

RESTful API 设计指南

以及 Egg 为单页面应用 Vue Angular 提供 api 接口实现前后端分离

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

一、 RESTful API 设计指南.....	1
二、 路由分组.....	2
三、 Egg 中通过 egg-cors 配置服务器端允许跨域.....	2

一、 RESTful API 设计指南

网络应用程序，分为前端和后端两个部分。当前的发展趋势，就是前端设备层出不穷（手机、平板、桌面电脑、其他专用设备.....）。

因此，必须有一种统一的机制，方便不同的前端设备与后端进行通信。这导致 API 构架的流行，甚至出现"API First"的设计思想。RESTful API 是目前比较成熟的一套互联网应用程序 API 设计理论。RESTful API 在企业中用的非常多。

一个好的 RESTful API 我们从以下几个方面考虑。

1. 协议：建议使用更安全的 https 协议
2. 域名：尽量部署在专属域名下面，比如 <https://a.itying.com> <https://api.itying.com>
3. 应该将 api 的版本号放入 URI 中：
 - (1) 比如：<https://a.itying.com/api1/newslst> <https://a.itying.com/api2/newslst>
 - (2) 比如：<https://a1.itying.com> <https://a2.itying.com>
4. 路径：在 RESTful 架构中，每个网址代表一种资源（resource），配置路由的时候尽量见名知意。
5. http 请求数据的方式：（7 个 HTTP 方法：GET/POST/PUT/DELETE/PATCH/HEAD/OPTIONS）
 - GET（SELECT）：从服务器取出资源（一项或多项）。
 - POST（CREATE）：在服务器新建一个资源。
 - PUT（UPDATE）：在服务器更新资源（客户端提供改变后的完整资源）。

DELETE (DELETE)：从服务器删除资源。

还有三个不常用的 HTTP 请求方式。

HEAD：获取资源的元数据。

OPTIONS：获取信息，关于资源的哪些属性是客户端可以改变的。

PATCH (UPDATE)：在服务器更新资源（客户端提供改变的属性）。

6、过滤方式、请求数据方式、返回数据方式、安全问题。

二、路由分组

新建 `routes/admin.js` `routes/api.js` 然后 `router.js` 中引入分组后的路由

代码如下：

```
module.exports = app => {  
  require('./routes/admin')(app);  
  require('./routes/api')(app);  
};
```

三、Egg 中通过 egg-cors 配置服务器端允许跨域

1、安装插件

```
npm i egg-cors --save
```

2、配置插件

```
// {app_root}/config/plugin.js  
exports.cors = {  
  enable: true,  
  package: 'egg-cors',  
};
```

3、配置安全域名：

```
// {app_root}/config/config.default.js  
exports.security = {  
  domainWhiteList: [ 'http://localhost:4200' ],  
};
```

项目中建议配置方式：

```
config.security = {
  csrf: {
    ignore: ctx => {
      if (ctx.request.url == `/${config.adminPath}/product/doUpload`) {
        return true;
      } else if (ctx.request.url.indexOf('/api') !== -1) {
        return true;
      } else {
        return false;
      }
    },
  },
  domainWhiteList: ['http://localhost:8081'],
}
```

4、配置允许跨域

基于：<https://www.npmjs.com/package/koa2-cors>

```
// {app_root}/config/config.default.js
config.cors = {
  origin: '*',
  allowMethods: 'GET,PUT,POST,DELETE'
};
```