- 解决跨域问题

```js
// webpack解决前端跨域问题
module.exports = {
  devServer:{
      proxy:{
          '/getUser':{
              target:'http://localhost:3000'
          }
      }
  }
}
```

- 请求拦截器 / 响应拦截器

```js
import axios from 'axios';
let baseUrl = process.env.NODE_ENV !== 'production'?'http://localhost:8080':'/';
class AjaxRequest{
  constructor(){
      this.baseURL = baseUrl; // 配置默认路径
      this.requestQueue = {}; // 缓存当前请求队列用来操作loading的显示或者隐藏
  }
  configInterceptors(instance,url){
      // 配置请求拦截
      instance.interceptors.request.use( (config) => {
          // loading效果  开始请求前显示loading
          if(Object.keys(this.requestQueue).length === 0){
              console.log('显示loading');
          }
          this.requestQueue[url] = config.url;
          return config;
      });
      // 配置响应拦截
      instance.interceptors.response.use(res=>{
          // 取消loading效果
          delete this.requestQueue[url];
          if(Object.keys(this.requestQueue).length === 0){
              console.log('取消loading');
          };
          if(res.status !== 200){
              return Promise.reject(res);
          }else{
              return res.data;
          }
      })

  }
  request(config){
      // 创建一个axios实例
      let instance = axios.create();
      // axios(config);
      config = {baseURL:this.baseURL,...config};
      this.configInterceptors(instance,config.url);
      return instance(config);
  }
}
export default new AjaxRequest();
```