## 公共导航栏

未登录显示登录、注册按钮，已登陆显示你是谁及退出登录按钮。

以及到我的申请、我的审批、我的流程三页面的导航。

## creatProcess.html

创建新流程页面。

Php需要向JS提供所有审核人的数组。

首先输入流程名字，所有参与审核的角色及所有判断变量。由于判断变量只有int型，相当于只需要输入判断变量的名字。

页面有一块类似画布的操作区，一条选择添加、删除结点工具，添加、删除边工具的工具栏。

初始时画布有起点和终点。

添加结点的操作：

点击工具栏添加结点按钮，选择审核人，拖动新结点（用一个HTML div实现）到画布以添加结点。不能选择已被选过的审核人，应动态改变select的选项。

点击工具栏删除结点按钮，再点击一个结点则删除它及所有与它相连的边。不能删除起点或终点。

点击工具栏添加边按钮，再点击两个不同的结点，创建一条边。只有两个不同且没有边相连的结点才可以添加边。

点击工具栏添加条件判断边按钮，选择条件判断变量，输入变量大小范围，再点击两个不同的结点，创建一条带条件判断边。只有两个不同且没有边相连的结点才可以添加边。

点击工具栏删除边按钮，再点击两个不同的结点，删除一条边。只有两个不同且有边相连的结点才可以删除边。

完成编辑后该页面生成一个表示图结构的JSON对象str。Str的对象结构为：

{

title:“”， //流程的标题

vars:[““，““,“”], //流程的所有变量

nodes:[“”,””,””], //nodes数组，元素是结点的审核人的名字

edges:[ //edges数组，元素是一个个对象，表示边的信息

{

from:””, //起点的审核人的名字

to:””, //终点的审核人的名字

varName:””, //条件判断变量的名字，如果没有则为空

varLow:””, //条件判断变量的左边值，如果没有则为空

varHi:”” //条件判断变量的右边值，如果没有则为空

},

{},

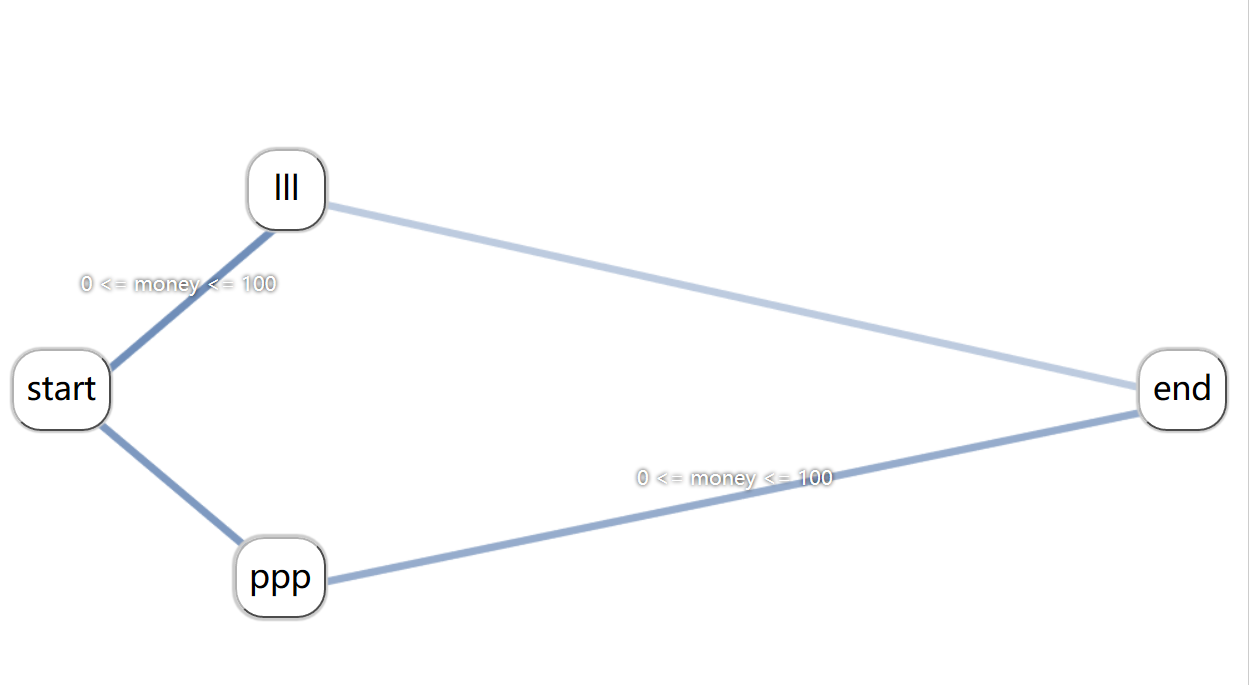
{}

]

}

Str示例：

{"title":"jsjfjds","nodes":["start","end","lll","ppp"],"edges":[{"from":"lll","to":"end","varName":"","varLow":"","varHi":""},{"from":"ppp","to":"end","varName":"money","varLow":"0","varHi":"100"},{"from":"start","to":"lll","varName":"money","varLow":"0","varHi":"100"},{"from":"start","to":"ppp","varName":"","varLow":"","varHi":""}]}



## applyInfo.html

GET参数:applyid申请的id

显示该申请的申请者、所有变量的值、其详细内容，

并在该页面选择通过审核或拒绝通过。

## showApply.html

GET参数：applyid 申请的id

以画布的形式画出申请的状态

Php向JS提供一个JSON串jstr表示图的结构以及流程审核情况。

Jstr结构如下：

{

nodes:{“”,””,””}, //图中所以结点名字

edges:[

{

from:””,

to:””,

varName:””,

varLow:””,

varHi:””,

status:”” //status表示边审核的状态，0：审核通过，1：审核不通过

2：未审核

},

{},

{},

]

}

## showProcess.html

GET参数：processid显示的流程的id

用画布的方式画出流程。本质上是jstr参数更完整的showApply.html。

有一个按钮可以进入管理流程角色页面

## manage.html

GET参数： processid 要管理的流程的id

一个流程创建完后已经决定了需要哪些角色作为审核人，该页面则为这些角色分配用户，让这些用户作为不同角色参与审核。

每个角色只有一个，且一人只能担任一个角色。

php向前端提供一个json对象jstr.该对象结构为：

{

users:[ //users数组，保存该流程所有可担任审核人的

{ 用户的id和名字，包括已经担任该流程

id:””, 角色的人。

name:””,

},

{

},

{

}

],

positions:[ //positions数组，保存该流程所有角色及担任该角

{ 色的用户的id。如果该角色没人担任则

pos:””, userid=“”

userid:””

},

{

},

{

}

]

}

}

提交后，前端会向php提供一个Json对象jstr，其结构为

{

positions:[ //positions数组，保存该流程所有角色及担

{ 任角色的用户id。若无人担任则usreid=“”

pos:””,

userid:””

},

{

},

]

}

## login.html

两个input，输入邮箱密码。

有登录及注册按钮。

## register.html

三个input，输入邮箱密码昵称。

不需要验证邮箱、昵称、密码正确性。

## myApply.html

显示我所有提出的申请及结果，点击一个申请可以跳到以图形显示申请状态的showApply页面。

有创建一个申请的按钮，链接到”createApply.html?processid=”+流程id。

## createApply.html

GET参数：processid 要创建的申请的流程id

列出该流程所有参数及其input。

最后还有一个文本框保存额外的说明信息。

## myCheck.html

显示需要由我审批及已经由我审批过的申请，点击一个审批跳到详细展示一个申请的applyInfo页面

在allapplyedge表里，res字段的不同值代表的含义如下

|  |  |
| --- | --- |
| 0 | 待审核 |
| 1 | 通过 |
| 2 | 不通过 |

## myProcess.html

显示我创建的所有流程，我在的流程。有创建流程及加入流程的按钮。

## index.html

用户登录后进入的首页，未登录则需要转到登录页面。

显示一些无关紧要的信息，防止首页太空。

## manage.html

GET参数：

流程ID：pid

只有pid流程创建者可以进入。显示流程里所有人及其角色，并可以调整角色。

除普通用户外每个角色只有一个人。

## join.html

显示未加入的流程，并可选择加入。

processmember表中，固定的数值对应角色有

0 创建者

1 普通用户