# 1. 使用React脚手架创建一个React应用

## 1). react脚手架

```
1. xxx脚手架: 用来帮助程序员快速创建一个基于xxx库的模板项目
    * 包含了所有需要的配置
    * 指定好了所有的依赖
    * 可以直接安装/编译/运行一个简单效果
2. react提供了一个专门用于创建react项目的脚手架库: create-react-app
3. 项目的整体技术架构为: react + webpack + es6  + babel + eslint
```

## 2). 创建项目并启动

```
npm install -g create-react-app
create-react-app react-app
cd react-app
npm start
```

## 3). 使用脚手架开发的项目的特点

```
模块化: js是一个一个模块编写的
组件化: 界面是由多个组件组合编写实现的
工程化: 实现了自动构建/运行/打包的项目
```

## 4). 组件化编写项目的流程

```
拆分组件
实现静态组件--->静态页面
实现动态组件
    动态显示初始化数据
    交互
```

## 2). 拆分组件

```
App
    * state: searchName/string
Search
    * props: setSearchName/func
List
    * props: searchName/string
    * state: firstView/bool, loading/bool, users/array, errMsg/string
```

## 3). 编写组件

```
编写静态组件
编写动态组件
    componentWillReceiveProps(nextProps): 监视接收到新的props, 发送ajax
    使用axios库发送ajax请求
```

# 2. 组件间通信总结

## 1). 方式一: 通过props传递

```
共同的数据放在父组件上, 特有的数据放在自己组件内部(state)
一般数据-->父组件传递数据给子组件-->子组件读取数据
函数数据-->子组件传递数据给父组件-->子组件调用函数
问题: 多层传递属性麻烦, 兄弟组件通信不方便
```

## 2). 方式二: 使用消息订阅(subscribe)-发布(publish)机制: 自定义事件机制

```
工具库: PubSubJS
下载: npm install pubsub-js --save
使用:
  import PubSub from 'pubsub-js' //引入
  PubSub.subscribe('delete', function(msg, data){ }); //订阅
  PubSub.publish('delete', data) //发布消息
优点: 可以支持任意关系组件之间的通信
```

## 3). 事件监听理解

```
1. DOM事件
    * 绑定事件监听
        * 事件名(类型): 只有有限的几个, 不能随便写
        * 回调函数
    * 用户操作触发事件(event)
        * 事件名(类型)
        * 数据
2. 自定义事件
    * 绑定事件监听
        * 事件名(类型): 任意
        * 回调函数: 通过形参接收数据, 在函数体处理事件
    * 触发事件(编码)
        * 事件名(类型): 与绑定的事件监听的事件名一致
        * 数据: 会自动传递给回调函数
```

## 4) . react应用中的ajax请求

```
json-server: 模拟数据的
安装:  npm install json-server -g
启动
1）cd mock文件下
2）输入：json-server json数据的名字 --port 4000
```

```
axios: 包装XMLHttpRequest对象, promise风格, 支持浏览端/node服务器端
安装:  npm install --save axios

执行 GET 请求
// 为给定 ID 的 user 创建请求
axios.get('/user?ID=12345')
  .then(function (response) {
    console.log(response);
  })
  .catch(function (error) {
    console.log(error);
  });
// 上面的请求也可以这样做
axios.get('/user', {
    params: {
      ID: 12345
    }
  })
  .then(function (response) {
    console.log(response);
  })
  .catch(function (error) {
    console.log(error);
  });

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

##

# 3. ES6新语法总结

```
定义变量/常量: const/let
解构赋值: let {a, b} = this.props   import {aa} from 'xxx'
对象的简洁表达: {a, b}
箭头函数:
    组件的自定义方法: xxx = () => {}
    map/filter的回调方法: (item, index) => {}
    优点:
        * 简洁
        * 没有自己的this,使用引用this查找的是外部this
扩展运算符: ...
    拆解对象:  const MyProps = {}, <Xxx {...MyProps}>
类: class/extends/constructor/super
ES6模块化: export default | import
```

# 4. 项目打包运行

```
npm run build  //生成打包文件
npm install -g serve  //全局下载服务器包
serve build  //通过服务器命令运行打包项目
访问: http://localhost:5000  //浏览器访问
```