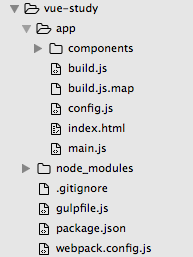
vuejs+webpack+gulp+Laravel 实现简单留言板

1. 前端部署

最近刚学习 vuejs，如有不对，请指出。下面开始。这个小项目也算是一个简单的前后分离案例吧（笑）。看之前一定要了解 webpack+gulp+vuejs+laravel。vuejs 文档至少要过几遍，没多少。

我们前端创建目录的如下:



gulpfile.js 为 gulp 配置文件， webpack.config.js 为 webpack 配置文件。 webpack 和 gulp 想想了解的可以百度快速入门，这里不详细赘述。 gulp 主要是用来监听文件变化自动打包的。 webpack 使用了 vue-loader 来解析编译 vuejs，官方的包。所有用到的 npm 包都在 package.json 文件中。

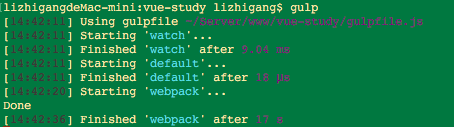
我们先看下 webpack.config.js:

|  |
| --- |
| var webpack = require('webpack');  module.exports = {  entry: './app/main.js', //入口js  output: {  path: './app', //编译输出目录  publicPath: '/app/',  filename: 'build.js' //输出文件名  },  module: {  // avoid webpack trying to shim process  noParse: /es6-promise\.js$/,  loaders: [  {  test: /\.vue$/,  loader: 'vue'  },  {  test: /\.js$/,  // excluding some local linked packages.  // for normal use cases only node\_modules is needed.  exclude: /node\_modules|vue\/dist|vue-router\/|vue-loader\/|vue-hot-reload-api\//,  loader: 'babel'  }  ]  },  babel: {  presets: ['es2015'],  plugins: ['transform-runtime']  }  } |

gulp 主要负责监听打包，配置代码如下：

|  |
| --- |
| var gulp = require('gulp'),  watch = require('gulp-watch');  var webpack = require("webpack");  var batch = require('gulp-batch');  var webpackConfig = require("./webpack.config.js");  //gulp --product  gulp.task('default', ['watch']);  gulp.task('watch',function(){  watch('app/components/\*.vue',batch(function (events, done) {  gulp.start('webpack', done);  }));  watch('app/components/partials/\*.vue',batch(function (events, done) {  gulp.start('webpack', done);  }));  watch('app/main.js',batch(function (events, done) {  gulp.start('webpack', done);  }));  });  //webpack静态处理  gulp.task('webpack', function(callback) {  //webpack配置文件  var config = Object.create(webpackConfig);  webpack(config, function(err, stats) {  // console.log(stats.toString());  console.log('Done');  callback();  });  }); |

这样我们就可以监听文件变化，自动打包发布了:



接下来创建 main.js, 这里使用了官方的路由包 vue-router 和 http 包 vue-resource:

|  |
| --- |
| import Vue from "vue";  import Router from "vue-router";  import App from "./components/App.vue";  import Home from "./components/Home.vue";  import Comment from "./components/Comment.vue";  import VueResource from 'vue-resource';  Vue.use(VueResource);  Vue.use(Router);  var router=new Router();  router.map({  '/':{  name:'home',  component:Home  },  '/:page':{  name:'home-comments',  component:Home  },  '/comment':{  name:'comment',  component:Comment  }  });  router.redirect({  '\*':'/'  });  router.start(App,'#app'); |

由于使用了 vue-loader 来解析，所以每一个页面都是一个组件。App.vue 代码如下:

|  |
| --- |
| <template>  <div id="wrapper">  <!--挂载点-->  <router-view>  </router-view>  <footer>  <p>©CopyRight:lizhigang 2016</p>  </footer>  </div>  </template>  <script>  export default{  name:'App'  }  </script>  <style>  header,footer{  height:50px;  width:100%;  }  header h3{  color:red;  text-align: center;  line-height: 50px;  }  footer p{  color:green;  text-align: center;  line-height: 50px;  }  </style> |

入口 index.html 代码如下:

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Loading....</title>  <!-- 新 Bootstrap 核心 CSS 文件 -->  <link rel="stylesheet" href="http://cdn.bootcss.com/bootstrap/3.3.0/css/bootstrap.min.css">  <!-- 可选的Bootstrap主题文件（一般不用引入） -->  <link rel="stylesheet" href="http://cdn.bootcss.com/bootstrap/3.3.0/css/bootstrap-theme.min.css">  <!-- jQuery文件。务必在bootstrap.min.js 之前引入 -->  <script src="http://cdn.bootcss.com/jquery/1.11.1/jquery.min.js"></script>  <!-- 最新的 Bootstrap 核心 JavaScript 文件 -->  <script src="http://cdn.bootcss.com/bootstrap/3.3.0/js/bootstrap.min.js"></script>  </head>  <body>  <div id="app"></div>  <script type="text/javascript" src="./build.js"></script>  </body>  </html> |

这样一个基本的前端单页应用结构就搭好了。

我们要实现以下这样的：







2. 后端部署

后端使用的是 laravel5.3。

后端只写下简单的 api 即可，还是很简单的，这里没涉及到 auth 啥的，所以比较简单，有兴趣的可以研究下 auth 验证，jwt 啥的，这里就不做了。

创建留言表 comments:

|  |
| --- |
| CREATE TABLE `comments` (  `id` int(10) unsigned NOT NULL AUTO\_INCREMENT,  `nickname` varchar(32) NOT NULL DEFAULT '',  `content` text NOT NULL,  `ip` varchar(15) NOT NULL DEFAULT '',  `created\_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',  `updated\_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',  PRIMARY KEY (`id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8; |

模型 app/Models/Api:

|  |
| --- |
| <?php  namespace App\Models\Api;  use Illuminate\Database\Eloquent\Model;  class Comment extends Model{  protected $fillable = ['nickname', 'content','ip'];  } |

控制器 app/Http/Api/Controllers：

|  |
| --- |
| <?php  namespace App\Http\Controllers\Api;  use Illuminate\Http\Request;  use App\Http\Requests;  use App\Http\Controllers\Controller;  use App\Models\Api\Comment;  class CommentController extends Controller{  public function index(Request $request){  sleep(1);  $comments=Comment::orderBy('id','DESC')->paginate(5);  // dd($comments->toArray()['data']);  // $comments->setPath('http://vue-study.org/?#!/');  return $comments;  }  public function store(Request $request){  //自己完善验证  $data=$request->all();  try{  Comment::create([  'nickname'=>$data['nickname'],  'content'=>$data['content'],  'ip'=>$request->ip()  ]);  return response()->json(['error'=>0,'msg'=>'留言成功']);  }catch(\Exception $e){  return reponse()->json(['error'=>1,'msg'=>'系统错误']);  }  }  } |

路由 routes/api.php：

|  |
| --- |
| Route::group(['namespace'=>'Api'],function(){  //留言  Route::post('/comment','CommentController@store');  //获取留言列表  Route::get('/items','CommentController@index');  }); |

都很简单，不细说了。最后 apache 配置下.htaccess 来实现跨域请求:

|  |
| --- |
| <IfModule mod\_rewrite.c>  <IfModule mod\_negotiation.c>  Options -MultiViews  </IfModule>  RewriteEngine On  # Redirect Trailing Slashes If Not A Folder...  RewriteCond %{REQUEST\_FILENAME} !-d  RewriteRule ^(.\*)/$ /$1 [L,R=301]  # Handle Front Controller...  RewriteCond %{REQUEST\_FILENAME} !-d  RewriteCond %{REQUEST\_FILENAME} !-f  RewriteRule ^ index.php [L]  # Handle Authorization Header  RewriteCond %{HTTP:Authorization} .  RewriteRule .\* - [E=HTTP\_AUTHORIZATION:%{HTTP:Authorization}]  ##跨域请求配置  Header set Access-Control-Allow-Origin "\*"  Header set Access-Control-Allow-Headers "Origin, X-Requested-With, Content-Type, Access-Control-Allow-Origin, Authorization, Accept , Range-Unit , Range"  Header set Access-Control-Expose-Headers "Range-Unit ,Content-Range"  Header set Access-Control-Allow-Methods "PUT, GET, POST, DELETE, HEAD"  Header set Access-Control-Max-Age 60  </IfModule> |

后端完成。也没啥，就是处理下留言的提交和留言的拉取。

3. 留言板首页

首先在 app/components 创建：Home.vue 文件。看代码之前要看下 vue-router 的文档。

代码如下：

|  |
| --- |
| <template>  <header>  <h3>留言板<small>vuejs案例</small></h3>  </header>  <div class="container">  <div class="row">  <div class="col-lg-12">  <div class="col-lg-8 col-lg-offset-2">  <a class="btn btn-sm btn-primary pull-right" v-link="{path:'/comment'}">我要留言</a>  </div>  <div class="col-lg-8 col-lg-offset-2" v-show="items.length==0">  <button class="btn btn-lg btn-warning"><span class="glyphicon glyphicon-refresh glyphicon-refresh-animate"></span> Loading...</button>  </div>  <div class="col-lg-8 col-lg-offset-2" v-show="items.length>0">  <ul class="list-group" >  <li class="list-group-item" v-for="item in items"><span class="badge">{{item.created\_at}}</span>{{item.nickname}} 说：{{item.content}}</li>  </ul>  </div>  <div class="col-lg-8 col-lg-offset-2">  <Page :current-page="current\_page" :last-page="last\_page" :total="total"></Page>  </div>  </div>  </div>  </div>  </template>  <script>  import {config} from '../config.js';  import Page from './partials/Page.vue';  export default{  name:'Home',  methods:{  },  data:function(){  return {  items:[],  current\_page:1,  last\_page:1,  total:1  };  },  components:{  Page  },  route:{  activate: function (transition) {  // console.log('hook-example activated!')  document.title='留言板-vuejs';  // console.log(this.$route.params.page);  transition.next();  },  data:function(transition){  this.current\_page=this.$route.params.page?parseInt(this.$route.params.page):1;  this.$http.get(config.api+'/items?page='+this.current\_page).then((response)=>{  transition.next({  items:response.data.data,  current\_page:response.data.current\_page,  last\_page:response.data.last\_page,  total:response.data.total  });  },(response)=>{  });  }  },  events:{  loading:function(page){  this.items=[];  }  }  }  </script>  <style lang="sass">  .messages{  .title{  color:blue;  }  }  .list-group{  margin-top:5px;  }  .glyphicon-refresh-animate {  -animation: spin .7s infinite linear;  -webkit-animation: spin2 .7s infinite linear;  }  @-webkit-keyframes spin2 {  from { -webkit-transform: rotate(0deg);}  to { -webkit-transform: rotate(360deg);}  }  @keyframes spin {  from { transform: scale(1) rotate(0deg);}  to { transform: scale(1) rotate(360deg);}  }  </style> |

配置 config.js 如下

|  |
| --- |
| var config = {  api:'http://vue-study-api.org/api'  }  export {config}; |

这里使用了子组件 Page, app/components/partials/Page.vue

|  |
| --- |
| <template>  <nav v-show="lastPage>1">  <ul class="pagination">  <li><a v-link="{name:'home-comments',params:{page:1}}" @click="notify(1)">«</a></li>  <li v-for="n in lastPage" :class="{'active':currentPage==(n+1)}" @click="notify(n+1)"><a v-link="{name:'home-comments',params:{page:(n+1)}}">{{n+1}}</a></li>  <li><a v-link="{name:'home-comments',params:{page:lastPage}}" @click="notify(lastPage)">»</a></li>  </ul>  </nav>  </template>  <script>  export default{  name:'Page',  props:{  currentPage:Number,  lastPage:Number,  total:Number  },  methods:{  //每次点击 显示loading  notify(page){  this.$dispatch('loading', page);  }  }  }  </script>  <style lang="sass">  </style> |

http 请求使用了官方插件：vue-resource，很简单。

4. 留言页面

在 app/components 创建：Comment.vue 文件:

|  |
| --- |
| <template>  <header>  <h3>我要留言 <small><a v-link="{path:'/'}" class="btn btn-xs btn-default">返回</a></small></h3>  </header>  <div class="container">  <div class="col-lg-12">  <message :msg="msg" :type="msgType"></message>  <form role="form" v-on:submit="comment">  <div class="form-group">  <label for="nickname">昵称</label>  <input type="text" class="form-control" id="nickname" placeholder="Enter your nickname" v-model="nickname">  </div>  <div class="form-group">  <label for="content">留言</label>  <textarea class="form-control" id="content" placeholder="Content" rows="6" v-model="content"></textarea>  </div>  <button type="submit" class="btn btn-default">提交</button>  </form>  {{nickname}}<br>  {{content}}  </div>  </div>  </template>  <script>  import {config} from '../config.js';  import Message from './partials/Message.vue';  export default {  name:'Comment',  data:function(){  return {  nickname:'',  content:'',  msg:'',  msgType:''  }  },  components:{  Message  },  methods:{  comment(event){  if(this.nickname==''){  this.msgType='alert-danger';  this.msg='昵称不能为空';  event.preventDefault();  return false;  }  if(this.content==''){  this.msgType='alert-danger';  this.msg="请输入留言内容";  event.preventDefault();  return false;  }  this.$http.post(config.api+'/comment',{nickname:this.nickname,content:this.content}).then((response)=>{  //sucess  // console.log(response.data);  if(response.data.error==0){  this.msg=response.data.msg;  this.msgType="alert-success";  var route=this.$router;  setTimeout(function(){  route.go({name:'home'});  },1500);  }else{  this.msgType="alert-warning";  this.msg=response.data.msg;  }  },(response)=>{  //error  console.log(response);  });  event.preventDefault();  }  },  route:{  activate: function (transition) {  // console.log('hook-example activated!')  document.title='我要留言';  transition.next()  },  data:function(){  // alert(1);  }  }  }  </script>  <style lang="sass">  </style> |

使用了一个简单的子组件 Message, 其他都很简单。

5. 总结

基本上是贴代码，这也算是一个简单的前后分离小项目吧，感兴趣的可以看下前端代码，至少以后用 vuejs 写项目的时候这些配置文件可以直接拿来使用，减少部署时间。