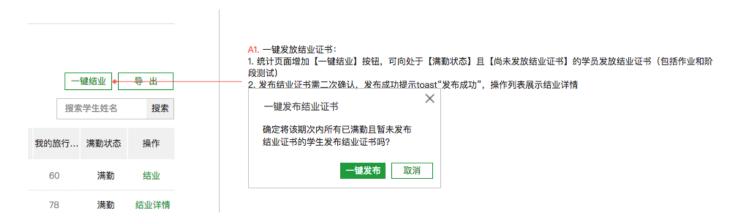
任务业务梳理

一、需求背景





这次遇到两个需求,一键结业和复制期次,都是长任务,需要轮询。

咋一看跟导入业务很像,区别在于,这个不需要对错误信息的特殊处理,也不需要上传文件,在接口规范方面也有一些区别。

二、为什么要封装

首先由于它跟导入业务有一些区别,所以不封装在一起。

有可能是短任务,一个接口,也可能是长任务,两个接口,两种情况封装在一起。

对发送请求的过程进行封装,如果不封装的话遇到类似业务还需要单独建cache很麻烦,并且可以制定此类接口规范,大家不用再每次遇到类似需求 去讨论接口。

三、资源

四、用法

用法参考demo k12-module-component/tutorials/common/task/demo.html

短任务,不需要轮询的时候配置hasPolling为false,然后不写pollingOpts,当然写了也不会用。

五、未来的扩展

暂时不支持带进度的

六、接口约定

启动接口

Method

没有特别要求,前端可以通过data.taskOpts.method配置

Content-Type

没有特别要求,不过前端目前没有加header的配置,会根据nej和requestConfig自动处理。需要加的话请前端同学自己加然后找我review。

传参

可自定义。

返回值



第一种情况是,<mark>前端并不需要用到返回的数据</mark>,如一键结业,那么后端返回值的规范是:第一次返回false,若为false,前端会继续轮询,直到返回true为止。

名称	类型	描述	默认值	生成规则
code	Number	描述	0	生成规则
message	String	描述	默认值	生成规则
result	Number	班次id	默认值	生成规则

第二种情况是,<mark>前端需要用到返回的数据</mark>,如期次复制,前端需要拿到复制后期次的id,那么后端返回值的规范是:第一<mark>次返回false或者</mark>-1,若为false或者-1,前端会继续轮询,直到返回true或不为-1为止。

```
judgeSuccess:function(res) {
    if(typeof res !== 'object'){
        if(typeof res === 'boolean'){
            return res:
        }else if(typeof res === 'number'){
            return res !== _1;
    }else{
            // 待补充
    }
},
judgeUnfinished:function (res) {
    if(typeof res !== 'object'){
        if(typeof res === 'boolean'){
            return !res;
        }else if(typeof res === 'number'){
            return res === _1;
        }else {
            // 待补充
        }
}else {
            // 待补充
    }
}else {
            // 待补充
    }
},
```

这两种种情况,都只需要一个接口。

需要两个接口的规范,以及需要复杂数据(如一个对象)的规范尚未制定,待后续填坑。

轮询接口

由于目前遇到的需求实际上启动和轮询都是同一个接口,所以暂时跟上面是一样的。

待补充

接口可参考:

一键结业: https://nei.netease.com/interface/detail/res/?pid=21961&id=89737

期次复制: https://nei.netease.com/interface/detail/?pid=21961&id=52616