服务设计文档

# 1 初步划分微服务

在将一个系统划分成多个微服务后，服务之间仅通过API交互，屏蔽内部实现细节，为使服务达到更好的扩展性和稳定性，我们应遵循以下划分原则：1. 按功能模块划分，尽量做到一个功能模块一个微服务；2. 微服务之间减少互相调用，做到低耦合高内聚。

此次我们需要划分的CyberChair系统大致可以分为作者提交文章、主席发起会议、审核文章、管理员审核会议的功能，我们依据以上划分原则并降低服务间的代码重复，将CyberChair拆分成以下五个服务： User、Author、Chair、PCmember、Meeting。下面对这五个服务的具体设计进行介绍。

# 2 User服务

User服务提供用户角色所需要的基础功能，比较综合，具体功能如下：

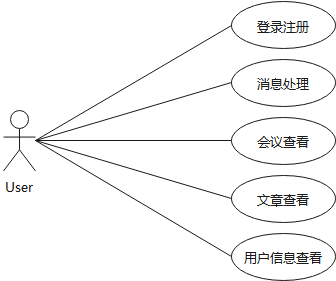


图1 User用例图

主要功能是用户可以注册、登录，查看目前可用的会议信息，查看自己的投稿信息，处理相关消息等等。其中涉及到的实体类有PostMessage、User，提供的API接口信息具体如下：

|  |  |  |
| --- | --- | --- |
| URL | RequestMethod | Function |
| /postmessage | POST | 新增PostMessage |
| /postmessage/article | GET | 查找文章下的postmessage |
| /postmessage/id | GET | 根据id查找PostMessage |
| /user/author | GET | 通过author信息找用户 |
| /user/id | GET | 根据id查找用户 |
| /user/username | GET | 根据username查找用户 |
| /user/alreadyDealedNotifications | GET | 根据username获取用户已经处理的通知 |
| /user/invitationRepo | POST | 回复邀请 |
| /user/undealedNotifications | GET | 根据username来找用户下的未处理的通知 |
| /user/undealedNotificationsNum | GET | 根据username获取用户未处理的通知的数量 |
| /user/authorMeeting | GET | 获取author下的meeting |
| /user/availableMeeting | GET | 返回可以投稿的meeting  chair不可以投稿自己的会议 |
| /user/chairMeeting | GET | 获取chair下的meeting |
| /check | GET | 检查登录状态 |
| /login | POST | 登录 |
| /register | POST | 注册 |
| /user/userinfo | GET | 根据username返回用户 |
| /util/users | GET | 根据fullname查找用户 |
| /welcome | GET | welcome |

表1 User提供的API

需调用其他服务的API如下：

|  |  |  |
| --- | --- | --- |
| MicroservicesName | URL | RequestMethod |
| Author | /article/findArticleById | GET |
| Author | /article/findArticleByContributorName | GET |
| Meeting | /meeting/pcMemberRelation/getByPcmemberIdAndStatus | GET |
| Meeting | /meeting/getById | GET |
| Meeting | /meeting/getByName | GET |
| Meeting | /meeting/getByChairName | GET |
| Meeting | /meeting/getByStatusAndChairNameNot | GET |
| Meeting | /meeting/pcMemberRelation/getByPcmemberIdAndMeetingId | GET |
| Meeting | /meeting/pcMemberRelation/save | POST |
| Meeting | /meeting/pcMemberRelation/getByPcmemberIdAndStatusNot | GET |

表2 User调用的API

# 3 Author服务

Author服务提供用户角色为作者时，文章投稿等相关功能，具体功能如下：

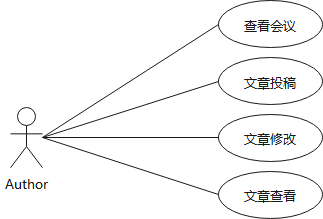


图2 Author用例图

系统流程是作者在会议列表选择一个会议投递文章，编辑投稿信息，除去在User中提供的基本功能，在这个服务中主要提供文章投稿、文章查看、文章修改的功能，其中涉及到的实体类有Article，提供的API接口信息具体如下：

|  |  |  |
| --- | --- | --- |
| URL | RequestMethod | Function |
| /user/updateArticle | POST | 更新文章信息 |
| /user/reviews | GET | 查看文章 |
| /user/articleDetail | GET | 获取一篇文章的详细信息 |
| /user/articleSubmission | POST | 提交一个文章 |
| /meeting/submissionList | GET | 获取一个author在一个meeting上的所有提交的文章 |
| /author/authorMeeting | GET | 获取一个author投稿过的所有meeting |
| /article/findArticleById | GET | 根据文章ID获取文章 |
| /article/findArticalByIdNot | GET | 获取不等于此文章ID的文章 |
| /article/findArticleByContributorName | GET | 获取对应contributor的所有文章 |
| /article/findArticleByMeetingName | GET | 获取对应meetingName的所有文章 |
| /article/findArticleByMeetingNameAndStatus | GET | 获取对应meetingName和status的所有文章 |
| /article/findArticleByContributorNameAndMeetingName | GET | 获取对应contributor和meeting的所有文章 |

表3 Author提供的API

需调用其他服务的API如下：

|  |  |  |
| --- | --- | --- |
| MicroservicesName | URL | RequestMethod |
| User | /check | GET |
| Meeting | /meeting/getByName | GET |
| PCmember | /reviewRelation/findReviewRelationsByArticleId | GET |

表4 Author调用的API

# 4 Chair服务

Chair服务提供用户角色为Chair时，申请会议等相关功能，具体功能如下：

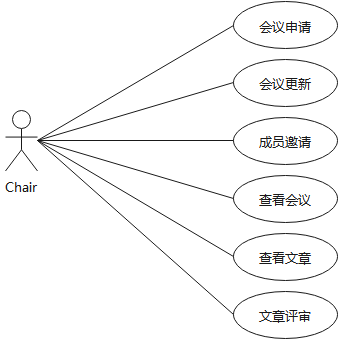


图3 Chair用例图

主要流程是用户在首页发起一个会议，编辑会议相关信息，发布会议之后可邀请成员加入，同时他可以审核参与者发布的文章并评审。其中查看会议、查看文章是所有用户都可以进行的操作，在后面的User中会统一介绍。在Chair服务中涉及到的实体类有Meeting、PCMemberRelation，提供的API接口信息具体如下：

|  |  |  |
| --- | --- | --- |
| URL | RequestMethod | Function |
| /meeting/beginSubmission | POST | 设置meeting状态为  submissionAvaliable |
| /meeting/finalPublish | POST | 设置meeting状态为reviewPublish |
| /meeting/pcmInvitation | POST | 设置pcMemberRelation状态为applyFailed |
| /meeting/publish | POST | 设置meeting状态为resultPublished |

表5 Chair提供的API

需调用其他服务的API如下：

|  |  |  |
| --- | --- | --- |
| MicroservicesName | URL | RequestMethod |
| Meeting | /meeting/save | POST |
| Meeting | /meeting/pcMemberRelation/save | POST |

表6 Chair调用的API

# 5 PCmember服务

PCmember服务提供用户角色为PCmember时，文章评审等相关功能，具体功能如下：

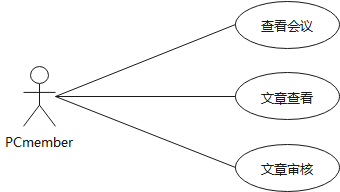


图4 PCmember用例图

主要功能是用户作为审稿人查阅文章、进行文章审核，可给予文章“接受”、“拒绝”的处理结果。除了User中的主要功能是文章审核，其中涉及到的实体类有Rebuttal、reviewRelation，提供的API接口信息具体如下：

提供给其他服务的API：

|  |  |  |
| --- | --- | --- |
| URL | RequestMethod | Function |
| /rebuttal/findByArticleId | GET | 根据ArticleId查rebuttal表 |
| /rebuttal/findById | GET | 根据id查rebuttal表 |
| /rebuttal/findByIdNot | GET | 根据id查rebuttal表中不是这个id的记录 |
| /rebuttal/save | GET | 将rebuttal对象存入rebuttal表 |
| /reviewRelation/findByArticleId | GET | 根据ArticleId查reviewRelation表 |
| /reviewRelation/findById | GET | 根据Id查reviewRelation表 |
| /reviewRelation/findByIdNot | GET | 根据Id查reviewRelation表中不是这个Id的对象 |
| /reviewRelation/findByMeetingId | GET | 根据MeetingId查reviewRelation表 |
| /reviewRelation/findByReviewerIdAndArticleId | GET | 根据ReviewerId和ArticleId查reviewRelation表 |
| /reviewRelation/findByReviewerIdAndMeetingId | GET | 根据ReviewerId和MeetingId查reviewRelation表 |
| /reviewRelation/findByReviewStatus | GET | 根据ReviewStatus  查reviewRelation表 |
| /reviewRelation/findByReviewStatusAndMeetingId | GET | 根据ReviewStatus和MeetingId查reviewRelation表 |
| /reviewRelation/findReviewRelationsByArticleId | GET | 根据ArticleId查reviewRelation表 |
| /reviewRelation/save | POST | 将reviewRelation对象存入表中 |

提供给前端的API：

|  |  |  |
| --- | --- | --- |
| URL | RequestMethod | Function |
| /meeting/alreadyReviewedInfo | GET | 查询已经review过的文章 |
| /meeting/beginReview | POST | 开始review前的一些配置 |
| /meeting/rebuttal | POST | 拒稿操作 |
| /meeting/reviewArticle | GET | 根据文章查看review状态 |
| /meeting/reviewArticles | GET | 根据会议名称查看review状态 |
| /meeting/reviewConfirm | POST | Review分为多次确认步骤 |
| /meeting/reviewer | POST | 对当前会议的文章进行review，并存储数据 |
| /meeting/updateReview | POST | 拒稿后再次review的操作 |
| /user/reviews | GET | 根据articleId对文章进行review，并存储数据 |

表7 PCmember提供的API

需调用其他服务的API如下：

|  |  |  |
| --- | --- | --- |
| Microservices  Name | URL | RequestMethod |
| User | /user/findByUsername | GET |
| User | /user/findById | GET |
| User | /user/findByEmail | GET |
| User | /user/findByFullnameAndEmail | GET |
| Meeting | /meeting/findByMeetingName | GET |
| Meeting | /meeting/findById | GET |
| Meeting | /meeting/save | POST |
| Author | /article/findById | GET |
| Author | /article/findByIdNot | GET |
| Author | /article/save | POST |
| Author | /article/findByMeetingNameAndStatus | GET |
| Meeting | /pcMemberRelation/findByMeetingIdAndStatus | GET |
| User | /check | GET |

表8 PCmember调用的API

# 6 Meeting服务

Meeting服务提供会议相关的功能，例如发布会议、获取会议文章信息、评审文章、邀请成员、消息处理的功能。另外这个部分还包含了admin相关功能，用于处理会议申请、用户注册申请等消息，由此，这个Meeting服务最主要的还是给其他服务提供一系列支持。其中涉及到的实体类有Meeting、PCMemberRelation，提供的API接口信息具体如下：

|  |  |  |
| --- | --- | --- |
| URL | RequestMethod | Function |
| /admin/alreadyApplication | GET | 获取已经处理的meeting申请 |
| /admin/queueingApplication | GET | 查找未处理的meeting申请 |
| /admin/ratify | POST | 给会议打分 |
| /meeting/alreadyReviewedInfo | GET | 获取审核的信息 |
| /meeting/beginReview | POST | 开始审核文章 |
| /meeting/beginSubmission | POST | 会议开始接受投稿 |
| /meeting/publish | POST | 将会议状态设置为publish |
| /meeting/reviewArticle | GET | 获得⽂章进⾏审核 |
| /meeting/reviewArticles | GET | 获取审核的⽂章列表 |
| /meeting/reviewer | POST | 审核文章 |
| /meeting/postList | GET | 获取当前文章对应的post消息 |
| /meeting/reviewPost | POST | 发送review⽂章的消息 |
| /meeting/getByChairName | GET | 根据chairName查找meeting列表 |
| /meeting/getById | GET | 根据meetingId查找meeting |
| /meeting/getByName | GET | 根据meeting名字查找meeting |
| /meeting/getByStatusAndChairNameNot | GET | 根据chairName和会议状态查找meeting列表 |
| /meeting/pcMemberRelation/getByMeetingId | GET | 通过meetingId来获取该meeting下的所有pcMemberRelation |
| /meeting/pcMemberRelation/getByMeetingIdAndStatus | GET | 通过meetingId和状态来筛选所有pcMemberRelation |
| /meeting/pcMemberRelation/getByPcmemberIdAndMeetingId | GET | 通过pcmemberId和MeetingId来筛选所有pcMemberRelation |
| /meeting/pcMemberRelation/getByPcmemberIdAndStatus | GET | 通过pcmemberId和状态来筛选所有pcMemberRelation |
| /meeting/pcMemberRelation/getByPcmemberIdAndStatusNot | GET | 通过pcmemberId和⾮状态来筛选所有pcMemberRelation |
| /meeting/pcMemberRelation/save | POST | 通过post向数据库中添加pcMemberRelation |
| /meeting/pcMemberRelation/save | PUT | 通过put来修改数据库中pcMemberRelation |
| /meeting/save | POST | 通过post⽅式保存meeting进⼊数据库 |
| /meeting/save | PUT | 通过put⽅式修改数据库中的meeting |
| /meeting/application | POST | 提交新增meeting的申请 |
| /meeting/invitationStatus | GET | 获取会议所有邀请的状态 |
| /meeting/meetingInfo | GET | 根据meetingName获取会议信息 |
| /meeting/pcmInvitation | POST | 邀请⽤户成为pcmember |
| /meeting/submissionList | GET | 根据会议名称和作者获取相应的投稿 |

表9 meeting提供的API

需调用其他服务的API如下：

|  |  |  |
| --- | --- | --- |
| MicroservicesName | URL | Requet  Method |
| Author | /article/findArticleById | GET |
| Author | /article/findArticleByIdNot | GET |
| Author | /article/findArticleByMeetingNameAndStatus | GET |
| Author | /article/save | POST |
| Author | /article/findArticleByContributorNameAndMeetingName | GET |
| User | /check | GET |
| User | /user/username | GET |
| User | /user/email | GET |
| User | /user/id | GET |
| User | /user/author | GET |
| User | /postmessage | GET |
| User | /postmessage/article | GET |
| User | /postmessage/id | GET |
| PCmember | /reviewRelation/findByReviewerIdAndMeetingId | GET |
| PCmember | /reviewRelation/findByReviewStatusAndMeetingId | GET |
| PCmember | /reviewRelation/findByReviewerIdAndArticleId | GET |
| PCmember | /reviewRelation/save | POST |
| PCmember | /reviewRelation/findReviewRelationsByArticleId | GET |
| PCmember | /reviewRelation/findByArticleId | GET |
| PCmember | /rebuttal/save | POST |
| PCmember | /rebuttal/findByArticleId | GET |
| PCmember | /rebuttal/findByIdNot | GET |

表10 meeting调用的API

# 7 各微服务协作

在kubernetes的框架下，以上五个微服务通过eureka注册，通过gateway进行服务间的调用，主要协作如下图：

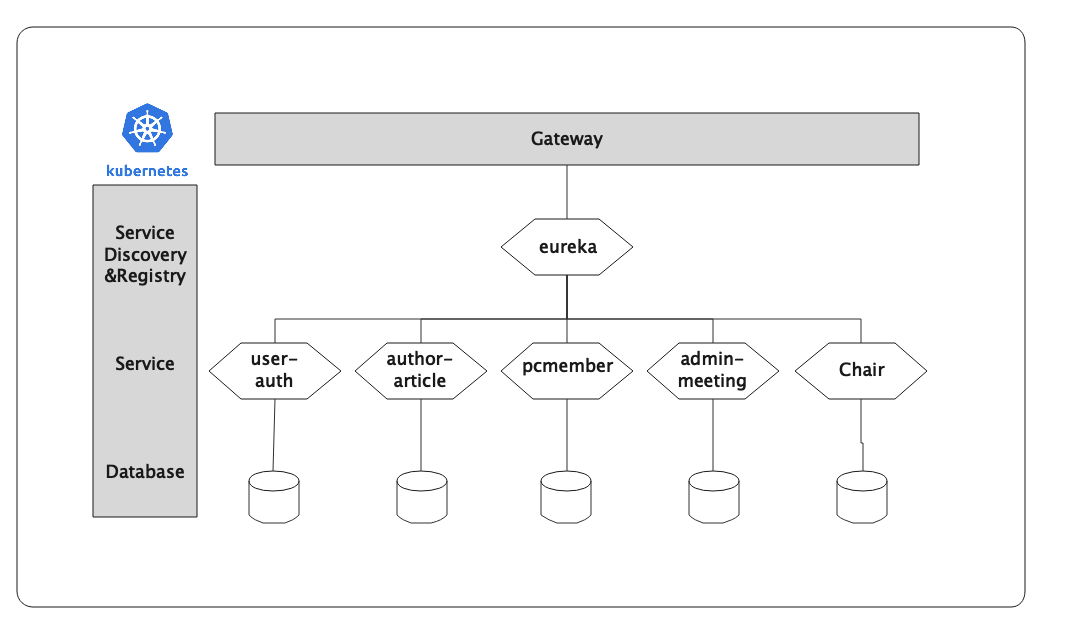


图5 各服务协作机制图