**表8 业务逻辑层模块的职责**

**表8 业务逻辑层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| **userbl** | 负责实现对应于用户登录，管理员增删改查用户界面所需要的服务 |
| **managebl** | 负责实现对应于总经理增删改查机构、管理人员、设置基础数据、审批单据、查看经营情况表界面所需要的服务 |
| **repertorybl** | 负责实现仓库管理员对商品进行出、入库管理,库存初始化、报警，进行库存查看和库存盘点界面所需要的服务 |
| **receiptbl** | 负责实现所有单据的审批，修改各项单据业务所需的服务 |

**5.3.2业务逻辑层模块的接口规范**

**表9 userbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| User.login | 语法 | public int login(String userID, String password) |
| 前置条件 | 快递物流系统的用户在登录界面输入了自己的用户编号和密码 |
| 后置条件 | 查找是否存在编号为userID的用户，根据输入的password返回登录验证的结果 |
| User.addUser | 语法 | public int addUser(UserVO uservo) |
| 前置条件 | 经过验证的管理员有添加新用户的权利，且管理员输入了完整有效的新用户信息 |
| 后置条件 | 系统保存新增的用户信息并更新用户列表 |
| User.deleteUser | 语法 | public int deleteUser(UserVO uservo) |
| 前置条件 | 经过验证的管理员有删除用户的权利，且管理员输入了有效的待删除用户信息 |
| 后置条件 | 系统查找对应的用户信息，并删除该用户对象 |
| User.modifyUserPassword | 语法 | public int modifyUserPassword (UserVO uservo) |
| 前置条件 | 系统中每个用户有修改自己密码的权利，且该用户输入了正确的原有密码和有效的新密码 |
| 后置条件 | 系统根据用户编号和输入的新密码更新用户信息 |
| User.modifyUserAuthority | 语法 | public int modifyUserAuthority(UserVO uservo) |
| 前置条件 | 经过验证的管理员有更改用户权限的权利，且管理员输入了有效的待更改用户信息 |
| 后置条件 | 系统根据用户编号和新的用户权限更新用户信息 |
| User.findUser | 语法 | public UserVO findUser(String userID); |
| 前置条件 | 经过验证的管理员或总经理有查询用户的权利，且管理员或总经理输入了有效的待查询用户编号 |
| 后置条件 | 系统根据输入的用户编号查找用户，并返回用户信息 |
| User.showAllUsers | 语法 | public ArrayList<UserVO> showAllUsers() |
| 前置条件 | 经过验证的管理员或总经理有查询用户的权利，且管理员或总经理没有输入 |
| 后置条件 | 系统返回所有的用户信息 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| FileFactory.getUserData | | 得到User文件的引用 |
| UserDataService.check(String userID ,String password) | | 得到根据输入的ID和password验证的结果 |
| UserDataService.add(UserPO userpo) | | 在用户数据文件中插入一个UserPO对象 |
| UserDataService.delete(UserPO userpo) | | 在用户数据文件中删除一个UserPO对象 |
| userDataService.modify(UserPO userpo) | | 在用户数据文件中更新一个UserPO对象 |
| userDataService.find (String userID) | | 根据输入的用户编号，返回对应的UserPO对象 |
| userDataService.showAll () | | 返回所有的UserPO对象 |

**表9 managebl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Manage.addOrganization | 语法 | public int addOrganization(OrganizationVO organizationvo) | |
| 前置条件 | 经过验证的总经理有添加新机构的权利，且总经理输入了完整有效的新机构信息 | |
| 后置条件 | 系统保存新增的机构信息并更新机构列表 | |
| Manage.deleteOrganization | 语法 | public int deleteOrganization(OrganizationVO organizationvo) | |
| 前置条件 | 经过验证的总经理有删除机构的权利，且总经理输入了有效的待删除机构信息 | |
| 后置条件 | 系统查找对应的机构信息，并删除该机构对象 | |
| Manage.modifyOrganization | 语法 | public int modifyOrganization(OrganizationVO organizationvo) | |
| 前置条件 | 经过验证的总经理有修改机构信息的权利，且总经理输入了有效的待修改机构信息 | |
| 后置条件 | 系统更新机构信息 | |
| Manage.findOrganization | 语法 | public OrganizationVO findOrganization(String organizationID) | |
| 前置条件 | 经过验证的总经理有查询机构的权利，且总经理输入了有效的待查询机构编号 | |
| 后置条件 | 系统根据输入的机构编号查找机构，并返回机构信息 | |
| Manage.chooseDepartment | 语法 | public int chooseDepartment(String userID, String organizationID) | |
| 前置条件 | 经过验证的总经理有为新员工选择所属机构的权利，且总经理输入了有效的员工编号和机构编号 | |
| 后置条件 | 系统根据输入的员工编号查找对应的员工，并将该员工所属机构修改为机构编号对应的机构 | |
| Manage.addPerWage | 语法 | public int addPerWage(PerWageVO perwagevo) | |
| 前置条件 | 经过验证的总经理有添加新的每次工资的权利，且总经理输入了完整有效的新每次工资信息 | |
| 后置条件 | 系统保存新增的每次工资信息并更新每次工资列表 | |
| Manage.deletePerWage | 语法 | public int deletePerWage(PerWageVO perwagevo) | |
| 前置条件 | 经过验证的总经理有删除每次工资的权利，且总经理输入了有效的待删除每次工资信息 | |
| 后置条件 | 系统查找对应的每次工资信息，并删除该每次工资对象 | |
| Manage.modifyPerWage | 语法 | public int modifyPerWage(PerWageVO perwagevo) | |
| 前置条件 | 经过验证的总经理有修改每次工资的权利，且总经理输入了有效的待修改每次工资信息 | |
| 后置条件 | 系统更新每次工资信息 | |
| Manage.findPerWage | 语法 | public double findPerWage(int profession) | |
| 前置条件 | 经过验证的总经理有查询每次工资的权利，且总经理输入了有效的对应职业编号 | |
| 后置条件 | 系统根据输入的职业编号查找每次工资，并返回每次工资信息 | |
| Manage.addBasicSalary | 语法 | public int addBasicSalary(BasicSalaryVO basicsalaryvo) | |
| 前置条件 | 经过验证的总经理有添加新的基础月薪的权利，且总经理输入了完整有效的新基础月薪信息 | |
| 后置条件 | 系统保存新增的基础月薪信息并更新基础月薪列表 | |
| Manage.deleteBasicSalary | 语法 | public int deleteBasicSalary(BasicSalaryVO basicsalaryvo) | |
| 前置条件 | 经过验证的总经理有删除基础月薪的权利，且总经理输入了有效的待删除基础月薪信息 | |
| 后置条件 | 系统查找对应的基础月薪信息，并删除该基础月薪对象 | |
| Manage.modifyBasicSalary | 语法 | public int modifyBasicSalary(BasicSalaryVO basicsalaryvo) | |
| 前置条件 | 经过验证的总经理有修改基础月薪的权利，且总经理输入了有效的待修改基础月薪信息 | |
| 后置条件 | 系统更新基础月薪信息 | |
| Manage.findBasicSalary | 语法 | public double findBasicSalary(int profession) | |
| 前置条件 | 经过验证的总经理有查询基础月薪的权利，且总经理输入了有效的对应职业编号 | |
| 后置条件 | 系统根据输入的职业编号查找基础月薪，并返回基础月薪信息 | |
| Manage.addCityDistance | 语法 | public int addCityDistance(CityDistanceVO citydistancevo) | |
| 前置条件 | 经过验证的总经理有添加新的城市距离的权利，且总经理输入了完整有效的新城市距离信息 | |
| 后置条件 | 系统保存新增的城市距离信息并更新城市距离列表 | |
| Manage.deleteCityDistance | 语法 | public int deleteCityDistance(CityDistanceVO citydistancevo) | |
| 前置条件 | 经过验证的总经理有删除城市距离的权利，且总经理输入了有效的待删除城市距离信息 | |
| 后置条件 | 系统查找对应的城市距离信息，并删除该城市距离对象 | |
| Manage.modifyCityDistance | 语法 | public int modifyCityDistance(CityDistanceVO citydistancevo) | |
| 前置条件 | 经过验证的总经理有修改城市距离的权利，且总经理输入了有效的待修改城市距离信息 | |
| 后置条件 | 系统更新城市距离信息 | |
| Manage.findCityDistance | 语法 | public double findCityDistance(String cityA, String cityB) | |
| 前置条件 | 经过验证的总经理有查询城市距离的权利，且总经理输入了有效且存在的两个城市 | |
| 后置条件 | 系统根据输入的两个城市名查找城市距离，并返回城市距离信息 | |
| Manage.addCost | 语法 | public int addCost(CostVO costvo) | |
| 前置条件 | 经过验证的总经理有添加新运费系数的权利，且总经理输入了完整有效的新运费系数信息 | |
| 后置条件 | 系统保存新增的运费系数信息并更新运费系数列表 | |
| Manage.deleteCost | 语法 | public int deleteCost(CostVO costvo) | |
| 前置条件 | 经过验证的总经理有删除运费系数的权利，且总经理输入了有效的待删除运费系数信息 | |
| 后置条件 | 系统查找对应的运费系数信息，并删除该运费系数对象 | |
| Manage.modifyCost | 语法 | public int modifyCost(CostVO costvo) | |
| 前置条件 | 经过验证的总经理有修改运费系数的权利，且总经理输入了有效的待修改运费系数信息 | |
| 后置条件 | 系统更新运费系数信息 | |
| Manage.findCost | 语法 | public double findCost(int category) | |
| 前置条件 | 经过验证的总经理有查询运费系数的权利，且总经理输入了有效且存在的类别号 | |
| 后置条件 | 系统根据输入的类别号查找运费系数，并返回运费系数信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| FileFactory.getManageData | | | 得到Manage文件的引用 |
| ManageDataService.addOrganization(OrganizationPO organizationpo) | | | 在管理数据包的机构数据文件中插入一个OrganizationPO对象 |
| ManageDataService.deleteOrganization(OrganizationPO organizationpo) | | | 在管理数据包的机构数据文件中删除一个OrganizationPO对象 |
| ManageDataService.modifyOrganization(OrganizationPO organizationpo) | | | 在管理数据包的机构数据文件中更新一个OrganizationPO对象 |
| ManageDataService.findOrganization(String OrganizationID) | | | 根据输入的机构编号，返回对应的OrganizationPO对象 |
| ManageDataService.modifyUser(UserPO userpo) | | | 在用户数据文件中更新一个UserPO对象 |
| ManageDataService.addPerWage(PerWagePO perwagepo) | | | 在管理数据包的每次工资数据文件中插入一个PerWagePO对象 |
| ManageDataService.deletePerWage(PerWagePO perwagepo) | | | 在管理数据包的每次工资数据文件中删除一个PerWagePO对象 |
| ManageDataService.modifyPerWage(PerWagePO perwagepo) | | | 在管理数据包的每次工资数据文件中更新一个PerWagePO对象 |
| ManageDataService.findPerWage(int profession) | | | 根据输入的职业编号，返回对应的PerWagePO对象 |
| ManageDataService. addBasicSalary(BasicSalaryPO citydistancepo) | | | 在管理数据包的基础月薪数据文件中插入一个BasicSalaryPO对象 |
| ManageDataService.deleteBasicSalary(BasicSalaryPO citydistancepo) | | | 在管理数据包的基础月薪数据文件中删除一个BasicSalaryPO对象 |
| ManageDataService.modifyBasicSalary(BasicSalaryPO citydistancepo) | | | 在管理数据包的基础月薪数据文件中更新一个BasicSalaryPO对象 |
| ManageDataService.findBasicSalary(int profession) | | | 根据输入的职业编号，返回对应的BasicSalaryPO对象 |
| ManageDataService.addCityDistance(CityDistancePO citydistancepo) | | | 在管理数据包的城市距离数据文件中插入一个CityDistancePO对象 |
| ManageDataService.deleteCityDistance(CityDistancePO citydistancepo) | | | 在管理数据包的城市距离数据文件中删除一个CityDistancePO对象 |
| ManageDataService.modifyCityDistance(CityDistancePO citydistancepo) | | | 在管理数据包的城市距离数据文件中更新一个CityDistancePO对象 |
| ManageDataService.findCityDistance(String cityA, String cityB) | | | 根据输入的两个城市名，返回对应的CityDistancePO对象 |
| ManageDataService.addCost(CostPO costpo) | | | 在管理数据包的运费系数数据文件中插入一个CostPO对象 |
| ManageDataService.deleteCost(CostPO costpo) | | | 在管理数据包的运费系数数据文件中删除一个CostPO对象 |
| ManageDataService.modifyCost(CostPO costpo) | | | 在管理数据包的运费系数数据文件中更新一个CostPO对象 |
| ManageDataService. findCost(int category) | | | 根据输入的类别，返回对应的CostPO对象 |

**表9 repertorybl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Repertory.inventoryInitialization | 语法 | public int inventoryInitialization(String repertoryID, int maxRow, int maxShelf, int Maxdigit, int warningRatio) | |
| 前置条件 | 经过验证的仓库管理员有库存信息初始化的权利，且仓库管理员输入了有效的库存信息值 | |
| 后置条件 | 系统保存仓库初始化信息 | |
| Repertory.enterRepertory | 语法 | public int enterRepertory(String repertoryID, String JJD-ID, int blockNum, int rowNum, int shelfNum, int digitNum, String date) | |
| 前置条件 | 经过验证的仓库管理员有将商品入库的权利，且仓库管理员输入了有效的商品信息和对应仓库位置、时间信息 | |
| 后置条件 | 系统保存商品入库的位置和时间信息，生成入库单 | |
| Repertory.leaveRepertory | 语法 | public int leaveRepertory(String repertoryID, String JJD-ID, int transType, String date) | |
| 前置条件 | 经过验证的仓库管理员有将商品出库的权利，且仓库管理员输入了有效的商品信息和转运方式、时间信息 | |
| 后置条件 | 系统保存商品出库的转运方式和时间，生成出库单 | |
| Repertory.inventoryWarning | 语法 | public boolean inventoryWarning() | |
| 前置条件 | 当某区的库存总量与某区最大容量的比值达到或者超过警戒比例时，系统提示仓库管理员该区库存已达警戒比例 | |
| 后置条件 | 无 | |
| Repertory.inventoryCheck | 语法 | public ArrayList<GoodsVO> inventoryCheck(String repertoryID, String beginDate, String endDate) | |
| 前置条件 | 经过验证的仓库管理员有库存查看的权利，且仓库管理员输入了有效的时间段信息 | |
| 后置条件 | 系统根据输入的时间段查找对应的出入库商品，并返回这段时间出/入库商品数量和金额总和，库存数量合计 | |
| Repertory.inventoryStockTaking | 语法 | public ArrayList<GoodsVO> inventoryStockTaking(String repertoryID) | |
| 前置条件 | 经过验证的仓库管理员有库存盘点的权利，且仓库管理员输入了库存盘点命令 | |
| 后置条件 | 系统自动根据提交库存盘点命令的时间生成一个截止点，并根据该截止点查找当天到截止点之前的仍在仓库中的商品，返回商品信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| FileFactory.getRepertoryData | | | 得到Repertory文件的引用 |
| RepertoryDataService.modifyRepertory(RepertoryPO repertorypo) | | | 在仓库数据文件中插入一个RepertoryPO对象 |
| RepertoryDataService.findRepertory(String repertoryID) | | | 在仓库数据文件中删除一个RepertoryPO对象 |
| RepertoryDataService.addGoods(String repertoryID, GoodsPO goodspo) | | | 在编号为repertoryID的仓库对应库存列表中插入一个GoodsPO对象 |
| RepertoryDataService.deleteGoods(String repertoryID, GoodsPO goodspo) | | | 在编号为repertoryID的仓库对应库存列表中删除一个GoodsPO对象 |
| RepertoryDataService.modifyGoods(String repertoryID, GoodsPO goodspo) | | | 在编号为repertoryID的仓库对应库存列表中更新一个GoodsPO对象 |
| RepertoryDataService.findGoodsbyID(String repertoryID, String JJD-ID) | | | 根据输入的仓库编号和订单号，返回对应的GoodsPO对象 |
| RepertoryDataService.findGoodsbyDate(String repertoryID, String beginDate, String endDate) | | | 根据输入的仓库编号和起始日期，返回对应的GoodsPO对象 |
| RepertoryDataService.findGoodsbyTime(String repertoryID, String time) | | | 根据输入的仓库编号和截止点时间，返回对应的GoodsPO对象 |

**表23 数据层模块（UserDataService）的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserDataService.check | 语法 | public boolean check(String userID, String password) |
| 前置条件 | 无 |
| 后置条件 | 返回用户名和密码是否匹配 |
| UserDataService.add | 语法 | public int add(UserPO userpo) |
| 前置条件 | 同样ID的userpo在文件中不存在 |
| 后置条件 | 在外部文件中增加一个po记录 |
| UserDataService.delete | 语法 | public int delete(UserPO userpo) |
| 前置条件 | 在外部文件中存在某一个po需要删除 |
| 后置条件 | 删除一个userpo对象 |
| UserDataService.modify | 语法 | public int modify(UserPO userpo) |
| 前置条件 | 在外部文件中存在某一个po需要更新 |
| 后置条件 | 更新一个userpo对象 |
| UserDataService.find | 语法 | public UserPO find (String ID) |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的UserPO的结果 |
| UserDataService.showAll | 语法 | public ArrayList<UserPO> showAll() |
| 前置条件 | 无 |
| 后置条件 | 返回所有的UserPO的结果 |

**表24数据层模块（ManageDataService）的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ManageDataService.addOrganization | 语法 | public int addOrganization(OrganizationPO organizationpo) |
| 前置条件 | 同样ID的organizationpo在文件中不存在 |
| 后置条件 | 在外部文件中增加一个organizationpo记录 |
| ManageDataService.deleteOrganization | 语法 | public int deleteOrganization(OrganizationPO organizationpo) |
| 前置条件 | 在外部文件中存在某一个organizationpo需要删除 |
| 后置条件 | 删除一个organizationpo对象 |
| ManageDataService.modifyOrganization | 语法 | public int modifyOrganization(OrganizationPO organizationpo) |
| 前置条件 | 在外部文件中存在某一个organizationpo需要更新 |
| 后置条件 | 更新一个organizationpo对象 |
| ManageDataService.findOrganization | 语法 | public OrganizationPO findOrganization(String organizationID) |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的OrganizationPO的结果 |
| ManageDataService.modifyUser | 语法 | public int modifyUser(UserPO userpo) |
| 前置条件 | 在外部文件中存在某一个userpo需要更新 |
| 后置条件 | 更新一个userpo对象 |
| ManageDataService.addPerWage | 语法 | public int addPerWage(PerWagePO perwagepo) |
| 前置条件 | 同样profession的perwagepo在文件中不存在 |
| 后置条件 | 在外部文件中增加一个perwagepo记录 |
| ManageDataService.deletePerWage | 语法 | public int deletePerWage(PerWagePO perwagepo) |
| 前置条件 | 在外部文件中存在某一个perwagepo需要删除 |
| 后置条件 | 删除一个perwagepo对象 |
| ManageDataService.modifyPerWage | 语法 | public int modifyPerWage(PerWagePO perwagepo) |
| 前置条件 | 在外部文件中存在某一个perwagepo需要更新 |
| 后置条件 | 更新一个perwagepo对象 |
| ManageDataService.findPerWage | 语法 | public double findPerWage(int profession) |
| 前置条件 | 无 |
| 后置条件 | 按profession进行查找返回相应的PerWage的值 |
| ManageDataService.addBasicSalary | 语法 | public int addBasicSalary(BasicSalaryPO basicsalarypo) |
| 前置条件 | 同样profession的basicsalarypo在文件中不存在 |
| 后置条件 | 在外部文件中增加一个basicsalarypo记录 |
| ManageDataService.deleteBasicSalary | 语法 | public int deleteBasicSalary(BasicSalaryPO basicsalarypo) |
| 前置条件 | 在外部文件中存在某一个basicsalarypo需要删除 |
| 后置条件 | 删除一个basicsalarypo对象 |
| ManageDataService.modifyBasicSalary | 语法 | public int modifyBasicSalary(BasicSalaryPO basicsalarypo) |
| 前置条件 | 在外部文件中存在某一个basicsalarypo需要更新 |
| 后置条件 | 更新一个basicsalarypo对象 |
| ManageDataService.findBasicSalary | 语法 | public double findBasicSalary((int profession)) |
| 前置条件 | 无 |
| 后置条件 | 按profession进行查找返回相应的BasicSalary的值 |
| ManageDataService.addCityDistance | 语法 | public int addCityDistance(CityDistancePO citydistancepo) |
| 前置条件 | 同样City A, City B的citydistancepo在文件中不存在 |
| 后置条件 | 在外部文件中增加一个citydistancepo记录 |
| ManageDataService.deleteCityDistance | 语法 | public int deleteCityDistance(CityDistancePO citydistancepo) |
| 前置条件 | 在外部文件中存在某一个citydistancepo需要删除 |
| 后置条件 | 删除一个citydistancepo对象 |
| ManageDataService.modifyCityDistance | 语法 | public int modifyCityDistance(CityDistancePO citydistancepo) |
| 前置条件 | 在外部文件中存在某一个citydistancepo需要更新 |
| 后置条件 | 更新一个citydistancepo对象 |
| ManageDataService.findCityDistance | 语法 | public double findCityDistance(String CityA, String CityB) |
| 前置条件 | 无 |
| 后置条件 | 按CityA, CityB进行查找返回相应的CityDistance的值 |
| ManageDataService.addCost | 语法 | public int addCost(CostPO costpo) |
| 前置条件 | 同样category的costpo在文件中不存在 |
| 后置条件 | 在外部文件中增加一个costpo记录 |
| ManageDataService.deleteCost | 语法 | public int deleteCost(CostPO costpo) |
| 前置条件 | 在外部文件中存在某一个costpo需要删除 |
| 后置条件 | 删除一个costpo对象 |
| ManageDataService.modifyCost | 语法 | public int modifyCost(CostPO costpo) |
| 前置条件 | 在外部文件中存在某一个costpo需要更新 |
| 后置条件 | 更新一个costpo对象 |
| ManageDataService.findCost | 语法 | public double findCost(category) |
| 前置条件 | 无 |
| 后置条件 | 按category进行查找返回相应的Cost的值 |

**表24数据层模块（RepertoryDataService）的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| RepertoryDataService. modifyRepertory | 语法 | public int modifyRepertory(RepertoryPO repertorypo) |
| 前置条件 | 需要修改仓库信息 |
| 后置条件 | 更新一个po对象 |
| RepertoryDataService. findRepertory | 语法 | public RepertoryPO findRepertory(String repertoryID) |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的RepertoryPO的结果 |
| RepertoryDataService. addGoods | 语法 | public int addGoods(String repertoryID, GoodsPO goodspo) |
| 前置条件 | 无 |
| 后置条件 | 在仓库的库存信息列表中增加一个po记录 |
| RepertoryDataService. deleteGoods | 语法 | public int deleteGoods(String repertoryID, GoodsPO goodspo) |
| 前置条件 | 在仓库的库存信息列表中存在某一个po需要删除 |
| 后置条件 | 删除一个po对象 |
| RepertoryDataService. modifyGoods | 语法 | public int modifyGoods(String repertoryID, GoodsPO goodspo) |
| 前置条件 | 在仓库的库存信息列表中存在某一个po需要更新 |
| 后置条件 | 更新一个po对象 |
| RepertoryDataService. findGoodsbyID | 语法 | public GoodsPO findGoodsbyID(String repertoryID, String JJD-ID) |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的GoodsPO的结果 |
| RepertoryDataService.findGoodsbyDate | 语法 | public GoodsPO findGoodsbyDate(String repertoryID, String beginDate, String endDate) |
| 前置条件 | 无 |
| 后置条件 | 按beginDate和endDate进行查找返回相应的GoodsPO的结果 |
| RepertoryDataService.findGoodsbyTime | 语法 | public GoodsPO findGoodsbyTime(String repertoryID, String time) |
| 前置条件 | 无 |
| 后置条件 | 按time进行查找返回相应的GoodsPO的结果 |