

实验4-数据表示层架构的设计与实现

郭 勇





实验4-数据表示层的设计与实现

●实验目的

- 学习数据表示层的设计原理
- 学会使用React进行数据表示层的设计及实现
- 结合实验3的内容，将React应用到实验3的项目中进行表示层设计及实现





什么是React

- React 是一个用于构建用户界面的 JavaScript 库。
- React 主要用于构建UI，React 是 MVC 中的 V（视图）。
- React 起源于 Facebook 的内部项目，用来架设 Instagram 的网站，并于 2013 年 5 月开源。
- React 拥有较高的性能，代码逻辑非常简单





React 特点

- 声明式设计 - React采用声明范式，可以轻松描述应用。
- 高效 - React通过对DOM的模拟，最大限度地减少与DOM的交互。
- 灵活 - React可以与已知的库或框架很好地配合。
- JSX - JSX 是 JavaScript 语法的扩展。React 开发不一定使用 JSX ，但我们建议使用它。
- 组件 - 通过 React 构建组件，使得代码更加容易得到复用，能够很好的应用在大项目的开发中。
- 单向响应的数据流 - React 实现了单向响应的数据流，从而减少了重复代码，这也是它为什么比传统数据绑定更简单





React 安装

● React 主要库

- react.js: React的核心库
- react-dom.js: 提供操作DOM的扩展库
- babel.min.js: 解析JSX语法代码转为纯JS语法代码的库





React 安装

- React 可以直接下载使用，下载包中也提供了很多学习的实例。
- 你可以在官网 <https://reactjs.org/> 下载最新版。
- 你也可以直接使用 Staticfile CDN 的 React CDN 库，地址如下：

```
<script  
src="https://cdn.staticfile.org/react/16.4.0/umd/react.development.js">  
</script>  
<script src="https://cdn.staticfile.org/react-dom/16.4.0/umd/react-  
om.development.js"></script>  
<!-- 生产环境中不建议使用 -->  
<script src="https://cdn.staticfile.org/babel-  
standalone/6.26.0/babel.min.js"></script>
```





使用 React

● 安装 Node.js

■ 下载地址 <https://nodejs.org/en/>

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)

10.15.3 LTS	11.13.0 Current
Recommended For Most Users	Latest Features
Other Downloads Changelog API Docs	Other Downloads Changelog API Docs

Or have a look at the Long Term Support (LTS) schedule.

Sign up for Node.js Everywhere, the official Node.js Monthly Newsletter.





使用 React

●npm 使用介绍

■npm:node package management

■npm是随同NodeJS一起安装的包管理工具，能解决NodeJS代码部署上的很多问题，常见的使用场景有以下几种：

- ◆允许用户从pm服务器**下载**别人编写的**第三方包**到本地使用。
- ◆允许用户从npmM服务器**下载**并安装别人编写的**命令行程序**到本地使用。
- ◆允许用户将自己编写的包或命令行程序**上传**到npm服务器供别人使用。
- ◆由于新版的nodejs已经集成了npm，所以之前npm也一并安装好了。同样可以通过输入 "npm -v" 来测试是否成功安装。





使用 React

● 安装 create-react-app

- create-react-app 是用于搭建 react 项目的脚手架。
- 它的优势在于省略了很多涉及配置的地方，让新手能够更加容易上手，减低入门的门槛。

■ 安装方法：

`npm install -g create-react-app yarn`

查看全局安装路径

`npm root -g`





使用 React

●使用create-react-app 创建项目

■创建项目框架

■进入命令行，选择要保存的项目目录，执行以下命令：

`create-react-app my_app`

■create-react-app 将会在当前目录下创建目录`my-app`，初始化项目并安装依赖。





使用 React

●启动创建的项目

- 项目创建完成后，进入my_app目录，可以在idea的terminal使用命令。
- 启动项目：npm start
- 后台显示如下信息，包含网络地址，在浏览器中输入给出的地址就可打开默认的主页面。

```
> react-scripts start

i [wds]: Project is running at http://192.168.1.2/
Compiled successfully!

You can now view rdemo in the browser.

   Local:            http://localhost:3000
  On Your Network:  http://192.168.1.2:3000

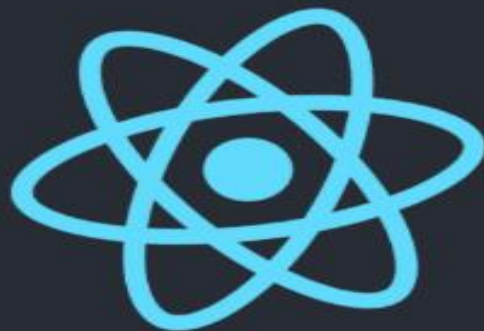
Note that the development build is not optimized.
To create a production build, use yarn build.
```





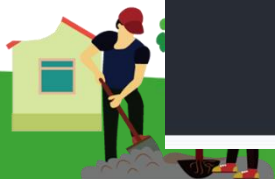
使用 React

- 直接启动，因为create-react-app已经配置好了项目，启动后打开浏览器输入<http://localhost:3000/>。
- 到此项目创建完成。



Edit `src/App.js` and save to reload.

[Learn React](#)





使用 React

● 项目目录结构

- |—— README.md 项目介绍
- |—— package.json npm包配置文件，里面定义了项目的npm脚本，依赖包等信息
- |—— package-lock 上线锁定版本信息
- |—— gitignore 这个是git的选择性上传的配置文件
- |—— src 源码目录 这个目录里边放的是我们开放的源代码
- |—— App.css
- |—— App.js 这个文件相当于一个方法模块，也是一个简单的模块化编程
- |—— App.test.js
- |—— index.css
- |—— index.js 这个就是项目的入口文件
- |—— serviceWorker.js 这个是由于写移动端开发的，PWA必须用到这个文件
- |—— public 公共文件，里边有公用模板和图标等一些东西
- |—— favicon.ico：这个是网站或者项目的图标，一般在浏览器标签页的左上角显示
- |—— index.html：首页的模板文件，我们可以试着改动一下，就能看到结果。
- |—— manifest.json：移动端配置文件。





使用 React

● React文档

■ <https://reactjs.org/tutorial/tutorial.html> 基于案例学习



谢谢观看

