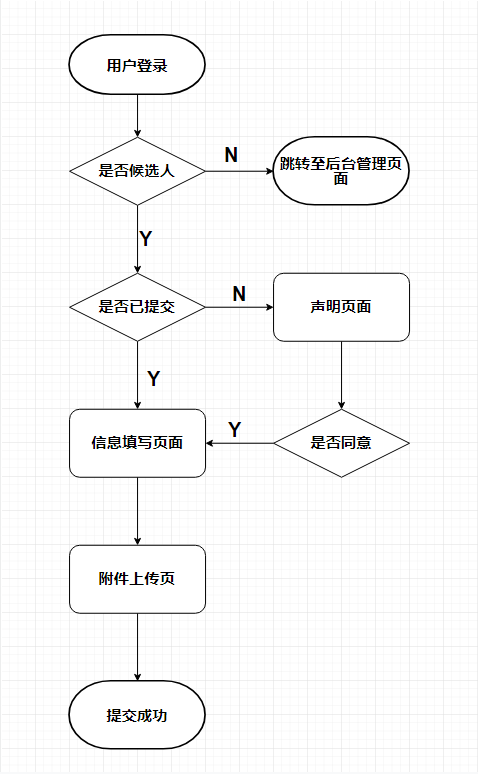
前台流程图



填写基础信息模块接口入口

@PostMapping("save")  
public ResultResponse save(@RequestBody @Valid BasicInfoNaDto basicInfoNaDto, BindingResult bindingResult) throws ParseException {  
 if (bindingResult.hasErrors()) {  
 return ResultResponse.*buildFailed*(Constants.*ILLEGAL\_ARGUMENT*, bindingResult.getFieldError().getDefaultMessage());  
 }  
 String dateOfBirthStr = basicInfoNaDto.getDateOfBirthStr();  
 SimpleDateFormat sdf = new SimpleDateFormat(Constants.*TIME\_FORMAT*);  
 if (StringUtils.*isNotBlank*(dateOfBirthStr)) {  
 basicInfoNaDto.setDateOfBirth(sdf.parse(dateOfBirthStr));  
 }  
 //判断出生日期是否大于当前日期  
 Date dateOfBirth = basicInfoNaDto.getDateOfBirth();  
 if (null != dateOfBirth && dateOfBirth.after(new Date())) {  
 return ResultResponse.*buildFailed*(Constants.*ILLEGAL\_ARGUMENT*, "The date of birth should be earlier than the current date !");  
 }  
 return basicInfoNaFeignService.save(basicInfoNaDto);  
}

Service实现

@Override  
@Transactional(rollbackFor = Exception.class)  
public ResultResponse save(@RequestBody BasicInfoNaDto dto) {  
 UserInfo userInfo = operatorContextHolder.getUserInfo();  
 String userId = userInfo.getId();  
 String applicantId = userInfo.getApplicantId();  
 if (StringUtils.*isBlank*(applicantId)) {  
 return ResultResponse.*buildFailed*(ResultCode.*NOT\_AN\_INTERVIEWER*);  
 }  
  
 HashMap map = new HashMap();  
 BasicInfoNaBean bean = new BasicInfoNaBean();  
 BeanUtils.*copyProperties*(dto, bean);  
 bean.setStatus(1);  
 bean.setDeleted(0);  
 bean.setApplicationId(applicantId);  
  
 //防止多条数据  
 basicInfoNaBeanMapper.deleteByApplicantId(applicantId,userId);  
  
 //如果有id则为修改 无则新增  
 int conut = 0;  
 if (StringUtils.*isNotBlank*(dto.getId())) {  
 bean.setModifier(userId);  
 bean.setModifiedTime(new Date());  
 conut = basicInfoNaBeanMapper.updateByPrimaryKeySelective(bean);  
 map.put("id",bean.getId());  
 } else {  
 bean.setId(UUIDUtils.*generateUUID*());  
 bean.setCreator(operatorContextHolder.getUserId());  
 bean.setCreateTime(new Date());  
 conut = basicInfoNaBeanMapper.insertSelective(bean);  
 map.put("id",bean.getId());  
  
 }  
 if (conut != 1) {  
 throw new InterviewServiceException(ResultCode.*INSERT\_FAILED*);  
 }  
  
 return ResultResponse.*buildSuccess*(map);  
}