SIM日志规范

文档版本：1.0

产品版本：${version}

1.内容目录

2.文档信息

2.1 文档目标

目标读者：开发人员。

目标：统一开发人员规范日志打印、异常码定义以及编写详细的异常码文档；让实施人员能够看懂日志，根据异常码查询异常信息可以初步解决问题。

2.2 修订历史

时间	版本	内容	修订人
2020-03-10	1.0	新建	周亮

2.3 相关术语

术语	说明

3. 日志说明

3.1 操作说明

打印日志时操作如下（ERROR级别日志必须如下操作）：

1. 在ExceptionConsts类中定义异常码，以及简要的异常描述信息。

2. 在com.bizenit.idm.sim.commons.exception目录下查找与第一步关联的自定义异常类，不存在则创建。

3. 维护【SIM异常码参考】文档，文档机构如下：

序号	异常名称	是否运行时异常	异常编号	异常说明	解决方案
1	XxxxException	否	E0101	xxxx	xxxx

- 异常码定义类

```
com.bizenit.idm.sim.commons.consts.ExceptionConsts
```

- 异常类存放目录

```
com.bizenit.idm.sim.commons.exception
```

- API接口异常码定义类

```
com.bizenit.idm.sim.commons.consts.RestCodeConsts
```

3.2 异常码规范

异常码结构：日志级别+模块+序号（示例：E0101）

日志级别(一位)：ERROR: E，WARN: W，INFO: I，DEBUG: D，TRACE: T

业务模块(两位)：用户：01，权限：02，系统设置：03 ......（后续梳理）

序号(两位)：两位数序号

3.3 日志格式

- 日志信息格式：异常码+异常描述+参数

```
LOG.error("E0101|GroupType not found|[userId:[{}],GroupTypeId:[{}]]",userId,GroupTypeId,
new Exception("GroupType not found"));
```

如下图：E0101|GroupType not found|[userId:[U123123123],GroupTypeId:[100025]]

- 抛异常消息格式：异常描述+参数

```
throw new QueryNotFoundException("Attribute named is not found in db|[userId:[%s],name:
[%s]]",userId,name);
```

3.4 日志规则

1. 尽量打印关键业务参数，如操作用户ID、业务ID等；
2. 不能打印密码、电话号、身份证号等私密信息；
3. 多个异常码可以对应一个异常，在抛出异常时尽量将异常信息描述清楚；
4. 接口输入参数需要用TRACE打印，输出参数根据实际情况。

4. 模块层级日志处理

1. Dao层、工具类层不允许将异常吃掉；
2. 业务处理层需要处理完所有的异常，不能将异常抛到用户页面。

4.1 Action层异常

注意：必须在Action层将所有异常处理完成，不能将异常再次抛出。

示例：com.bizenit.idm.sis.web.app.AppDataAction

```java
    public OpResult fileUploadJson(MultipartFile file) {
        String username = WebUtils.getUserPrincipal();
        Optional<User> userOptional = userService.getUserByUsername(username);
        if (userOptional.isEmpty()){
            return ApiResultUtil.opFail(RestCodeConsts.ATTR_ILLEGAL,USER_ID_ERROR_USER_NOT_EXIST);
        }
        if(file == null) {
            return ApiResultUtil.opFail(RestCodeConsts.PHOTO_IS_NULL);
        }
        BufferedImage bi;
        ByteArrayOutputStream baos = null;
        try {
            LOG.error(ExceptionConsts.PERSISTS_EXCEPTION, e);
            return ApiResultUtil.systemError(RestCodeConsts.DATABASE_ERROR);
        } catch (UserAttributeNotFoundException e) {
            LOG.error(UPLOAD_USER_PHOTO_ERROR, e);
            return ApiResultUtil.systemError(RestCodeConsts.DATABASE_ERROR);
        } catch (IOException e) {
            LOG.error(UPLOAD_USER_PHOTO_ERROR, e);
            return ApiResultUtil.systemError(RestCodeConsts.SYSTEM_ERROR,e.getMessage());
        }finally{
            if(baos != null) {
                try {
                    baos.close();
                } catch (IOException e) {
                    LOG.error(ExceptionConsts.IO_EXCEPTION, e);
                }
            }
        }
    }
```

4.2 Service层异常

注意：尽量在Service层将所有异常处理完成。

示例：com.bizenit.idm.sim.identity.ldap.LdapUserAdapterImpl

```java
349
350        @Override
351        public boolean addUser(User user)  {
352            try {
353                User u = buildUserForAdd(user);
354
355                LOG.debug("add user {}", u.getName());
356                List<com.unboundid.ldap.sdk.Attribute> ldapAttrs = ldapAttrTransform.userAttrToResAttr(u.getAttributes());
357
358                LDAPResult result = ldapProvider.getConnectionPool().add(u.getName(), ldapAttrs);
359                return result.getResultCode().equals(ResultCode.SUCCESS);
360            } catch (PersistsException e) {
361                LOG.error(ExceptionConsts.PERSISTS_EXCEPTION,e);
362            } catch (LDAPSearchException e) {
363                LOG.error(ExceptionConsts.LDAP_SEARCH_EXCEPTION,e);
364            } catch (ConnectionException e) {
365                LOG.error(ExceptionConsts.CONNECTION_EXCEPTION,e);
366            } catch (UserAttributeNotFoundException e) {
367                LOG.error(ExceptionConsts.USER_ATTRIBUTE_NOTFOUND_EXCEPTION,e);
368            } catch (UserTypeNotFoundException e) {
369                LOG.error(ExceptionConsts.USER_TYPE_NOTFOUND_EXCEPTION,e);
370            } catch (LDAPException e) {
371                LOG.error(ExceptionConsts.LDAP_EXCEPTION,e);
372            }
373            return false;
374        }
375
```

4.3 API业务异常

注意：必须在API Service所有异常处理完成，不能将异常再次抛出。

示例：com.bizenit.idm.sim.rest.api.v3.FlowApiService

```java
@ApiOperation(value = "查询认证流程往模板", httpMethod = "POST", response = OpResult.class, notes = "查询认证流程往模板")
public OpResult<List<Map<String,Object>>> getTemplate(@RequestBody FlowTemplateQueryParam flowTemplateQueryParam){
    String type = flowTemplateQueryParam.getType();
    List<Map<String, Object>> queryParams = flowTemplateQueryParam.getQueryParams();
    try{
        ParamCheckUtils.checkNotEmpty(TYPE,type);
        ParamCheckUtils.checkNotEmpty(QUERY_PARAMS,queryParams);
        List<Map<String,Object>> templates = new ArrayList<>();
        for(Map<String,Object> map : queryParams){
            // [start]
        }
        return ApiResultUtil.success(templates);
    }catch (IllegalParamException e){
        return ApiResultUtil.opFail(e.getRestCode(), e.getMessage());
    }catch (PersistsException e){
        logger.error(ExceptionConsts.PERSISTS_EXCEPTION, e);
        return ApiResultUtil.systemError(RestCodeConsts.DATABASE_ERROR,e.getMessage());
    }catch (Exception e){
        logger.error(SystemConsts.ERROR, e);
        return ApiResultUtil.systemError(RestCodeConsts.SYSTEM_ERROR,e.getMessage());
    }
}
```

4.4 Dao层异常

注意：该层不允许将异常捕获之后不抛出。

示例：com.bizenit.idm.sim.persists.appresource.impl.AppResourceDaoImpl

```java
@Override
public boolean merge(AppResource appResource) throws PersistsException {
    try {
        entityManager.merge(appResource);
    } catch (Exception e) {
        throw new PersistsException(e);
    }
    return true;
}
```

4.5 工具类

注意：该层不允许将异常捕获之后不抛出。

示例：com.bizenit.idm.sim.commons.utils.StringUtil

```
/**
public static String changeCharset(byte[] srcBytes, String srcCharset, String targetCharset) throws UnsupportedEncodingException {
    byte[] targetBytes = new String(srcBytes, srcCharset).getBytes(targetCharset);
    return new String(targetBytes);
}
```

5.通用基础异常

以下为常用的基础异常类；在底层抛出基础异常时，需要尽量将异常信息、参数信息描述清楚可读。

如果调用的方法抛出了运行时异常，那么在业务层必须处理。

5.1 查询异常

异常类：com.bizenit.idm.sim.commons.exception.QueryNotFoundException

使用示例：com.bizenit.idm.sim.persists.impl.AttributeDaoImpl

```
@Cacheable(value = "redisCache:default", key = "'sim:Attribute:name:'+#name")
public Attribute findAttributeByName(String name) throws PersistsException,QueryNotFoundException {
    Query query = entityManager.createNamedQuery("attribute.findByName");
    query.setParameter("name", name);
    List<Attribute> result = query.getResultList();
    if(!result.isEmpty()) {
        return result.get(0);
    }
    throw new QueryNotFoundException("Attribute named is not found in db|%s",name);
}
```

5.2 数据存储异常

异常类：com.bizenit.idm.sim.commons.exception.PersistsException

使用示例：com.bizenit.idm.sim.persists.impl.GroupTypeDaoImpl

```
        @CacheEvict(value = "redisCache:default", key = "'sim:GroupType:typeId:'+#type.id") })
public boolean updateGroupTypes(GroupType type) throws PersistsException {
    try {
        GroupType gt = entityManager.merge(type);
        return gt != null;
    } catch (Exception e) {
        throw new PersistsException(e,"update grouptype error , id:[%s]",type.getId());
    }
}
```

5.3 删除失败异常

异常类：com.bizenit.idm.sim.commons.exception.DeleteFailException

使用示例：com.bizenit.idm.sim.persists.impl.GroupTypeDaoImpl

```
public boolean deleteGroupTypes(GroupType type) throws DeleteFailException {
    Long id = null ;
    try {
        boolean result = false;
        id = type.getId() ;
        GroupType groupType = entityManager.find(GroupType.class, id);
        if (groupType != null) {
            entityManager.remove(groupType);
            result = true;
        }
        return result;
    } catch (Exception e) {
        throw new DeleteFailException(e,"delete GroupType error,id:[%s]",id);
    }
}
```

5.4 邮件发送异常

异常类：org.springframework.mail.MailException；为运行时异常

使用示例：com.bizenit.idm.sim.service.mail.MailSenderServiceImpl

```
@Override
public boolean sendHtmlMailWithAttachment(String[] addressees, String[] cc, String[] bcc, String subject,
        String content, String filePath) {
    try {
        JavaMailSenderImpl mailSender = mailSenderProvider.getMailSender();
        if (mailSender != null) {

            MimeMessage mimeMessage = mailSender.createMimeMessage();
            MimeMessageHelper messageHelper = new MimeMessageHelper(mimeMessage, true);
            if (!this.checkNull(addressees)) {
                messageHelper.setTo(addressees);
            }
            if (!this.checkNull(cc)) {
                messageHelper.setCc(cc);
            }
            if (!this.checkNull(bcc)) {
                messageHelper.setBcc(bcc);
            }
            messageHelper.setFrom(mailSender.getUsername());
            messageHelper.setSubject(subject);
            messageHelper.setText(content, true);
            FileSystemResource file = new FileSystemResource(new File(filePath));
            messageHelper.addAttachment(file.getFilename(), file);
            mailSender.send(mimeMessage);
            LOG.info(FINISH);
            return true;
        }
    } catch (PersistsException e) {
        LOG.error(ExceptionConsts.PERSISTS_EXCEPTION,e);
    } catch (MessagingException e) {
        LOG.error(ExceptionConsts.MESSAGING_EXCEPTION,e);
    } catch (MailException e) {
        LOG.error(ExceptionConsts.API_EXCEPTION,e);
    }
    return false;
}
```

JavaMailSenderImpl.send 方法API文档如下：

```
@Override
public void send(MimeMessage mimeMessage) throws MailException {
    send(new MimeMessage[] {mimeMessage});
}
```

5.5 邮件信息异常

异常类：javax.mail.MessagingException

使用示例：com.bizenit.idm.sim.service.mail.MailSenderServiceImpl

```java
public boolean sendHtmlMail(String[] addressees, String[] cc, String[] bcc, String subject, String content) {
    try {
        JavaMailSenderImpl mailSender = mailSenderProvider.getMailSender();
        if (mailSender != null) {
            MimeMessage mimeMessage = mailSender.createMimeMessage();
            MimeMessageHelper messageHelper = new MimeMessageHelper(mimeMessage);
            messageHelper.setFrom(mailSender.getUsername());
            if (!this.checkNull(addressees)) {□
            if (!this.checkNull(cc)) {□
            if (!this.checkNull(bcc)) {□
            messageHelper.setSubject(subject);
            messageHelper.setText(content, true);
            mailSender.send(mimeMessage);
            LOG.info(FINISH);
            return true;
        }
    } catch (PersistsException e) {
        LOG.error(ExceptionConsts.PERSISTS_EXCEPTION, e);
    } catch (MessagingException e) {
        LOG.error(ExceptionConsts.MESSAGING_EXCEPTION, e);
    }
    return false;
}
```

5.6 LDAP连接异常

异常类：com.bizenit.idm.sim.identity.provider.ConnectionException

使用示例：com.bizenit.idm.sim.identity.ldap.LdapGroupAdapterImpl.addUserToGroup

```java
Modification mod = new Modification(ModificationType.ADD, key, value);
try {
    ldapProvider.getConnectionPool().modify(groupBuilder.toString(), mod);
} catch (LDAPException e) {
    LOG.error(ExceptionConsts.LDAP_EXCEPTION, e);
} catch (ConnectionException e) {
    LOG.error(ExceptionConsts.CONNECTION_EXCEPTION, e);
}
```

5.7 LDAP操作异常

异常类：com.unboundid.ldap.sdk.LDAPException

使用示例：com.bizenit.idm.sim.identity.ldap.LdapGroupAdapterImpl.removeUrlFromGroup

```java
Modification mod = new Modification(ModificationType.DELETE, key, value);
try {
    ldapProvider.getConnectionPool().modify(groupBuilder.toString(), mod);
    removeFlag = true;
} catch (LDAPException e) {
    removeFlag = false;
    LOG.error(ExceptionConsts.LDAP_EXCEPTION, e);
} catch (ConnectionException e) {
    removeFlag = false;
    LOG.error(ExceptionConsts.CONNECTION_EXCEPTION, e);
}
```

5.8 LDAP查询异常

异常类：com.unboundid.ldap.sdk.LDAPSearchException

使用示例：com.bizenit.idm.sim.identity.ldap.LdapCommonAdapterImpl

```
@Override
public boolean checkEntryExist(LDAPConnectionPool conn, String baseDN, Filter filter)
    SearchResult searchResult = null;
    boolean result = false;
    try {
        if (conn != null) {
            searchResult = conn.search(baseDN, SearchScope.SUB, filter, "");
        }
        if (searchResult != null) {
            result = searchResult.getEntryCount() >= 1;
        }
        return result;
    } catch (LDAPSearchException e) {
        LOG.error(ExceptionConsts.LDAP_SEARCH_EXCEPTION, e);
        return false;
    }
}
```

5.9 日期转换异常

异常类：java.text.ParseException

使用示例：com.bizenit.idm.sim.service.appresource.impl.ITSubAccountServiceImpl

```
@Override
public String accountDateFormat(String dateStr) {
    try {
        String oldFormat = SystemConsts.FORMAT_YYYY_MM_DD_HH_MM_SS;
        if (dateStr.toUpperCase().endsWith("Z")) {
            oldFormat = SystemConsts.FORMAT_YYYYMMDDHHMMSSZ;
        } else if (dateStr.toUpperCase().indexOf('T') > -1) {
            oldFormat = SystemConsts.FORMAT_YYYY_MM_DDTHH_MM_SS;
        }

        Date date = DateUtils.getDate(dateStr, oldFormat);
        return DateUtils.getDateStr(date, SystemConsts.FORMAT_YYYY_MM_DD_HH_MM_SS);
    } catch (ParseException e) {
        LOG.error(ExceptionConsts.PARSE_EXCEPTION, e);
    }
    return "";
}
```

5.10 编码转换异常

异常类：java.io.UnsupportedEncodingException

使用示例：com.bizenit.idm.sim.service.mail.MailSenderServiceImpl

```
@Override
public boolean sendMailWithTemplate(String[] addressees, String[] cc, String[] bcc, String subject,
        String mailTemplateName, Map<String, Object> model)  {
    try {
        Object userStatus = model.get(LdapAttributes.USER_STATUS);
        boolean isUserDisable = StringUtil.isNotEmpty(userStatus) && SIMStatusEnum.DISABLED.getValue().equals(use
        if (isUserDisable){
        JavaMailSenderImpl mailSender = mailSenderProvider.getMailSender();
        TemplateData templateData = templateDataDao.getTemplateDataByName(mailTemplateName).orElse(null);
        if (mailSender != null && templateData != null && templateData.isEnable()) {
    } catch (PersistsException e) {
        LOG.error(ExceptionConsts.PERSISTS_EXCEPTION,e);
    } catch (UnsupportedEncodingException e) {
        LOG.error(ExceptionConsts.UNSUPPORTEDENCODING_EXCEPTION,e);
    }
    return false;
```

5.11 数字转化异常

异常类：java.io.NumberFormatException

使用示例：com.bizenit.idm.sim.identity.ldap.LdapProviderImpl

```java
@override
public boolean initialize() throws ConnectionException {
    try {
        List<Resource> resources = resourceDao.getEnabledResources(LdapResourceParameter.LDAP_RESOURCE_NAME);
        if (resources != null && !resources.isEmpty()) {
    } catch (NumberFormatException e) {
        LOG.error(ExceptionConsts.NUMBERFORMAT_EXCEPTION, e);
        return false;
    } catch (PersistException e) {
```

5.12 网络IO异常

异常类：java.io.IOException

使用示例：com.bizenit.idm.sim.service.rest.implApiServiceImpl.authAndCheckSign

```java
// 数字签名
String body = null;
try {
    ServletInputStream br = request.getInputStream();
    int count;
    int c;
    if (request.getContentLength() == -1) {
        count = request.getContentLength();
    }
    int i = 0;
    byte[] b = new byte[count];
    while ((c = br.read()) != -1) {
        b[i++] = (byte) c;
    }
    body = new String(b, "UTF-8");
} catch (IOException e) {
    LOG.error(ExceptionConsts.IO_EXCEPTION,e);
    return ApiResultUtil.systemError(RestCodeConsts.IO_ERROR);
}
```

5.13 未找到对应类异常

异常类：java.lang.ClassNotFoundException

使用示例：com.bizenit.idm.sim.service.appresource.impl.ITAccountSyncServiceImpl.accountFilter

```java
Class<?> cls = null;
try {
    cls = Class.forName(attrFilter.getFilterClass());
} catch (ClassNotFoundException e) {
    LOG.error(ExceptionConsts.CLASSNOTFOUND_EXCEPTION, e);
}
```

5.14 定制服务异常

异常类：com.bizenit.idm.sim.service.exception.ServiceException

使用示例：com.bizenit.idm.sim.service.appresource.filter.ITAccountAddOperDefaultFilter

```java
public AccountResult checkOperFilter(User user, ITAccount newItAccount, ITAccount oldItAccount,
        ITAccountAttrFilter attrFilter) throws ServiceException {
    AccountResult accountResult = new AccountResult();
    String sql = "";
    try {
        if (StringUtil.isEmpty(attrFilter)) {
            accountResult.setMessage("filter is not exsit");
            accountResult.setResultCode(AccountConstValue.ACCOUNT_FILTER_IS_NOT_EXSIT);
            accountResult.setSuccess(AccountConstValue.FAILURE);
            return accountResult;
        }
        if (AccountAttributes.UNIQUE.equals(attrFilter.getAction())) {
            accountResult.setMessage("operation and filter is not match");
            accountResult.setResultCode(AccountConstValue.OPERATION_FILTER_IS_NOT_MATCH);
            accountResult.setSuccess(AccountConstValue.FAILURE);
            return accountResult;
        }
    } catch (Exception e) {
        LOG.error(sql + SystemConsts.ERROR, e);
        throw new ServiceException(sql + SystemConsts.ERROR,e);
    }

    accountResult.setResultCode(AccountConstValue.SUCCESS_CODE);
    accountResult.setSuccess(AccountConstValue.SUCCESS);
    return accountResult;
}
```

5.15 汉字拼音转换异常

异常类：net.sourceforge.pinyin4j.format.exception.BadHanyuPinyinOutputFormatCombination

使用示例：com.bizenit.idm.sim.service.identity.ldap.LdapUserServiceImpl.checkUserAttrs

```java
boolean pyMatch = false;
try {
    pyMatch = PinyinUtils.isPinyinMatchChinese(usercn, cnPinyin);
} catch (BadHanyuPinyinOutputFormatCombination e) {
    LOG.error(ExceptionConsts.BAD_HANYUPINYIN_OUTPUTFORMATCOMBINATION,e);
}
```

5.16 base64转码异常

异常类：org.apache.ws.security.WSSecurityException

使用示例：com.bizenit.idm.sim.identity.ldap.LdapGroupAdapterImpl

```java
@Override
public GroupSearchResult simplePagedSearch(Filter filter, String searchBase, String cookie, int pageSize, String... returnAttributes)
    GroupSearchResult searchResults = new GroupSearchResult();
    try {
        LOG.error(ExceptionConsts.PERSISTS_EXCEPTION,e);
    } catch (AttributeNotFoundException e) {
        LOG.error(ExceptionConsts.ATTRIBUTE_NOTFOUND_EXCEPTION,e);
    } catch (WSSecurityException e) {
        LOG.error(ExceptionConsts.WSSECIRITY_EXCEPTION,e);
    } catch (ConnectionException e) {
        LOG.error(ExceptionConsts.CONNECTION_EXCEPTION,e);
    } catch (LDAPException e) {
        LOG.error(ExceptionConsts.LDAP_EXCEPTION,e);
    }
    return searchResults;
}
```

5.17 连接器异常

异常类：com.bizenit.idm.sim.prov.exception.ConnectorException

使用示例：com.bizenit.idm.sim.service.icf.impl.ObjectConnectorOperationImpl

```
    public String constuctDirectoryDN(SSMResource resource, SSMProvAddPolicy provAddPolicy,
                                SSMProvGlobalPolicy provGlobalPolicy, SyncDeltaObject user)  {
        String name = null;
        try {
            String majorKey = resource.getUniqueidentifier();
            SSMDBConnector dbConnector = resource.getDbConnector();
            String connectorName = dbConnector.getConnectorname();
            String userId = user.getUid();

            if (!StringUtils.checkNull(connectorName) && (connectorName.endsWith(SSMConsts.LDAP_SUFFIX_CONNECTOR_NAME)
                    || connectorName.endsWith(SSMConsts.EXCHANGE_SUFFIX_CONNECTOR_NAME))) {
                if (resource.getIstree() != null && resource.getIstree() == SSMResourceIsTreeEnum.TREE.getCode()) {
            } else {
                name = userId;
            }
        } catch (ProvisionException e) {
            LOG.error(SystemConsts.ERROR,e);
        } catch (ConnectorException e) {
            LOG.error(ExceptionConsts.CONNECTOR_EXCEPTION,e);
        }
        user.setDn(name);
        return name;
    }
```

5.18 证书异常

异常类：GeneralSecurityException

使用示例：com.bizenit.idm.sim.identity.ldap.LdapUserAdapterImpl

```
    @Override
    public boolean addUserByRegister(User user)  {

        try {
            LOG.error(ExceptionConsts.PERSISTS_EXCEPTION,e);
        } catch (UserAttributeNotFoundException e) {
            LOG.error(ExceptionConsts.USER_ATTRIBUTE_NOTFOUND_EXCEPTION,e);
        } catch (UserTypeNotFoundException e) {
            LOG.error(ExceptionConsts.USER_TYPE_NOTFOUND_EXCEPTION,e);
        } catch (GeneralSecurityException e) {
            LOG.error(ExceptionConsts.GENERAL_SECURITY_EXCEPTION,e);
        } catch (LDAPException e) {
            LOG.error(ExceptionConsts.LDAP_EXCEPTION,e);;
        } catch (ConnectionException e) {
            LOG.error(ExceptionConsts.CONNECTION_EXCEPTION,e);
        }

        return false;

    }

    /**
```

5.19 非法属性异常

异常类：com.bizenit.idm.sim.rest.exception.IllegalParamException

使用示例：com.bizenit.idm.sim.rest.utils.ParamCheckUtils

```
public  static AppResource checkAndGetAppResource(IAppResourceService appResourceService,String appId) throws  IllegalParamException,
    if (StringUtil.isNotEmpty(appId)) {
        AppResource appResource = appResourceService.getAppResourceByCode(appId).orElse(null);
        if (appResource == null) {
            throw  new IllegalParamException("app resource [%s] does not exists",appId);
        } else {
            return  appResource;
        }
    }
    return null;
}
```