template>
  <div id="app">
    <HelloWorld msg="Welcome to Your Vue.js App"/>
    <el-button type="primary">测试</el-button>
    <CompOne v-on:sayHello="onSayHello"/>
    <SimpleBarChart chartTitle="Echarts入门"/>
  </div>
</template>

<script>
import HelloWorld from '../../components/HelloWorld.vue'
import CompOne from '../../components/CompOne.vue'
import SimpleBarChart from '../../components/SimpleBarChart.vue'

export default {
  name: 'app',
  components: {
    HelloWorld,
    CompOne,
    SimpleBarChart
  },
  methods: {
    onSayHello(params) {
      console.log(params)
    }
  }
}
</script>
```

- 查看结果



```
来自CompOne的Hello
```

- 解决Unexpected console statement (no-console)报错
  在package.json文件中配置rules

```
    "rules": {
      "no-console":"off"
    }
```

4.5 父组件通过refs调用子组件的方法

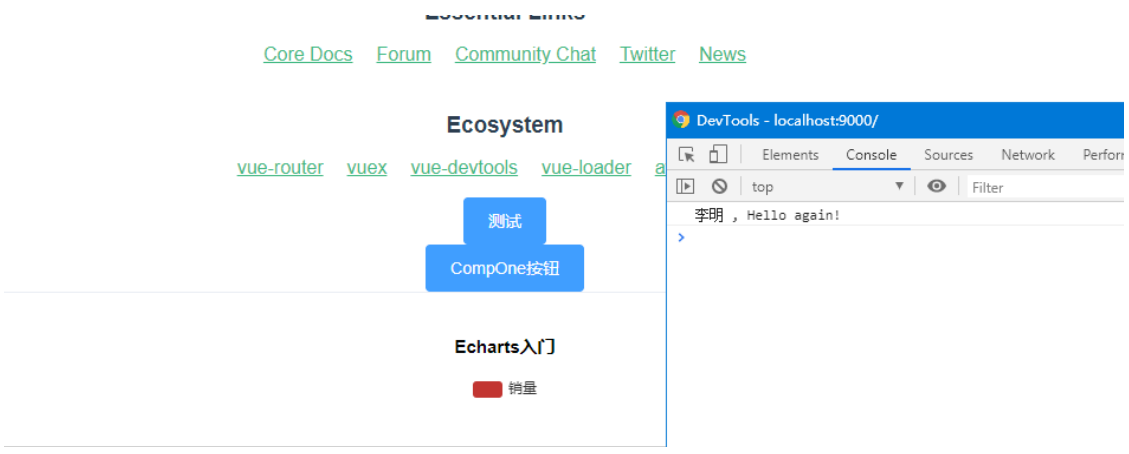
- 修改CompOne.vue添加方法

```
<template>
    <div>
        <el-button type="primary" @click="onBtnClick">CompOne按钮</el-
button>
    </div>
</template>

<script>
export default {
    name: 'CompOne',
    methods: {
        onBtnClick() {
            this.$emit('sayHello', '来自CompOne的Hello')
        },
        printHello(name) {
            console.log(name, ', Hello again!')
        }
    }
}
</script>
```

- 修改index.vue，通过refs调用

```
<template>
  <div id="app">
    <HelloWorld msg="Welcome to Your Vue.js App"/>
    <el-button type="primary" @click="callCompOneFunc">测试</el-button>
    <CompOne v-on:sayHello="onSayHello" ref="refCompOne"/>
    <SimpleBarChart chartTitle="Echarts入门"/>
  </div>
</template>

<script>
import HelloWorld from '../../components/HelloWorld.vue'
import CompOne from '../../components/CompOne.vue'
import SimpleBarChart from '../../components/SimpleBarChart.vue'

export default {
  name: 'app',
  components: {
    HelloWorld,
    CompOne,
    SimpleBarChart
  },
  methods: {
    onSayHello(params) {
      console.log(params)
    },
    callCompOneFunc() {
      this.$refs.refCompOne.printHello('李明')
    }
  }
}
</script>
```

- 查看调用结果



```
李明 , Hello again!
```

5. 添加axios组件库

Axios，基于 Promise 的 HTTP 客户端，可以工作于浏览器中，也可以在 node.js 中使用。

功能：

- 从浏览器中创建 XMLHttpRequest
- 从 node.js 中创建 http 请求
- 支持 Promise API
- 拦截请求和响应
- 转换请求和响应数据
- 取消请求
- 自动转换 JSON 数据
- 客户端支持防止 XSRF 攻击

示例代码：

- 执行一个 GET 请求

```
// Make a request for a user with a given ID
axios.get('/user?ID=12345')
  .then(function (response) {
    console.log(response);
  })
  .catch(function (error) {
    console.log(error);
  });
// Optionally the request above could also be done as
axios.get('/user', {
    params: {
      ID: 12345
    }
  })
  .then(function (response) {
    console.log(response);
  })
  .catch(function (error) {
    console.log(error);
  });
```

- 执行一个 POST 请求

```
axios.post('/user', {
    firstName: 'Fred',
    lastName: 'Flintstone'
  })
  .then(function (response) {
    console.log(response);
  })
  .catch(function (error) {
    console.log(error);
  });
```

- 执行多个并发请求

```
function getUserAccount() {
  return axios.get('/user/12345');
}
function getUserPermissions() {
  return axios.get('/user/12345/permissions');
}
axios.all([getUserAccount(), getUserPermissions()])
  .then(axios.spread(function (acct, perms) {
    // Both requests are now complete
  }));
```

6. 练习

- 利用axios从服务端请求数据，并用Echarts创建一个时间趋势图表。
- 从服务端请求数据，利用ElementUI中的table组件创建一个表格。

附录：

示例代码github地址：https://github.com/bold881/vuet1.git