

异步导出业务

交互描述

A1. 新增导出功能（包括作业和阶段测试）

1. 异步导出，导出当前期次内的所有数据
2. 命名为“K12作业统计-期次名称”或“K12阶段测试统计-期次名称”
3. 点击导出，导出置灰状态，hover导出展示导出文件名称，进度条，导出百分比
4. 导出成功，系统浏览器获取到url链接，自动下载

若异常情况导出失败提示toast“导出失败”；若导出过程中，离开页面或关闭页面，导出停止，用户下次点击导出需重新获取数据进行导出



前端涉及资源

component-button->button_async_export，使用方法可以参考demo，设计稿可以参考arch。

新建一个cache，包含\$export方法

```
pro.$export = function (data) {  
  this.__super(  
    'member-student-export', data, data, 'member-student-export-progress'  
  );  
};
```

config文件：

```
{  
  'member-student-export': {  
    method: 'GET',  
    url: '/j/org/{termId}/student/export',  
    rest: true,  
    isPolling: true  
  },  
  'member-student-export-progress': {  
    method: 'GET',  
    url: '/j/org/{termId}/student/export/process',  
    rest: true,  
    isPolling: true  
  }  
}
```

实际案例

机构后台学生管理导出功能

<http://teach.geek.163.com/terms/400000003117046#/studentEnroll>

接口设计

涉及两个接口：获取url（亦开启任务）接口，获取进度接口。

关于接口返回值，由于前端底层实现的问题（参见尾部附录），建议接口遵循以下原则：

1. 获取url（亦开启任务）接口：在result里面包一个对象

result数据模型：<https://nei.netease.com/datatype/detail/?pid=21559&id=96162>

```
1 {
2   "code": 0,
3   "message": "0J1WtIQNY0",
4   "result": {
5     "url": "http://ain.xbsxlflssojjw.tnntro",
6     "filename": "CW223LT7Z9"
7   }
8 }
```

还没获取url的时候，url为null或空串都行。

2. 获取进度接口：

result数据模型：<https://nei.netease.com/datatype/detail/?pid=21961&id=80418>

```
1 {
2   "code": 0,
3   "message": "jraFuMGc5G",
4   "result": {
5     "retCode": 21405,
6     "loaded": 25250,
7     "total": 83761,
8     "message": "pFEuGZbZBL",
9     "data": "n06xWnmRlT",
10    "extra": {
11      "filename": "P5dMZxCqAt"
12    }
13  }
14 }
```

目前还没有接口实现filename，前端把文件名写死的。这样如果要改文件名，前后端都需要改，尤其是当有条件过滤时，文件名可能因为条件的不同而发生变化，这样前后端都需要写一套逻辑。所以建议之后的接口加上filename。最好的做法是在「获取url接口」里面加，这样有些读取数据库的操作不需要反复进行（比如读取期次名称），不过前端需要去改一下底层代码；也可以在「获取进度接口」里面加，坏处就是有可能需要反复读取数据库，因为这是一个轮训的接口。



除此之外，接口实现细节还需要注意一下。

感觉盼盼的设计比较合理：在任务开启的时候就把任务数存下来，下次调获取进度时可以立刻拿到；total返回的不是n，而是n+1；文件为空也会返回url。这样的问题就是total并不代表真实的任务数量，比如导出7个人，实际上给的total为8，但是也可以理解，因为除了导出人数之外，还有额外的任务。

我和梁栋的约定是：total返回n；第一次调获取进度，由于他说拿不到任务数，所以造一个假的：total:1,loaded:0，这样前端可以展示0%，符合交互；如果任务为空，返回：total1,loaded:2，其实只要loaded/total>=100%就可以，原因可以参考附录的流程图。

第二种方案会有一个问题，当正常情况下拿到total为n且loaded为n时，实际上这个时候后端不一定就生成了url，但是前端底层会判断进度结束，这时候前端展示就是正常状态（不是导出中）了，然后去调用获得url的接口，此时不一定能拿到url，于是又去调用progress，拿到的同样是total为n且loaded为n，然后又去调获取url的接口，才拿到url开始下载。如此，用户可能看到导出已经结束了，但是还要等一两秒才能开始下载。

所以还是建议用第一种方案。

前端注意事项

- 1. 接口config文件中，两个接口都需要添加isPolling或者hideLoading为true，用来隐藏loading状态。
- 2. \$export的底层实现是：失败或成功的时候返回total和loaded都为100,并不会区分失败与否。

附录

\$export底层实现

