

ssr预习

资源

<https://ssr.vuejs.org>

概念

服务端渲染：将vue实例渲染为HTML字符串直接返回，在前端激活为交互程序

优点

- seo 如果产品对网页搜索要求较高，则在服务端渲染会比较好
- 首屏内容到达时间
服务端渲染的话由于是HTML字符串直接返回，所以是一次首屏，一次渲染
spa的话是一次请求，一次响应，结构完成之后还需要ajax请求数据渲染，起码要两次来回

服务端知识

express

```
npm i express -S
```

基础http服务

```
// nodejs代码
const express = require('express')

// 获取express实例
const server = express()

// 编写路由处理不同url请求
server.get('/', (req, res)=>{
  res.send('<strong>hello world</strong>')
})

// 监听端口
server.listen(80, () => {
  console.log('server running!');
})
```

基础实现

使用渲染器将vue实例成HTML字符串并返回

开课吧web全栈架构师

安装vue-server-renderer

```
npm i vue vue-server-renderer -S
```

同时安装vue 和 vue-server-renderer
确保版本相同且匹配

使用vsr

```
// 1.创建vue实例
const Vue = require('vue')
const app = new Vue({
  template: '<div>hello world</div>'
})

// 2.获取渲染器实例
const { createRenderer } = require('vue-server-renderer')
const renderer = createRenderer()

// 3.用渲染器渲染vue实例 此处app是vue实例
renderer.renderToString(app).then(html => {
  console.log(html);
}).catch(err => {
  console.log(err);
})
```

整合express

```
// nodejs代码
const express = require('express')

// 获取express实例
const server = express()

const Vue = require('vue')
// 2.获取渲染器实例
const { createRenderer } = require('vue-server-renderer')
const renderer = createRenderer()

// 编写路由处理不同url请求
server.get('/', (req, res) => {
  // res.send('<strong>hello world</strong>')
  // 1.创建vue实例
  const app = new Vue({
    template: '<div @click="onClick">{{msg}}</div>',
    data() {
      return {msg: 'vue SSR'}
    },
    methods: {
```

```
onClick() {  
  console.log('do something');  
}  
},  
})  
  
// 3.用渲染器渲染vue实例  
renderer.renderToString(app).then(html => {  
  res.send(html)  
}).catch(err => {  
  res.status(500)  
  res.send('Internal Server Error, 500!')  
})  
})  
  
// 监听端口  
server.listen(80, () => {  
  console.log('server running!');  
})  
})
```

现在只有死的内容，如果要使程序可以交互还需要激活
我们将在正课带大家~