

🏠 (/bin/view/Main/) ▾ / 05-TEG (/bin/view/5-TEG/) ▾ / 22-TEG-共享服务线-业务交易 (/bin/view/5-TEG/2 / 03-库存资源研发部 (/bin/view/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E / TC内部java开发规范 (/bin/view/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%

# TC内部java开发规范

在2021/03/12 16:55上被王亚奎(wangyakui) (/bin/view/XWiki/wangyakui)修改 被浏览了12次

- 1. 制定该规范的目的: 码出高效,码出质量.结合TC自己的业务场景,整理出适合自己的规范.
- 2. 当前版本规范主要包含下列内容:



## 代码编码规范

- 基本规范

### A. 命名规范

- 通用：以英文为主导的驼峰格式，不允许出现拼音命名和中文命名，非实际意义的数字不可使用。
- 【正例】expense, name
- 【反例】feiyong, name1
- (1) 【强制】常量以下划线拼接，全大写格式。常量应抽离放在类级别定义或常量类中单独定义，不能在
- 【正例】EXPENSE\_DETAIL = “expense\_detail”
- 【反例】expense\_detail = “expense\_detail”、expenseDetail = “expense\_detail”
- (2) 【推荐】命名推荐全称英文，建议不要出现缩写、业界公认的缩写名称可使用。以系统固有使用规
- 【正例】conditionDetail、mvc、statistic
- 【反例】conDetail、stat
- (3) 【强制】各模式命名以具体的后缀结尾。方便理解和协同开发。
- 【正例】DefaultException、ExpenseTest.
- 【反例】DefaultExceptions(异常类)、ExpenseDemo(测试类)
- (4) 【强制】包名统一用小写。不允许出现复数形式包名
- 【正例】com.bj58.coin.utils、com.bj58.coin.controller
- 【反例】com.bj58.coin.coinModel
- (5) 【强制】领域模型对象如DTO/BO/DO/VO/PO，模型类命名应加响应模型后缀，且后缀全为大写。
- 【正例】ExpenseDTO、ExpenseVO
- 【反例】ExpenseDto
- (6) 【推荐】Dao层命名规约。
- l. 获取单个对象方法。用get做方法名前缀。
- 【正例】getExpense
- 【反例】selectExpense、searchExpense、findExpense、queryExpense
- l. 获取集合对象方法。用list做方法名前缀。
- 【正例】listExpense

收缩

【反例】getExpense、getExpenses、selectExpenseList

I. 获取统计值方法。用count做方法名前缀或后缀

【正例】countExpense

【反例】getExpenseCount、selectCountOfExpense

I. 新增对象方法。用add做方法名前缀。

【正例】addExpense

【反例】insertExpense

I. 删除方法。用remove做方法名前缀。

【正例】removeExpense

【反例】updateExpense、clearExpense、delExpense、deleteExpense

I. 更新方法。用modify做方法名前缀。逻辑删除亦然。

【正例】modifyExpense

【反例】changeExpense、updateExpense

B. 代码格式

(1) 【推荐】大括号使用约定，若大括号内为空。简洁写成{}即可。否则：除左大括号左边不需要换行外

【正例】

```
if ( a == b ) {  
    // TODO  
}
```

【反例1】

```
if (a == b){ // TODO }
```

【反例2】

```
if (a == b)  
{  
  
    // TODO  
}
```

【反例3】

```
if (a == b)  
{// TODO }
```

(2) 【推荐】数学运算符左右两边均需要空格。例如：

【正例】if (a == b)

【反例】if (a==b)、if (a== b)、if (a ==b)

(3) 【推荐】if/for/while/switch/do等保留字与括号之间都必须加空格。

【正例】if (a == b)、while (a == 1)

【反例】if(a==b)、while(a == 1)

(4) 【推荐】注释符号与注释内容之间有且仅有一个空格。

【正例】// 注释

【反例】//注释

(5) 【推荐】方法调用时，需要多个换行时，在点号或逗号前换行。

【正例】

```
CoinUtil.toggle(condition_one  
    , condition_two  
    , condition_three  
    , condition_four);
```

【反例】

```
CoinUtil.toggle(condition_one,  
    condition_two,  
    condition_three,  
    condition_four);
```

(6) 【推荐】方法参数在定义或调用时，多个参数逗号后必须加空格。

【正例】CoinUtil.toggle(one, two, three);

【反例】CoinUtil.toggle(one,two,three);

收缩

(7) Object类的equals()方法容易抛空指针异常。使用常量或确定的值来调用equals。例如：

【正例】“test”.equals(object)

【反例】

```
String object = null;

// 空指针异常

object.equals(“test”);
```

(8) 【推荐】所有的领域模型对象中，属性不允许使用基本类型定义。防止基本类型的默认值给业务使用。

【正例】Integer a; Boolean deletedFlag;

【反例】int a; boolean isDeleted;

(9) 【强制】领域模型对象不允许改造其getter或setter方法。亦不允许初始化属性值。分场景？？？

【正例】

```
public void setPrjId(long prjId) {

    this.prjId = prjId;

}
```

【反例】

```
public void setPrjId(Long prjId) {

    this.prjId = (null == prjId) ? 0L : prjId;

}
```

(10) 【强制】业务逻辑中查询产生的模型对象或集合对象都需要进行非空判断来避免空指针异常。若已

【正例】

```
List<Expense> expenseList = coinService.listExpense();

if (CollectionUtils.isEmpty(expenseList)) {

    // TODO

}
```

【反例】

```
List<Expense> expenseList = coinService.listExpense();

// TODO （直接使用expenseList可能产生空指针异常）
```

(11) 【推荐】使用StringBuilder类的append()方法来代替String类的“+”号拼接字符串操作。

【正例】StringBuilder date = new StringBuilder(“2020”)

```
        .append(“-”).append(“09”)
        .append(“-”).append(“21”);
```

【反例】String date = a + b + c + “-” + “21”;

(13) 【推荐】类中存在相似或相仿的方法时，应紧邻放在一起，例如getWord()和getWordForSometh

(14) 【推荐】if/else/for/while/do语句中必须使用大括号，即使只有一行代码。

【正例】

```
if (a == b) {

    a++;

}
```

【反例】

```
if (a == b) a++;
```

(15) 【推荐】if...else嵌套语句不得超过三层。采用取反、短路结束逻辑

【正例】

```
if (a != c) {

    return;

}

if (a == b) {

    a++;

}
```

【反例】

```
if (a == b) {

    if (a == c) {

        if ( a == d) {
```

```
        if (a == e) {
            a++;
        }
    }
}
```

(16) 【强制】外部传入的入参对象不能直接传入Dao层。应经过model转换后再传入底层。

【正例】

```
public void update(CoinDTO record) {
    CoinModel coinModel = new CoinModel();
    BeanUtils. copyProperties(record, coinModel);
    coinService.update(coinModel);
}
```

【反例】

```
public void update(CoinDTO record) {
    coinService.update(record);
}
```

C. 异常规范

- (1) 【推荐】 try...catch保证最小原则，即拒绝将大量已知稳定的业务逻辑放入其中。因为try...catch影
- (2) 【强制】 非兜底捕获时，使用具体的异常类，而非用Exception一概而全。

【正例】

```
public static Userinfo baseGetUserById(String userId) {
    try {
        Userinfo userinfo = Provider.getUserService().getUserById(userId);
        return userinfo;
    } catch (BspException e) {
        LoggerUtil.error( msg: "bsp-根据员工id查询员工信息异常!", e);
    }
    return null;
}
```

【反例】

```
public static Userinfo baseGetUserById(String userId) {
    try {
        Userinfo userinfo = Provider.getUserService().getUserById(userId);
        return userinfo;
    } catch (Exception e) {
        LoggerUtil.error( msg: "bsp-根据员工id查询员工信息异常!", e);
    }
    return null;
}
```

- (3) 【强制】 禁止捕获异常后catch块中处理逻辑为空。

【正例】

```
public static Userinfo baseGetUserById(String userId) {
    try {
        Userinfo userinfo = Provider.getUserService().getUserById(userId);
        return userinfo;
    } catch (BspException e) {
        LoggerUtil.error( msg: "bsp-根据员工id查询员工信息异常!", e);
    }
    return null;
}
```

【反例】

```
public static Userinfo baseGetUserById(String userId) {
    try {
        Userinfo userinfo = Provider.getUserService().getUserById(userId);
        return userinfo;
    } catch (BspException e) {
    }
    return null;
}
```

- (4) 【强制】 try...catch应采用最上层处理原则，拒绝每一个调用层内均增加try...catch。底层应选择

【正例】

```
public void methodOne() {
    try {
        methodTwo();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void methodTwo() throws Exception {
    methodThree();
    throw new ArrayIndexOutOfBoundsException("数组越界异常");
}

public void methodThree() throws Exception {
    methodFour();
    throw new IOException("IO异常");
}

public void methodFour() throws Exception {
    throw new IllegalArgumentException("空指针异常");
}
```

【反例】

```
public void methodOne() {
    try {
        methodTwo();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void methodTwo() {
    try {
        methodThree();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void methodThree() throws Exception {
    try {
        methodFour();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void methodFour() throws Exception {
    try {
        throw new IllegalArgumentException("空指针异常");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

• 注释规范

- (1·) 【强制】类、类属性、类方法的注释必须使用Javadoc规范。即使用“/\*\* 内容 \*/”格式。不得使用

【正例】

```
/**
 * P+核心服务
 * 删除费用单价配置 支持批量、逐条删除
 * @param priceConfigDTOList
 * @param modelBase
 * @return
 */
public static Boolean deleteExpensePriceBatch(List<PriceConfigDTO> priceConfigDTOList, ModelBase modelBase) {
```

【反例】

```
// P+核心服务 删除费用单价配置 支持批量、逐条删除
public static Boolean deleteExpensePriceBatch(List<PriceConfigDTO> priceConfigDTOList, ModelBase modelBase) {
```

- (2) 【推荐】类、接口、文件都需注明@author属性和@date属性。增加当前类描述说明。

【正例】

```
/**
 * @author ningkuizhang
 * @date 2020/6/10 15:17
 */
public class PPlusServiceTest {
```

【反例】

```
public class PPlusServiceTest {
```

- (3) 【强制】方法内部单行注释，在备注语句上方另起一行，使用//注释。方法内部多行注释使用/\* \*/

【正例】

```
public void methodOne() {
    // 当前申请费用ID
    String expenseId = beat.getRequest().getParameter("expenseId");
    String applyId = beat.getRequest().getParameter("applyId");
    /**
     * 传入参数：费用ID、费用申请ID
     * 返回值：费用申请明细集合
     * 主要用于后续的申请工作流提供数据支持
     */
    List<PrjExpenseApplyDetail> expenseApplyDetails = expenseApplyDetailManager.getExpenseApplyDetailList(Long.valueOf(expenseId), Long.valueOf(applyId));
}
```

【反例】

```
public void methodOne() {
    String expenseId = beat.getRequest().getParameter("expenseId"); // 当前申请费用ID
    String applyId = beat.getRequest().getParameter("applyId"); // 当前费用申请ID

    // 传入参数：费用ID、费用申请ID
    // 返回值：费用申请明细集合
    // 主要用于后续的申请工作流提供数据支持
    List<PrjExpenseApplyDetail> expenseApplyDetails = expenseApplyDetailManager.getExpenseApplyDetailList(Long.valueOf(expenseId), Long.valueOf(applyId));
}
```

- (4) 【强制】所有的枚举字段、常量字段均需要注释说明。说明具体用途。

【正例】

```
20 public class LoggerUtil {
21
22     /**
23      * 日志文件名 常量 -- 常规日志。
24      */
25     public static final String INFO = "info";
26
27     /**
28      * 日志文件名 常量 -- 错误日志。
29      */
30     public static final String ERROR = "error";
31 }
```

【反例】

```
public static final String ASC_LOWER = "asc";
public static final String FQLX = "FQLX";
public static final String CXTZ = "CXTZ";
public static final String NEXT = "NEXT";
public static final String END = "NEXT";
public static final String FORW = "FORW";

public static final String ZBFZRS = "ZBFZRS";
public static final String QYFZRS = "QYFZRS";
public static final String QYFZRS_CAM = "QYFZRS-CAM";
public static final String QYFZRS_IB = "QYFZRS-IB";
public static final String QYFZRS_RPO = "QYFZRS-RPO";
//征询单新增类型 开始
public static final String QYFZRS_EVAL = "QYFZRS-EVAL";
public static final String QYFZRS_TRAIN = "QYFZRS-TRAIN";
public static final String QYFZRS_CAMSTR = "QYFZRS-CAMSTR";
//征询单新增类型 结束
public static final String QYFZRS_BJHS = "QYFZRS-BJHS";
public static final String XSZBFZRTS = "XSZBFZRTS";
public static final String XSZBFZRTS_KA = "XSZBFZRTS-KA";
```

- (5) 【强制】代码修改后、相应的参数、返回值、异常、核心逻辑释义也要做相应修改。
- (6) 【强制】谨慎注释掉代码。做详细说明、如注释原因、后续是否会恢复。若永不再用，则直接删除
- (7) 待后续完成的功能。可使用TODO标注。代码错误无法工作的可使用FIXME标注。

【正例】存在TODO的方法，协作开发使用前会检查、更谨慎选择使用。

```
@Override
public String toString() {
    // TODO Auto-generated method stub
    return code + "-" + supplierName + "-" + bankName + "-" + bandCarNum + "-" + parentCode;
}
```

```
// 根据prjCode查询项目
Project projectCondition = new Project();
projectCondition.setPrjCode(prjCode);
Project project = null;
// FIXME 查询项目编号
List<Project> projectList = projectBiz.selectByConditionWithOrderBy(projectCondition, "orderBy: \"prj_id desc\"");
if (null == projectList || projectList.isEmpty()) {
    result.setResultCode(PPlusResultCode.PARAM_ERROR);
    result.setResultMessage("暂无对应项目!");
    result.setSuccess(false);
    return result;
}
```

【反例】

```
@Override
public String toString() {
    // Auto-generated method stub
    return code + "-" + supplierName + "-" + bankName + "-" + bandCarNum + "-" + parentCode;
}
```

• 依赖组件操作规范

• CDB使用规范

A. 建表规范

- (1) 【强制】主键必须为“id”，不允许自定义主键名，建表需有COMMENT属性，注明每个字段的含义，
- (2) 【强制】不允许使用外键约束。
- (3) 【强制】表名中的单词没有复数一说,因为表名只表达的是当前表的内容。

【反例】如表名 orders

- (4) 【推荐】表名一定要包含一定的业务含义,如果表名是关键字的时候最好加修饰字段.不要只使用单个字母。

【反例】如表名 order 【正例】pay\_order, resource\_order

- (5) 【强制】表名禁用保留字段: desc ,like, status等。

保留字段见链接: <https://www.cnblogs.com/zhuyeshen/p/11496545.html> (<https://www.cnblogs.com/zhuyeshen/p/11496545.html>)

- (6) 【强制】表存储引擎仅仅支持InnoDB，且必须指明ENGINE=InnoDB。
- (7) 【推荐】表必备三字段：id, created\_at, modified\_at,创建人，id表示主键, create\_time, modified\_time。
- (8) 【推荐】普通索引名称必须以idx\_开头，唯一索引必须以uniq\_开头，且索引名越简洁越好（越长占用空间越多）。
- 【正例】给create\_at,user\_id建联合索引：idx\_create\_at\_user\_id

- (9) 【推荐】大字段Text,varchar字段超过128以上不建议建立索引，varchar字段长度设置为2\*N次方。说明：索引的长度与区分度是一对矛盾体，一般对字符串类型数据，长度为20的索引，区分度会高；索引很长的字符列,会让索引变得大且慢.部分字段建索引，可以节约索引空间，从而提高索引效率

- (10) 【强制】小数类型使用decimal格式,按需要设置小数位。禁止使用float和double格式。
- (11) 【推荐】使用数字类型(Long,int)等尽量使用unsigned类型来标注,int类型建议统一使用11位长度。
- (12) 【强制】Long、varchar使用2的倍数设置字段长度。
- (13) 【强制】字段均需设置非空属性,varchar字段类型默认值是""（不能设置为Null）

B. 改表规范

- (1) 【强制】新增字段时、不能固定化直接放在所有字段之后，适当运用after/before方法。
- (2) 【强制】新建字段时必须指定默认值。且默认值不能为null
- (3) 【强制】更新表字段名时、若注释已改、相应的COMMENT值也要随之变化。
- (4) 【强制】数据库操作规范(DDL):包括增加,修改,删除表字段,增加索引，清理表碎片等操作.所有的DDL操作都必须在低峰期操作(高峰期的操作时容易引起主从数据不一致,进而影响交易)。
- b. 需和DBA沟通清楚,当前操作是否支持批量操作，是否会进行批量操作，当前操作的影响范围等信息。
- mongoDB的关于表的操作：同(4)

C. 数据操作规范

- (1) 【推荐】修复数据新增时、创建人应注明实际操作人。且时间为当前时间。
- (2) 【推荐】修复数据更新时、更新人应注明实际操作人。且时间为当前时间。
- (3) 【强制】删除数据时。自己做好数据备份。做好及时回滚的准备。
- (4) 【推荐】超过三表的联表查询需关注执行效率、谨慎使用。
- (5) 【强制】查询sql编写若非特定需求，禁止接口返回超过200条数据，需做分页。
- (6) 【推荐】查询SQL排序使用注意索引的顺序规则。
- (7) 【强制】sql条件里边的集合条件尽量用in选择确定值。

原始dao接口:

```
List<SplitSiteDetail> getUseableSplitSiteDetailList(@Param("cityIdList") List<Integer> cityIdList, @Param("productCode") String productCode, @Param("statusList") List<Integer> statusList);
```

【反例】状态值直接取范围区间

```
<if test="statusStart != null">
    <![CDATA[
        AND status >= #{statusStart}
    ]]>
</if>
<if test="statusEnd != null">
    <![CDATA[
        AND status <= #{statusEnd}
    ]]>
</if>
```

【正例】状态值取确定的范围

```
<if test="statusList.size()>0">
    AND status in
    <foreach collection="statusList" index="index" item="item" open="(" close=")" separator=",">
        #{item}
    </foreach>
</if>
```

- (8) 【推荐】在表查询中，不要使用 \* 作为查询的字段列表,需要哪些字段必须明确注明  
说明:(1) 增加查询分析器解析成本。  
(2) 无用字段增加网络消耗，尤其是text类型的字段
- (9) 【推荐】统计数量时请使用count(\*), count(\*)是SQL92定义的标准统计行数的语法，跟数据库无关。
- (10) 【强制】当某一列的值全是NULL时，count(col)的返回结果为0，但sum(col)的返回结果为NULL，因  
【正例】可以使用如下方式来避免sum的NPE问题：  
SELECT IFNULL(SUM(column), 0) FROM table;
- (11) 【推荐】不要写一个大而全的数据更新接口。传入为POJO类，不管是不是自己的目标更新字段,都进  
(12) 【推荐】Like查询时使用<bind name="bindeName" value="'%'+eName+'%'" />仅限于mybatis

• SCF使用规范

- (1) 【强制】序列化目前不支持内部类。
- (2) 【强制】接口定义不需要加public 等访问权限控制符。
- (3) 【强制】序列化不支持map类型key为null。
- (5) 序列化不支持 File, InputStream, JSON 等没有加@SCFSerializable的java对象。
- (6) 序列化SCFV3不支持嵌套的泛型，如List<Map<String, Object>>, 这种类型SCFV4可以支持（SCF
- (7) 不支持JDK11，扫包会出现问题
- (8) 【推荐】对外和对内的SCF接口应单独拆分类使用。即使功能相同。  
对外提供的接口粒度最小化。
- (9) 【强制】返回值使用固定包装类。拒绝数据裸奔。响应码，响应消息等信息。示例：

【正例】

```
@OperationContract
PPlusResult<Boolean> alterActualQuantityAndPayType(FeeAlterDTO feeAlterDTO);
```

【反例】

```
@OperationContract
public String toBindProject(String contractNo, long prjId, String saleId, String operator, double contractMoney) throws Exception;
```

- (10) 【推荐】返回结果若为集合时，应返回空集合，而非Null。

【正例】

```
return new PPlusResult<List<AuditNodeVO>>() {
    isSuccess: true
    , PPlusResultCode.SUCCESS
    , null == scfAuditNodeList ? new ArrayList<>() : scfAuditNodeList;
};
```

【反例】

```
List<AuditNodeVO> list = null;
return new PPlusResult<List<AuditNodeVO>>() {
    isSuccess: true
    , PPlusResultCode.SUCCESS
    , list;
};
```

• WF使用规范

- (1) 【强制】所有的Controller在com.bj58.\*.controllers下。
- (2) 【强制】所有的Controller必须是Controller结尾。





TC内部java开发规范 - iwiki	
回滚方案	
回滚版本	

- 2. 【推荐】区分正式jar包和快照jar包两套规则,不能跳包。例如例如正式为包1.0.0，最新快照包为1.0
- 3. 【推荐】使用jdk版本1.8;
- 4. 【推荐】Jvm参数设置:

设置metaspace: -XX:MetaspaceSize=256M -XX:MaxMetaspaceSize=256M

设置堆内存大小:最大值和最小值一样 -Xms4g -Xmx4g

设置测试环境远程debug端口:

-agentlib:jdwp=transport=dt\_socket,server=y,suspend=n,address=5855

设置出现oom时自动dump内存文件快照:

-XX:+HeapDumpOnOutOfMemoryError

设置dump的文件目录 -XX:HeapDumpPath=\${目录}

设置打印GC日志:

-XX:+PrintGC 输出GC日志

-XX:+PrintGCDetails 输出GC的详细日志

-XX:+PrintGCTimeStamps 输出GC的时间戳（以基准时间的形式）

-XX:+PrintGCDateStamps 输出GC的时间戳（以日期的形式，如 2013-05-04T21:53:59.234+0

• 日志打印规范

• 日志规范

- (1) 【强制】日志需注明调用时间。（日志初始化配置应有相关配置）
- (2) 【推荐】日志进行具体分类。但不应过细。推荐日志分类不超过8个。
- (3) 【推荐】日志内容应正确使用占位符，而非字符串拼接方法。

【正例】

logger.info(">>>>updateCity:更新城市操作完成.result={},userId={}", result, cityProcessEntity.getUserId());

反例】

```
/**
 * P+核心服务
 * 项目变更 --- 主要是做复制操作
 * @param param
 * @return
 */
public static Project projectChange(Long prjId, String opUserId, String opUserName) throws Exception {
    LoggerUtil.scf( "项目变更: param=" + prjId + ", " + opUserId + ", " + opUserName);
    PPlusResult<Project> pPlusResult = pPlusManageService.projectChange(prjId, opUserId, opUserName);
    LoggerUtil.scf( "结果: " + GsonUtil.toJson(pPlusResult));
    if(pPlusResult == null || !pPlusResult.isSuccess()){
        throw new Exception("项目变更接口异常: " + pPlusResult.getResultMessage());
    }
    return pPlusResult.getData();
}
```

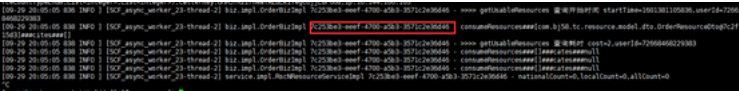
- (4) 【推荐】对于集合类结果的打印需要考虑内存的承受，需要防止数据过大将内存击垮。可以采用打印集合大小或者采用debug级别模式，在需要的时候再打印集合内容的方式

【反例】

```
pageIndex = pageIndex == null ? 1 : pageIndex;
pageSize = pageSize == null ? 500 : pageSize;
list<UserInfo> userInfoList = Provider.getRoleService().getUsersByRoleIdNew(pageIndex, pageSize, roleId);
LoggerUtil.info( "查询结果: %s ", GsonUtil.toJson(userInfoList));
```

- (5) 【强制】异常捕获后必须增加日志。且打印异常信息Exception。捕获后不处理是埋雷操作。
- (6) 【推荐】交易类接口需要打印业务的请求参数和响应参数,以方便排查业务调用相关的问题。
- (7) 【强制】日志打印内容需包括完整的类路径名称,以及交易的主标识信息，配置信息参考如下:

```
<RollingRandomAccessFile name="activeInfo" fileName="../log/nresource/nresource-log.log"
filePattern="../log/nresource/nresource-log.%d{yyyy-MM-dd-HH}.log">
  <Filters>
    <ThresholdFilter level="WARN" onMatch="DENY" onMismatch="NEUTRAL"/>
    <ThresholdFilter level="INFO" onMatch="ACCEPT" onMismatch="DENY"/>
  </Filters>
  <PatternLayout>
    <Pattern>%d{MM-dd HH:mm:ss SSS} %-5level [Xt] %c{3} %X(requestid) - %m%n%ex</Pattern>
  </PatternLayout>
  <Policies>
    <TimeBasedTriggeringPolicy/>
  </Policies>
</RollingRandomAccessFile>
```



Error级别的日志按重要度可以考虑输出到邮件组,以供及时介入问题。

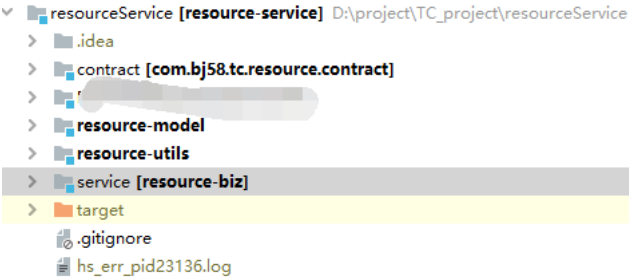
```
<RollingRandomAccessFile name="activeError" fileName="../log/nresource/error.log"
  filePattern="../log/nresource/error.%d{yyyy-MM-dd}.log">
  <ThresholdFilter level="ERROR" onMatch="ACCEPT" onMismatch="DENY"/>
  <PatternLayout>
    <Pattern>[%d{MM-dd HH:mm:ss SSS} %-5level] [%t] %c{3} %X(requestid) - %m%n%ex</Pattern>
  </PatternLayout>
  <Policies>
    <TimeBasedTriggeringPolicy/>
  </Policies>
</RollingRandomAccessFile>
```

邮件信息配置如下:

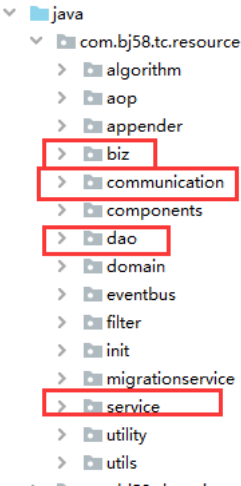
```
<SMTP name="mailer"
  subject="resourceService Error Alert"
  to="trading@58.com"
  from="tradingmonitor@58.com"
  smtpHost="smtp.exmail.qq.com"
  smtpUsername="tradingmonitor@58.com"
  smtpPassword="Trading9"
  smtpDebug="false"
  bufferSize="0">
  <Filters>
    <PackageNameFilter level="ERROR" onMatch="DENY" onMismatch="NEUTRAL" >
      <packageName>com.bj58.spat,com.bj58.srvmgr</packageName>
    </PackageNameFilter>
    <ThresholdFilter level="ERROR" onMatch="ACCEPT" onMismatch="DENY"/>
    <BurstFilter level="ERROR" rate="0.2" maxBurst="1" onMatch="ACCEPT" onMismatch="DENY"/>
  </Filters>
  <PatternLayout charset="utf-8">
    <Pattern>主机: ${sys:ip} [%d{MM-dd HH:mm:ss SSS} %-5level] [%t] %c{3} %X(requestid) - %m%n%ex</Pattern>
  </PatternLayout>
</SMTP>
```

## 五. 服务架构

1. 整体工程可以分为对外协议(contract), 业务实现(service)



Service层用来进行具体的业务实现,可以划分为component, communication,dao,biz层等。



component: 参数校验, 模型转换


biz处理真正的业务逻辑

communication负责与第三方系统通信交互, 比如操作日志, 与摊销交互;

dao 负责mapper,sql操作底层db或redis

收缩



	<p>1606553936738-830.png (<a 1606553983988-986.png"="" data-bbox="663 329 726 347" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 16:58上 (89.1 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;img alt=" of="" thumbnail=""></a></p>	<p>1606553983988-986.png (<a 1606553995476-600.png"="" data-bbox="663 519 726 537" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 16:59上 (24.1 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;img alt=" of="" thumbnail=""></a></p>	<p>1606553995476-600.png (<a 1606554015892-864.png"="" data-bbox="663 710 726 728" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 16:59上 (15.1 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;img alt=" of="" thumbnail=""></a></p>	<p>1606554015892-864.png (<a 1606554025157-418.png"="" data-bbox="663 900 726 918" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:00上 (15.8 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;img alt=" of="" thumbnail=""></a></p>	<p>1606554025157-418.png (<a 1606554055257-193.png"="" data-bbox="663 1090 726 1108" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:00上 (6.0 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;img alt=" of="" thumbnail=""></a></p>	<p>1606554055257-193.png (<a 1606554065280-610.png"="" data-bbox="663 1281 726 1299" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:00上 (28.3 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;img alt=" of="" thumbnail=""></a></p>	<p>1606554065280-610.png (<a 1606554075842-718.png"="" data-bbox="663 1471 726 1489" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:01上 (27.4 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;img alt=" of="" thumbnail=""></a></p>	<p>1606554075842-718.png (<a 1606554088630-646.png"="" data-bbox="663 1662 726 1680" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:01上 (34.6 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;img alt=" of="" thumbnail=""></a></p>	<p>1606554088630-646.png (<a 1606554115012-124.png"="" data-bbox="663 1852 726 1870" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:01上 (160.9 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;img alt=" of="" thumbnail=""></a></p>	<p>1606554115012-124.png (<a 1606554129078-739.png"="" data-bbox="663 2042 726 2060" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:01上 (18.9 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;img alt=" of="" thumbnail=""></a></p>	<p>1606554129078-739.png (<a 365="" 385="" 921="" 932"="" data-label="Text" href="https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E99A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A71.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:01上 (18.9 KB)&lt;/a&gt;&lt;/p&gt;&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/div&gt;&lt;div data-bbox="><p>收缩</p></a></p>
---	---	---	---	---	--	---	---	---	--	---	--



	<div>TC内部java开发规范 - iwiki</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:02上 (47.5 KB)</div>
	<div>1606554144282-250.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:02上 (18.3 KB)</div>
	<div>1606554304234-869.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:05上 (15.8 KB)</div>
	<div>1606554322874-680.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:05上 (11.8 KB)</div>
	<div>1606554347676-805.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:05上 (25.0 KB)</div>
	<div>1606554356195-432.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:05上 (21.2 KB)</div>
	<div>1606554363200-272.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:06上 (21.2 KB)</div>
	<div>1606554377099-795.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:06上 (19.4 KB)</div>
	<div>1606554388464-984.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:06上 (17.3 KB)</div>
	<div>1606554404217-769.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:06上 (26.2 KB)</div>
	<div>1606554489483-961.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:08上 (10.4 KB)</div>
	<div>1606554506233-411.png (https://wos.58cdn.com.cn/pQnMIKjpQSW/iwikibuck%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98_TC%E5%86%85%E9%83%A8java%E5%BC%8)</div> <div>9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7</div> <div>1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B</div> <div>%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8</div> <div>由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:08上 (10.4 KB)</div>

%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:08上 (66.7 KB)

1606554520662-344.png (https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98\_TC%E5%86%85%E9  
9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7  
1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:08上 (29.1 KB)

1606554541727-254.png (https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98\_TC%E5%86%85%E9  
9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7  
1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:09上 (73.5 KB)

1606554554344-639.png (https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98\_TC%E5%86%85%E9  
9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7  
1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:09上 (52.7 KB)

1606554566635-809.png (https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98\_TC%E5%86%85%E9  
9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7  
1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:09上 (57.8 KB)

1606554579109-272.png (https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98\_TC%E5%86%85%E9  
9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7  
1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:09上 (79.6 KB)

1606554611493-334.png (https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98\_TC%E5%86%85%E9  
9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7  
1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:10上 (12.5 KB)

1606554629580-847.png (https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98\_TC%E5%86%85%E9  
9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7  
1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:10上 (10.0 KB)

1606555063085-711.png (https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98\_TC%E5%86%85%E9  
9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7  
1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:17上 (22.6 KB)

1606555079966-341.png (https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98\_TC%E5%86%85%E9  
9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7  
1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:17上 (36.2 KB)

1606555131340-968.png (https://wos.58cdn.com.cn/pQnMLKjpQSW/iwikibuck  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98\_TC%E5%86%85%E9  
9A%25E5%258A%25A1%25E4%25BA%25A4%25E6%2598%2593%2F%25C2%25B703-%25E7%25A7  
1.1 (/bin/viewattachrev/5-TEG/22-TEG-%E5%85%B1%E4%BA%AB%E6%9C%8D%E5%8A%A1%E7%B  
%E7%A7%AF%E5%88%86%E5%BA%93%E5%AD%98/TC%E5%86%85%E9%83%A8java%E5%BC%8  
由刘创学(liuchuangxue) (/bin/view/XWiki/liuchuangxue)发布 在2020/11/28 17:18上 (14.6 KB)

收缩

绑定文件至该文档

选择文件 未选择任何文件

