

全部课程 (/courses/) / 用Vue.js 和 vue-router 创建单页导航和分页 (/courses/762) / 在vue-route路由组件中进行分页

在线实验，请到PC端体验

vue-router2.0 在路由组件中使用分页

一、实验简介

1.1 实验内容

本实验在路由的组件中使用分页功能。

1.2 实验知识点

- props
- created
- computed

1.3 实验环境

- sublime
- ubuntu

1.4 适合人群

本课程难度为一般，属于初级级别课程，适合具有 html 和 js 经验的小伙伴。

1.5 代码获取

本实验使用所有代码均已上传

<http://labfile.oss.aliyuncs.com/courses/762/vue-route2.zip>

二、课程介绍

vue-router 是 Vue.js 官方的路由插件，适合用于构建单页面应用。vue 的单页面应用是基于路由和组件的，路由用于设定访问路径，并将路径和组件映射起来。在组件中数据如何进行分页呢，我们一起来实验吧。

三、实验步骤

上节实验我们已经介绍了 vue-router 路由，本实验继续延续上节实验，

我们在给页面导航加上登录注册的效果。

```
<ul class="nav navbar-nav navbar-right">
  <li><router-link to="/login">注册</router-link></li>
  <li><router-link to="/reg">登录</router-link></li>
</ul>
```

加上注册和登录两条路由

动手实践是学习 IT 技术最有效的方式！

开始实验

```
var router = new VueRouter({
  routes: [
    { path: '/home', component: Home },
    { path: '/courses', component: Courses ,
      children:[
        {path:'all', component:coursesall},
        {path:'priview', component:coursespriview},
        {path:'develop', component:coursesdevelop}
      ]
    },
    { path: '/paths', component: Paths },
    { path: '/bootcamp', component: Bootcamp },
    { path: '/login', component: Login },
    { path: '/reg', component: Reg }
  ]
})
```

登录注册的组件内容,样式是 bootstrap 里面的基本样式,就不仔细说了,大家可以去学习 bootstrap 课程,本实验只是简单的把页面展现出来,不进行实现。

```
const Login = { template: '<div><div class="col-md-offset-3 col-md-6"><form><span>用户登录</span><div class="form-group"><input type="email" class="form-control" placeholder="用户名或电子邮件"><i class="fa fa-user"></i></div><div class="form-group help"><input type="password" class="form-control" placeholder="密 码"><i class="fa fa-lock"></i></div><a href="#" class="fa fa-question-circle"></a><div class="form-group"><button class="btn btn-md btn-primary btn-block" type="submit">登录</button></div></form></div></div></div>' }
const Reg = { template: '<div><div class="col-md-offset-3 col-md-6"><form><span>用户注册</span><div class="form-group"><input type="email" class="form-control" placeholder="用户名或电子邮件"><i class="fa fa-user"></i></div><div class="form-group help"><input type="password" class="form-control" placeholder="密 码"><i class="fa fa-lock"></i></div><a href="#" class="fa fa-question-circle"></a><div class="form-group"><button class="btn btn-md btn-primary btn-block" type="submit">注册</button></div></form></div></div></div>' }
```

我们往课程里添加数据, 这里为全部课程添加了一组数据。

```
data:{
  course[
    {coursetitle:"新手指南之玩转实验楼",coursecontent:"新手指南之玩转实验楼",coursehit:76471},
    {coursetitle:"Linux 基础入门",coursecontent:"Linux 基础入门",coursehit:93388},
    {coursetitle:"C语言实现大数计算器",coursecontent:"C语言实现大数计算器",coursehit:167},
    {coursetitle:"C++实现智能指针",coursecontent:"C++实现智能指针",coursehit:163},
    {coursetitle:"PHP 封装分页类",coursecontent:"PHP 封装分页类",coursehit:140}
  ],
}
```

我们要在子组件中使用 course 中的数据, 在 router-view 中把数据传到子路由中。可以使用

```
<router-view :course="course"></router-view>
```

当然你也可以自定义, 我们在子路由中使用 course, 还需要使用 props, 用于接收来自父组件的数据。父组件向子组件传递数据, 因为每一个组件都是独立的作用域, 所以要把父组件的数据在子组件中显示, 要使用 props 定义属性来绑定父组件里面的数据。

```
props:['course']
```

绑定到页面中我们使用 v-for, 在前面已经学到过 v-for 的使用方法。

```
const coursesall={template:'<div><div v-for="(course,index) in course">{{index+1}}<div>{{course.coursetitle}}<div>{{course.coursehit}}</div></div>',props:['course']};
```

现在绑定后数据效果如下

我们给添加个简单的样式。

```
const coursesall={template:'<div><table class="table table-bordered"><thead><tr><th>序号</th><th>课程</th><th>关注</th></tr></thead><tbody><tr v-for="(course,index) in course"><td>{{index+1}}</td><td>{{course.coursetitle}}</td><td>{{course.coursehit}}</td></tr></tbody></table></div></div>',props:['course']};
```

优化后代码保存为 vuedemo2-1.html ,

动手实践是学习 IT 技术最有效的方式!

开始实验

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>实验楼--vue-route使用</title>
  <link href="./bootstrap.min.css" rel="stylesheet">
  <script src="./vue.js"></script>
  <script src="./vue-router.js"></script>
  <style>
    .router-link-active{
      background-color: #e3e3e3 !important;
      color: #ff0000 !important;
      border-radius:5px;
    }
    .move-enter-active, .move-leave-active {
      transition: opacity .5s
    }
    .move-enter, .move-leave-active {
      opacity: 0
    }
    form{
      max-width:330px;
      padding:15px;
    }

    button{
      max-width: 100px;
      padding:5px;
    }
    input{
      margin: 3px;
    }

  </style>
</head>
<body>
<div id="app">
  <nav class="navbar navbar-inverse">
    <div class="container">

      <div class="navbar-header">
        
      </div>
      <div id="navbar" class="collapse navbar-collapse">
        <ul class="nav navbar-nav">
          <li><router-link to="/home">首页</router-link></li>
          <li><router-link to="/courses">课程</router-link></li>
          <li><router-link to="/paths">路径</router-link></li>
          <li><router-link to="/bootcamp">训练营</router-link></li>
        </ul>
        <ul class="nav navbar-nav navbar-right">
          <li><router-link to="/login">登录</router-link></li>
          <li><router-link to="/reg">注册</router-link></li>
        </ul>
      </div>
    </div>
  </nav>
  <div class="container">
    <router-view :course="course"></router-view>
  </div>
</script>

const Home = { template: '<div><h1>首页</h1></div>' }
const Login = { template: '<div><div class="col-md-offset-3 col-md-6"><form><span>用户登录</span><div class="form-group"><input type="email" class="form-control" placeholder="用户名或电子邮件"><i class="fa fa-user"></i></div> <div class="form-group help"><input type="password" class="form-control" placeholder="密码"><i class="fa fa-lock"></i><a href="#" class="fa fa-question-circle"></a> <div class="form-group"><button class="btn btn-md btn-primary btn-block" type="submit">登录</button></div></form></div></div></div>' }
const Reg = { template: '<div><div class="col-md-offset-3 col-md-6"><form><span>用户注册</span><div class="form-group"><input type="email" class="form-control" placeholder="用户名或电子邮件"><i class="fa fa-user"></i></div> <div class="form-group help"><input type="password" class="form-control" placeholder="密码"><i class="fa fa-lock"></i><a href="#" class="fa fa-question-circle"></a> <div class="form-group"><button class="btn btn-md btn-primary btn-block" type="submit">注册</button></div></form></div></div></div>' }
const Courses = {template: '<div><ul class="nav nav-tabs"><li><router-link to="/courses/all">全部课程</router-link>

```

```

ink></li><li><router-link to="/courses/priview">即将上线</router-link></li><li><router-link to="/courses/develop">开发者</router-link></li></ul><div><router-view :course=course></router-view></div></div>',props:['course']}]
const Paths = { template: '<div><h1>路径</h1></div>' }
const Bootcamp = { template: '<div><h1>训练营</h1></div>' }

const coursesall={template:'<div><table class="table table-bordered"><thead><tr><th @click="columnSort(course title,toggleSortOrder, head-name)" id="head-name">序号</th><th>课程</th><th>关注</th></tr></thead><tbody><tr v-for="(course,index) in course"><td>{{index+1}}</td><td>{{course.coursetitle}}</td><td>{{course.coursehit}}</td></tr></tbody></table></div></div>',props:['course']};

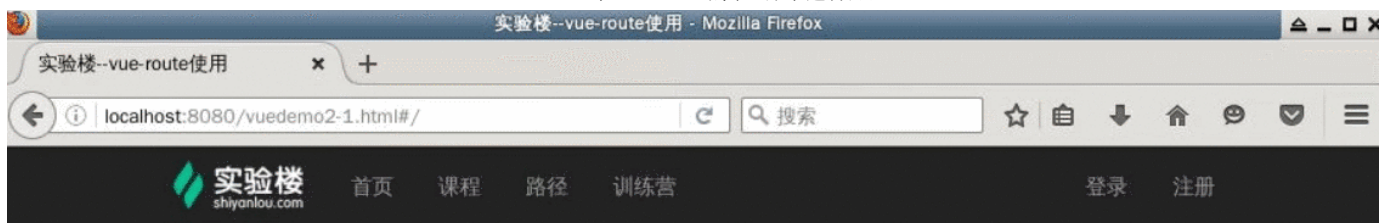
const coursespriview={template:'<div>我是即将上线</div>'};
const coursesdevelop={template:'<div>我是开发者</div>'};

var router = new VueRouter({
  routes: [
    { path: '/home', component: Home },
    { path: '/courses', component: Courses ,
      children:[
        {path:'all', component:coursesall},
        {path:'priview', component:coursespriview},
        {path:'develop', component:coursesdevelop}
      ]
    },
    { path: '/paths', component: Paths },
    { path: '/bootcamp', component: Bootcamp },
    { path: '/login', component: Login },
    { path: '/reg', component: Reg }
  ]
})

const app = new Vue({
  router,
  el:'#app',
  data:{
    course:[
      {coursetitle:'新手指南之玩转实验楼',coursecontent:'新手指南之玩转实验楼',coursehit:76471},
      {coursetitle:'Linux 基础入门',coursecontent:'Linux 基础入门',coursehit:93388},
      {coursetitle:'C语言实现大数计算器',coursecontent:'C语言实现大数计算器',coursehit:167},
      {coursetitle:'C++实现智能指针',coursecontent:'C++实现智能指针',coursehit:163},
      {coursetitle:'PHP 封装分页类',coursecontent:'PHP 封装分页类',coursehit:140}
    ]
  },
}).$mount('#app')
</script>
</body>
</html>

```

页面效果如下。



然后我们在全部课程中添加分页功能，我们需要在全部课程中添加分页，我们把方法，数据都写到组件中去。

修改的数据如下,在 coursesall 组件中加入数据，我们看到在组件中数据的加入和之前有点不同，在组件中的数据是一个函数，如果我们分页的话我们还需要定义每页的大小，还有默认的渲染。

```
const coursesall={template:'<div><table class="table table-bordered"><thead><tr><th>序号</th><th>课程</th><th>关注</th></tr></thead><tbody><tr v-for="(item,index) in arrayData"><td>{{index+1}}</td><td>{{item.name}}</td><td>{{item.age}}</td></tr></tbody></table><div class="pager" id="pager"><template v-for="item in pageCount+1"><span v-if="item==1" class="btn btn-default" v-on:click="showPage(pageCurrent-1,$event)">上一页</span><span v-if="item>0&&item<=pageCount-1&&item==showPagesStart&&item<=showPageEnd&&item<=pageCount" class="btn btn-default" v-on:click="showPage(item,$event)">{{item}}</span><span v-if="item==pageCount" class="btn btn-default" v-on:click="showPage(pageCurrent+1,$event)">下一页</span></template></div></div>',
  data(){
    return {
      course:[
        {name:'新手指南之玩转实验楼',age:76471},
        {name:'Linux 基础入门',age:93388},
        {name:'C语言实现大数计算器',age:167},
        {name:'C++实现智能指针',age:163},
        {name:'PHP 封装分页类',age:140},
        {name:'新手指南之玩转实验楼1',age:76471},
        {name:'Linux 基础入门1',age:93388},
        {name:'C语言实现大数计算器1',age:167},
        {name:'C++实现智能指针1',age:163},
        {name:'PHP 封装分页类1',age:140},
        {name:'新手指南之玩转实验楼2',age:76471},
        {name:'Linux 基础入门2',age:93388},
        {name:'C语言实现大数计算器2',age:167},
        {name:'C++实现智能指针2',age:163},
        {name:'C++实现智能指针2',age:163},
        {name:'C++实现智能指针3',age:163},
        {name:'C++实现智能指针3',age:163},
        {name:'C++实现智能指针3',age:163},
        {name:'PHP 封装分页类3',age:140}
      ],
      //当前页面
      pageCurrent: 1,
      //分页大小，每页有多少条
      pagesize : 5 ,
      //开始显示的分页按钮
      showPagesStart: 1,
      //结束显示的分页按钮
      showPageEnd: 10,
      //分页数据
      arrayData: [],
    }
  },
}
```

我们使用 computed 计算属性来计算分页数量。

动手实践是学习 IT 技术最有效的方式！

开始实验

```
computed:{
  pageCount: function () {
    //分页数量为数据的长度除以数据的分页大小取整后加一
    return (Math.ceil(this.course.length/this.pagesize))+1;
  },
}
```

然后是定义分页方法，具体实现就不写了。

```
methods:{
  //方法中传入页码
  showPage: function (pageIndex) {
    //判断页码
    if (pageIndex > 0) {
      if (pageIndex > this.pageCount) {
        pageIndex = this.pageCount;
      }
      //定义初始数据
      var newPageInfo = [];
      //循环输出每页内容
      for (var i = 0; i < this.pagesize; i++) {
        newPageInfo[newPageInfo.length] = {
          name: this.course[i + (pageIndex - 1)*5].name,
          age:this.course[i + (pageIndex - 1)*5].age
        };
      }
      this.pageCurrent = pageIndex;
      //数据渲染给arrayData
      this.arrayData = newPageInfo;
    }
  }
},
}
```

定义完分页方法后需要把数据在我们点击全部课程的时候初始化。

我们使用 vuejs 的一个方法 created 他的作用是实例已经创建完成之后被调用。

```
created: function (){
  this.showPage(1)
},
}
```

最后完成代码保存为 vuedemo2-2.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>实验楼--vue-route使用</title>
  <link href="./bootstrap.min.css" rel="stylesheet">
  <script src="./vue.js"></script>
  <script src="./vue-router.js"></script>
  <style>
    .router-link-active{
      background-color: #e3e3e3 !important;
      color: #ff0000 !important;
      border-radius:5px;
    }
    .move-enter-active, .move-leave-active {
      transition: opacity .5s
    }
    .move-enter, .move-leave-active {
      opacity: 0
    }

    form{
      max-width:330px;
      padding:15px;
    }

    button{
      max-width: 100px;
      padding:5px;
    }
    input{
      margin: 3px;
    }

  </style>
</head>
<body>
<div id="app">
  <nav class="navbar navbar-inverse">
    <div class="container">

      <div class="navbar-header">
        
      </div>
      <div id="navbar" class="collapse navbar-collapse">
        <ul class="nav navbar-nav">
          <li><router-link to="/home">首页</router-link></li>
          <li><router-link to="/courses">课程</router-link></li>
          <li><router-link to="/paths">路径</router-link></li>
          <li><router-link to="/bootcamp">训练营</router-link></li>
        </ul>
        <ul class="nav navbar-nav navbar-right">
          <li><router-link to="/login" >登录</router-link></li>
          <li><router-link to="/reg">注册</router-link></li>
        </ul>
      </div>
    </div>
  </nav>
  <div class="container">
<router-view></router-view>

    </div>
  </div>

<script>
const Home = { template: '<div><h1>首页</h1></div>' }
const Login = { template: '<div><div class="col-md-offset-3 col-md-6"><form><span>用户登录</span><div class="form-group"><input type="email" class="form-control" placeholder="用户名或电子邮件"><i class="fa fa-user"></i></div> <div class="form-group help"><input type="password" class="form-control" placeholder="密码"><i class="fa fa-lock"></i><a href="#" class="fa fa-question-circle"></a> <div class="form-group"><button class="btn btn-md btn-primary btn-block" type="submit">登录</button></div></form></div></div>' }
const Reg = { template: '<div><div class="col-md-offset-3 col-md-6"><form><span>用户注册</span><div class="form-group"><input type="email" class="form-control" placeholder="用户名或电子邮件"><i class="fa fa-user"></i></div> <div class="form-group help"><input type="password" class="form-control" placeholder="密码"><i class="fa fa-lock"></i><a href="#" class="fa fa-question-circle"></a> <div class="form-group"><button class="btn btn-md btn-primary btn-block" type="submit">注册</button></div></form></div></div>' }

```

```

-lock"></i><a href="#" class="fa fa-question-circle"></a> <div class="form-group"><button class="btn btn-md btn-primary btn-block" type="submit">注册</button></div></form></div></div></div>' }
const Courses = {template: '<div><ul class="nav nav-tabs"><li><router-link to="/courses/all">全部课程</router-link></li><li><router-link to="/courses/priview">即将上线</router-link></li><li><router-link to="/courses/develop">开发者</router-link></li></ul><div><router-view></router-view></div></div>'}
const Paths = { template: '<div><h1>路径</h1></div>' }
const Bootcamp = { template: '<div><h1>训练营</h1></div>' }

const coursesall={template:'<div><table class="table table-bordered"><thead><tr><th>序号</th><th>课程</th><th>关注</th></tr></thead><tbody><tr v-for="(item,index) in arrayData"><td>{{index+1}}</td><td>{{item.name}}</td><td>{{item.age}}</td></tr></tbody></table><div class="pager" id="pager"><template v-for="item in pageCount+1"><span v-if="item==1" class="btn btn-default" v-on:click="showPage(pageCurrent-1,$event)">上一页</span><span v-if="item>0&&item<=pageCount-1&&item<=showPagesStart&&item<=showPageEnd&&item<=pageCount" class="btn btn-default" v-on:click="showPage(item,$event)">{{item}}</span><span v-if="item==pageCount" class="btn btn-default" v-on:click="showPage(pageCurrent+1,$event)">下一页</span></template></div></div>',
data(){
  return {
    course:[
      {name: '新手指南之玩转实验楼',age:76471},
      {name: 'Linux 基础入门',age:93388},
      {name: 'C语言实现大数计算器',age:167},
      {name: 'C++实现智能指针',age:163},
      {name: 'PHP 封装分页类',age:140},
      {name: '新手指南之玩转实验楼1',age:76471},
      {name: 'Linux 基础入门1',age:93388},
      {name: 'C语言实现大数计算器1',age:167},
      {name: 'C++实现智能指针1',age:163},
      {name: 'PHP 封装分页类1',age:140},
      {name: '新手指南之玩转实验楼2',age:76471},
      {name: 'Linux 基础入门2',age:93388},
      {name: 'C语言实现大数计算器2',age:167},
      {name: 'C++实现智能指针2',age:163},
      {name: 'C++实现智能指针2',age:163},
      {name: 'C++实现智能指针3',age:163},
      {name: 'C++实现智能指针3',age:163},
      {name: 'C++实现智能指针3',age:163},
      {name: 'PHP 封装分页类3',age:140}
    ],
    //当前页面
    pageCurrent: 1,
    //分页大小, 每页有多少条
    pagesize: 5,
    //开始显示的分页按钮
    showPagesStart: 1,
    //结束显示的分页按钮
    showPageEnd: 10,
    //分页数据
    arrayData: [],
  },
},
computed:{
  pageCount: function () {
    return (Math.ceil(this.course.length/this.pagesize))+1;
  }
},
methods:{
  //方法中传入页码
  showPage: function (pageIndex) {
    //判断页码
    if (pageIndex > 0) {
      if (pageIndex > this.pageCount) {
        pageIndex = this.pageCount;
      }
      定义初始数据
      var newPageInfo = [];
      //循环输出每页内容
      for (var i = 0; i < this.pagesize; i++) {
        newPageInfo[newPageInfo.length] = {
          name: this.course[i + (pageIndex - 1)*5].name,
          age:this.course[i + (pageIndex - 1)*5].age
        };
        动手实践是学习IT技术最有效的方式!
      }
    }
  }
}

```

开始实验


```
    }
    this.pageCurrent = pageIndex;
    //数据渲染给arrayData
    this.arrayData = newPageInfo;
  }

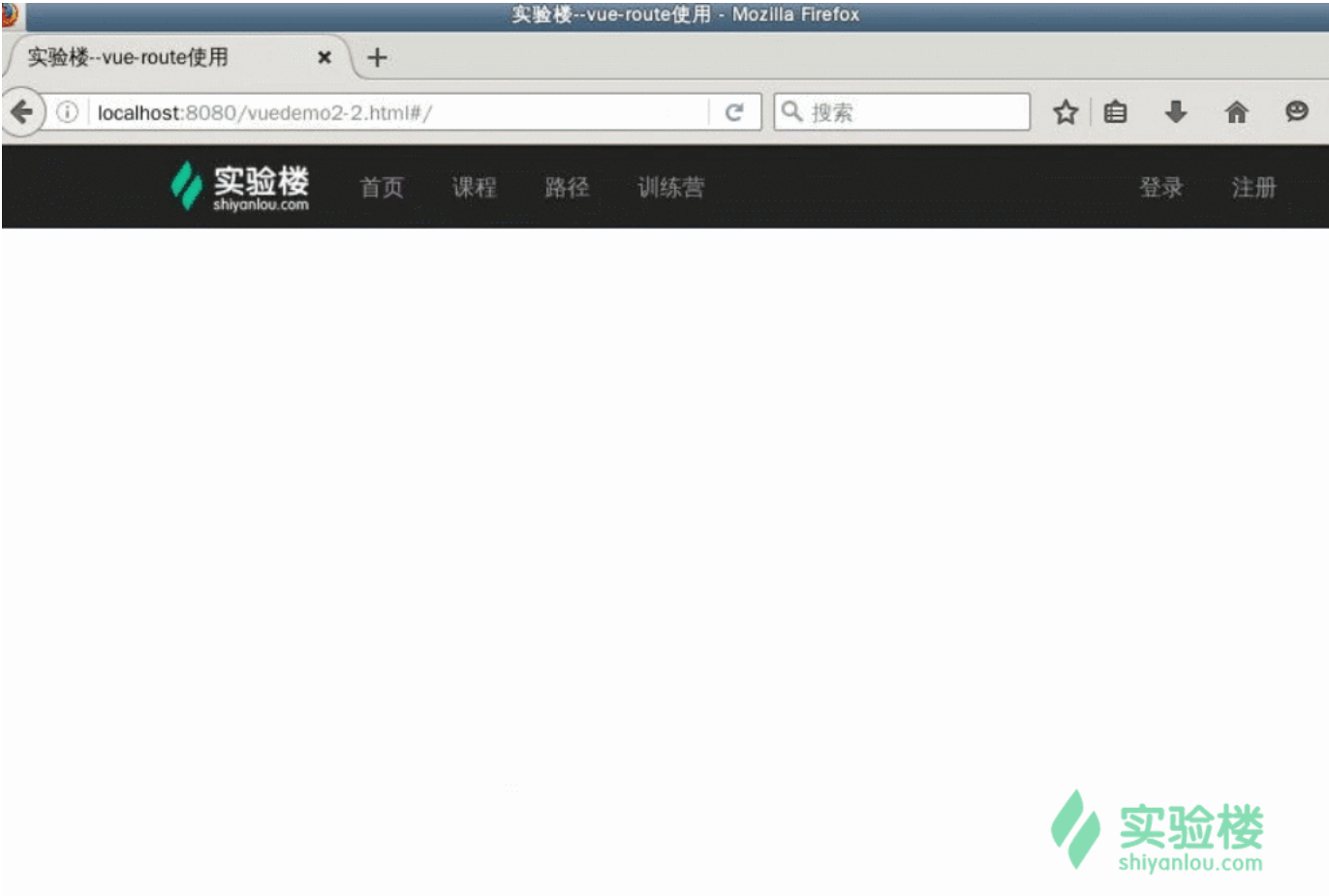
}
},
created: function () {
  this.showPage(1)
},
}

const coursespriview={template:'<div>我是即将上线</div>'};
const coursesdevelop={template:'<div>我是开发者</div>'};

var router = new VueRouter({
  routes: [
    { path: '/home', component: Home },
    { path: '/courses', component: Courses ,
      children:[
        {path:'all', component:coursesall},
        {path:'priview', component:coursespriview},
        {path:'develop', component:coursesdevelop}
      ]
    },
    { path: '/paths', component: Paths },
    { path: '/bootcamp', component: Bootcamp },
    { path: '/login', component: Login },
    { path: '/reg', component: Reg }
  ]
})

var app = new Vue({
  router,
  el: '#app',
}).$mount('#app');
</script>
</body>
</html>
```

页面效果如下



本次实验就先到这里。

四、实验总结

通过本次实验，大家可以简单的了解了在子路由中如何使用分页，以及数据如何传递到组件中。希望本次实验可以帮到小伙伴，大家回去可以把代码优化一下，把分页的代码更完善下，例如把分页最大值给使用计算属性计算出来。而不是固定的值。

五、参考链接

参考链接 [vue-route2.0官方文档 \(http://router.vuejs.org/zh-cn/\)](http://router.vuejs.org/zh-cn/)

实验楼感谢您对我们的信任与支持！

[< 上一节 \(/courses/762/labs/2554/document\)](#)

[下一节 > \(/courses/762/labs/2616/document\)](#)

课程教师



Adward 工程师
某信息安全公司
共发布过8门课程

[查看老师的所有课程 > \(/teacher/79217\)](#)



动手做实验，轻松学IT



公司 <http://weibo.com/shiyanlou2013>

合作

[关于我们 \(/aboutus\)](#)

[我要投稿 \(/contribute\)](#)

[联系我们 \(/contact\)](#)

[教师合作 \(/labs\)](#)

[高校合作 \(/edu/\)](#)

动手实践是学习 IT 技术最有效的方式 [开始实验](#)

2017/6/26

[加入我们 \(http://www.simplecloud.cn/jobs.html\)](http://www.simplecloud.cn/jobs.html)

[技术博客 \(https://blog.shiyanlou.com\)](https://blog.shiyanlou.com)

服务

[企业版 \(/saas\)](/saas)

[实战训练营 \(/bootcamp/\)](/bootcamp/)

[会员服务 \(/vip\)](/vip)

[实验报告 \(/courses/reports\)](/courses/reports)

[常见问题 \(/questions/?tag=%E5%B8%B8%E8%A7%81%E9%97%AE%E9%A2%98\)](/questions/?tag=%E5%B8%B8%E8%A7%81%E9%97%AE%E9%A2%98)

[隐私条款 \(/privacy\)](/privacy)

在vue-route路由组件中进行分页

[友情链接 \(/friends\)](/friends)

[开发者 \(/developer\)](/developer)

学习路径

[Python学习路径 \(/paths/python\)](/paths/python)

[Linux学习路径 \(/paths/linuxdev\)](/paths/linuxdev)

[大数据学习路径 \(/paths/bigdata\)](/paths/bigdata)

[Java学习路径 \(/paths/java\)](/paths/java)

[PHP学习路径 \(/paths/php\)](/paths/php)

[全部 \(/paths/\)](/paths/)

Copyright ©2013-2017 实验楼在线教育 | 蜀ICP备13019762号 (<http://www.miibeian.gov.cn/>)