

任务业务梳理

一、需求背景

一键结业

导出

搜索学生姓名

搜索

我的旅行...

满勤状态

操作

60

满勤

结业

78

满勤

结业详情

A1. 一键发放结业证书:
1. 统计页面增加【一键结业】按钮,可向处于【满勤状态】且【尚未发放结业证书】的学员发放结业证书(包括作业和阶段测试)
2. 发布结业证书需二次确认,发布成功提示toast“发布成功”,操作列表展示结业详情

一键发布结业证书

确定将该期次内所有已满勤且暂未发布结业证书的学生发布结业证书吗?

一键发布

取消

A1. 点击复制,弹出弹窗如下

复制期次到

本课程

新建课程

指定课程

请输入课程ID

确定

取消

A2. 选择“指定课程”,则输入框被激活,鼠标聚焦
此ID不属于本机构

A3. 点击“确定”后,
①若是“本课程”,则在当前课程下生成复制期次,页面跳转至期次介绍页;
②若是“新建课程”,则复制期次所在原课程,生成一门新课程,并在新课程下复制此期次,页面跳转至期次介绍页;
③若是“指定课程”,则校验ID是否属于本机构,且格式是否正确,校验通过在指定课程下复制期次,页面跳转至期次介绍页。

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

路径: kl2-module-component/src/common/task

四、用法

用法参考demo kl2-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。

```
c = new UI({
  data: {
    state: 'primary',
    hasPolling: true,
    taskOpts: {
      url: '/testupload/:termId',
      param: {
        termId: 123
      },
      data: {
        str: 'lzyzyzy'
      }
    },
    pollingOpts: {
      url: '/testpolling/:key',
      param: {
        key: 123
      },
      data: {
        str: 'pollllll'
      }
    }
  }
}).$inject(document.body)
```

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

返回结果: 哈希

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	Boolean	第一次返回false, 若为false, 前则会继续轮询, 直到返...	true	生成规则

第一种情况是，前端并不需要用到返回的数据，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。

返回结果: 哈希

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

添加

从数据模型导入

从JSON导入

从在线接口导入

从JavaBean导入

第二种情况是，前端需要用到返回的数据，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：第一次返回false或者-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
        }
    }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
    }
},
},
```

这两种情况，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>