link: null
title: 珠峰架构师成长计划
description: src\index.js
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=51 sentences=201, words=1680

1.初始化项目

```
yarn init -y
yarn add react react-dom lodash bootstrap  is-array reselect redux react-tiny-virtual-list

yarn add webpack webpack-cli webpack-dev-server html-webpack-plugin  optimize-css-assets-webpack-plugin babel-loader @babel/core @babel/preset-env
@babel/preset-react style-loader css-loader postcss-loader html-webpack-externals-plugin @babel/plugin-syntax-class-properties mini-css-extract-plugin --dev
```

2.编译阶段的优化

- 开发环境时重复构建更快
    - include
    - resolve
    - alias
    - external
    - 编译缓存
    - 开启多进程
- 生产环境时文件更小，加载更快
    - 开启tree-sharking
    - scope-hosting
    - splitChunks
    - 提供node的空mocks
    - 持久化缓存

```
const path = require('path');
const webpack = require('webpack');
const HtmlWebpackPlugin = require('html-webpack-plugin');
const TerserPlugin = require('terser-webpack-plugin');
const OptimizeCSSAssetsPlugin = require('optimize-css-assets-webpack-plugin');
const bootstrap = path.resolve('node_modules/bootstrap/dist/css/bootstrap.css');
const HtmlWebpackExternalsPlugin = require('html-webpack-externals-plugin');
const MiniCssExtractPlugin = require('mini-css-extract-plugin');
const shouldUseSourceMap = process.env.GENERATE_SOURCEMAP !== 'false';
module.exports = ({ development, production }) => {
    const isEnvDevelopment = development === 'development';
    const isEnvProduction = production === 'production';
    const getStyleLoaders = (cssOptions) => {
        const loaders = [
            isEnvDevelopment && require.resolve('style-loader'),
            isEnvProduction && MiniCssExtractPlugin.loader,
            {
                loader: require.resolve('css-loader'),
                options: cssOptions,
            },
            'postcss-loader',
        ].filter(Boolean);
        return loaders;
    };
    return {
        mode: isEnvProduction ? 'production' : isEnvDevelopment ? 'development' : 'development',
        devtool: isEnvProduction
            ? shouldUseSourceMap
                ? 'source-map'
                : false
            : isEnvDevelopment && 'cheap-module-source-map',
        cache: {
            type: 'filesystem'
        },
        entry: {
            main: './src/index.js'
        },
        optimization: {
            minimize: isEnvProduction,
            minimizer: [
                new TerserPlugin({ parallel: true }),
                new OptimizeCSSAssetsPlugin()
            ],
            splitChunks: {
                chunks: 'all',
                minSize: 0,
                minRemainingSize: 0,
                maxSize: 0,
                minChunks: 1,
                maxAsyncRequests: 30,
                maxInitialRequests: 30,
                enforceSizeThreshold: 50000,
                cacheGroups: {
                    defaultVendors: {
                        test: /[\\/]node_modules[\\/]/,
                        priority: -10,
                        reuseExistingChunk: true
                    },
                    default: {
                        minChunks: 2,
                        priority: -20,
                        reuseExistingChunk: true
                    }
                }
            },
```

```
            runtimeChunk: {
                name: entrypoint => `runtime-${entrypoint.name}`,
            },
            moduleIds: isEnvProduction ? 'deterministic' : 'named',
            chunkIds: isEnvProduction ? 'deterministic' : 'named'
        },
        resolve: {
            modules: [path.resolve('node_modules')],
            extensions: ['.js'],
            alias: {
                bootstrap
            },
            fallback: {
                crypto: false,
                buffer: false,
                stream: false
            }
        },
        module: {
            rules: [
                {
                    test: /\.js$/,
                    use: [
                        {
                            loader: 'babel-loader',
                            options: {
                                cacheDirectory: true,
                                presets: [
                                    "@babel/preset-react"
                                ],
                                plugins:[
                                    "@babel/plugin-proposal-class-properties"
                                ]
                            }
                        }
                    ],
                    include: path.resolve('src'),
                    exclude: /node_modules/
                },
                {
                    test: /\.css$/,
                    use: getStyleLoaders({ importLoaders: 1 })
                }
            ]
        },
        devServer: {},
        plugins: [
            new HtmlWebpackPlugin(
                Object.assign(
                    {},
                    {
                        inject: true,
                        template: './public/index.html'
                    },
                    isEnvProduction
                        ? {
                            minify: {
                                removeComments: true,
                                collapseWhitespace: true,
                                removeRedundantAttributes: true,
                                useShortDoctype: true,
                                removeEmptyAttributes: true,
                                removeStyleLinkTypeAttributes: true,
                                keepClosingSlash: true,
                                minifyJS: true,
                                minifyCSS: true,
                                minifyURLs: true,
                            },
                        }
                        : undefined
                )
            ),
            new HtmlWebpackExternalsPlugin({
                externals: [
                    {
                        module: 'lodash',
                        entry: "https://cdn.bootcdn.net/ajax/libs/lodash.js/4.17.20/lodash.js",
                        global: '_',
                    },
                ],
            }),
        ]
    }
}
```

src\index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
ReactDOM.render(
    <h1>helloh1>
    ,document.getElementById('root'));
```

public\index.html

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>reacttitle>
head>
<body>
    <div id="root">div>
body>
html>
```

package.json

```
{
   "scripts": {
       "build": "webpack --env=production",
       "start": "webpack serve --env=development"
   },
}
```

```
curl http:
curl http:
```

3.路由切换优化

```
import React from 'react';
import ReactDOM from 'react-dom';
import {HashRouter as Router,Route,Link} from 'react-router-dom';
import {dynamic} from './utils';
const LoadingHome = dynamic(()=>import('./components/Home'));
const LoadingUser = dynamic(()=>import('./components/User'));
ReactDOM.render(
    <Router>
        <ul>
            <li><Link to="/">HomeLink>li>
            <li> <Link to="/user">UserLink>li>
        ul>
        <Route path="/" exact={true} component={LoadingHome}/>
        <Route path="/user" component={LoadingUser}/>
    Router>
    ,document.getElementById('root'));
```

src\utils.js

```
const Loading = () => <div>Loadingdiv>;
export function dynamic(loadComponent) {
    const LazyComponent = lazy(loadComponent)
    return () => (
        <React.Suspense fallback={<Loading />}>
            <LazyComponent />
        React.Suspense>
    )
}
function lazy(load) {
    return class extends React.Component {
        state = { Component: null }
        componentDidMount() {
            load().then(result => {
                this.setState({ Component: result.default});
            });
        }
        render() {
            let { Component } = this.state;
            return Component && <Component />;
        }
    }
}
```

4.更新阶段优化

src\index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
ReactDOM.render(
<App/>
,document.getElementById('root'));
```

src\App.js

```
import React from 'react';
import {PureComponent,memo} from './utils';
export default class App extends React.Component{
  constructor(props){
    super(props);
    this.state = {title:'计数器',number:0}
  }
  add = (amount)=>{
    this.setState({number:this.state.number+amount});
  }
  render(){
    console.log('App render');
    return (
      <div>
        <Counter number={this.state.number}/>
        <button onClick={()=>this.add(1)}>+1button>
        <button onClick={()=>this.add(0)}>+0button>
        <ClassTitle title={this.state.title}/>
        <FunctionTitle title={this.state.title}/>
      div>
    )
  }
}
class Counter extends PureComponent{
  render(){
    console.log('Counter render');
    return (
     <p>{this.props.number}p>
    )
  }
}
class ClassTitle extends PureComponent{
  render(){
    console.log('ClassTitle render');
    return (
     <p>{this.props.title}p>
    )
  }
}
const FunctionTitle = memo(props=>{
  console.log('FunctionTitle render');
  return  <p>{props.title}p>;
});
```

src\utils.js

```
import React from 'react';
export class PureComponent extends React.Component{
    shouldComponentUpdate(nextProps,nextState){
        return !shallowEqual(this.props,nextProps)||!shallowEqual(this.state,nextState)
    }
}
export function memo(OldComponent){
    return class extends PureComponent{
      render(){
        return <OldComponent {...this.props}/>
      }
}
}
export function shallowEqual(obj1,obj2){
    if(obj1 === obj2)
        return true;
    if(typeof obj1 !== 'object' || obj1 ===null || typeof obj2 !== 'object' || obj2 ===null){
        return false;
    }
    let keys1 = Object.keys(obj1);
    let keys2 = Object.keys(obj2);
    if(keys1.length !== keys2.length){
        return false;
    }
    for(let key of keys1){
        if(!obj2.hasOwnProperty(key) || obj1[key]!== obj2[key]){
            return false;
        }
    }
    return true;
}
```

src\App.js

```
import React from 'react';
import {PureComponent} from './utils';
+import { Map } from "immutable";
export default class App extends React.Component{
  constructor(props){
    super(props);
+   this.state = {count:Map({ number: 0 })}
  }
  add = (amount)=>{
+   let count = this.state.count.set('number',this.state.count.get('number') + amount);
+   this.setState({count});
  }
  render(){
    console.log('App render');
    return (

        this.add(1)}>+1
        this.add(0)}>+0

    )
  }
}
class Counter extends PureComponent{
  render(){
    console.log('Counter render');
    return (
     {this.props.number}
    )
  }
}
```

src\utils.js

```
import React from 'react';
+import { Map,is } from "immutable";

export class PureComponent extends React.Component{
    shouldComponentUpdate(nextProps,nextState){
        return !shallowEqual(this.props,nextProps)||!shallowEqual(this.state,nextState)
    }
}
export function memo(OldComponent){
    return class extends PureComponent{
      render(){
        return
      }
}
}
export function shallowEqual(obj1,obj2){
    if(obj1
        return true;
    if(typeof obj1 !== 'object' || obj1
        return false;
    }
    let keys1 = Object.keys(obj1);
    let keys2 = Object.keys(obj2);
    if(keys1.length !== keys2.length){
        return false;
    }
    for(let key of keys1){
+       if (!obj2.hasOwnProperty(key) || !is(obj1[key],obj2[key])) {
            return false;
        }
    }
    return true;
}
```

```
import {createStore} from 'redux';
import { createSelector } from 'reselect';
let initialState = {
    count:{number:0},
    todos:[{text:'没完成的事',completed:false},{text:'完成的事',completed:true}],
    filter:true
};
const reducer = (state=initialState,action)=>{
  switch(action.type){
      case 'ADD':
        return {...state,count:{number:state.count.number+1}};
      default:
          return state;
  }
}
let store = createStore(reducer);
export const todosSelector = (state) => state.todos;
export const filterSelector = (state) => state.filter;
export const visibleTodosSelector = createSelector(
  [todosSelector,filterSelector],
  (todos,filter)=>{
    console.log('计算visibleTodos');
    return todos.filter(item=>item.completed == filter);
  }
);
const render = ()=>{
    let state = store.getState();
    console.log(state);
    const state1 = visibleTodosSelector(state);
    console.log(state1);
}
store.subscribe(render);
render();
store.dispatch({type:'ADD'});
```

5.大数据渲染

src\Home.js

```
import React from 'react';
export default class Home extends React.Component{
    state={
       list: []
    }
    handleClick=()=>{
       let starTime = new Date().getTime();
       this.setState({
           list: new Array(30000).fill(0)
       },()=>{
          const end =  new Date().getTime()
          console.log( (end - starTime ) / 1000 + '秒')
       })
    }
    render(){
        return (
            <ul>
              <button onClick={ this.handleClick }>点击button>
              {
                  this.state.list.map((item,index)=>(
                    <li  key={index} >{ index}li>
                  ))
              }
            ul>
        )
    }
}
```

src\Home.js

```
import React from 'react';
export default class Home extends React.Component{
    state={
       list: []
    }
+   handleClick=()=>{
+      this.timeSlice(550);
+   }
+   timeSlice = (times)=>{
+       //requestIdleCallback
+       requestAnimationFrame(()=>{
+         let minus = times>=100?100:times;
+         times-=minus;
+         this.setState({
+             list:[...this.state.list,...new Array(minus).fill(0)]
+         },()=>{
+             if(times>0){
+                 this.timeSlice(times);
+             }
+         });
+       });
+   }
    render(){
        return (

              点击
              {
                  this.state.list.map((item,index)=>(
                    {index+1}
                  ))
              }

        )
    }
}
```

```
import React from 'react';
import { render } from 'react-dom';

import VirtualList from './components/VirtualList';
const data = new Array(30).fill(0);

render(
    <VirtualList
        width='50%'
        height={500}
        itemCount={data.length}
        itemSize={50}
        renderItem={(data) => {
            let { index, item, style } = data;
            console.log(data);
            return (
                <div key={index} style={{ ...style, backgroundColor: index % 2 === 0 ? 'green' : 'orange' }}>
                    {index+1}
                div>
            )
        }
        }
    />,
    document.getElementById('root')
);
```

src\components\VirtualList.js

```
import React from 'react';
export default class Index extends React.Component {
    scrollBox = React.createRef()
    state = {start: 0}
    handleScroll = () => {
        const { itemSize } = this.props;
        const { scrollTop } = this.scrollBox.current;
        const start = Math.floor(scrollTop / itemSize);
        this.setState({start})
    }
    render() {
        const { height, width, itemCount, itemSize, renderItem } = this.props;
        const { start } = this.state;
        let end = start + Math.floor(height/itemSize)+1;
        end = end>itemCount?itemCount:end;
        const visibleList = new Array(end - start).fill(0).map((item,index)=>({index:start+index}));
        const style = {position:'absolute',top:0,left:0,width:'100%', height: itemSize};
        return (
            <div
                style={{overflow: 'auto',willChange:'transform', height,width}}
                ref={this.scrollBox}
                onScroll={this.handleScroll}
            >
                <div style={{position: 'absolute',width:'100%',height: `${itemCount * itemSize}px`}}>
                   {
                        visibleList.map(({index}) => renderItem({ index, style:{...style,top:itemSize*index} }))
                    }
                div>
            div>
        )
    }
}
```

6.React 性能分析器

- React 16.5 增加了对新的开发者工具 DevTools 性能分析插件的支持
- 此插件使用 React 实验性的 Profiler API 来收集有关每个组件渲染的用时信息，以便识别 React 应用程序中的性能瓶颈
- react-devtools (https://fb.me/react-devtools) 将为支持新的 `Profiler API` 的应用显示 `Profiler` 选项卡
- 浏览 commits（Browsing commits）
- 过滤 commits（Filtering commits）
- 火焰图表（Flame chart）
- 排序图表（Ranked chart）

7.其它性能优化

- React hooks性能优化
- 响应式数据的精细化渲染
- 通过DOM-DIFF原理进行性能优化
- Error Boundaries
- 骨架屏
- 预渲染
- 图片懒加载
- ……