|  |  |
| --- | --- |
| 测试用例编号 | Test01 |
| 测试项目 | 管理员查看所有优惠券规则 |
| 输入 | 页数page(≥1)且每页数据条数limit(≥1) |
| 操作步骤 | 1.初步判断page和limit是否正确  2.获取所有未删除的优惠券规则  3.计算起始下标floor和结束下标ceil  3.获取对应floor和ceil的优惠券规则列表 |
| 预期输出 | 对应page和limit的优惠券规则列表；  如果请求page的优惠券规则数量不足limit，则输出完优惠券规则为止 |
| 实际输出 | 该页面的优惠券规则列表：couponRulePoList |

|  |  |
| --- | --- |
| 测试用例编号 | Test02 |
| 测试项目 | 管理员查看所有优惠券规则 |
| 输入 | 页数page(≤0)或者每页数据条数limit（≤0） |
| 操作步骤 | 1.初步判断page和limit是否正确  2.获取所有未删除的优惠券规则  3.计算起始下标floor和结束下标ceil  3.获取对应floor和ceil的优惠券规则列表 |
| 预期输出 | “页数page非法”、“每条页数据条数limit非法” |
| 实际输出 | “页数page非法”、“每条页数据条数limit非法” |

|  |  |
| --- | --- |
| 测试用例编号 | Test03 |
| 测试项目 | 管理员根据couponRuleId修改一条优惠券规则内容 |
| 输入 | couponRulePo |
| 操作步骤 | 1.检查couponRuleId是否存在  2.尝试从数据库取出当前id的couponRule  3.根据counponRule和couponRulePo的statusCode判断请求是否合法  4.检测试图修改的属性是否存在且合法  5.设置couponRule必要属性  6.向数据库更新couponRule  7.向前端返回更新后的couponRulePo |
| 预期输出 | 更新后的couponRulePo |
| 实际输出 | 更新后的couponRulePo |

|  |  |
| --- | --- |
| 测试用例编号 | Test04 |
| 测试项目 | 管理员根据couponRuleId修改一条优惠券规则内容 |
| 输入 | couponRulePo，原本的状态statusCode为上架 |
| 操作步骤 | 1.检查couponRuleId是否存在  2.尝试从数据库取出当前id的couponRule  3.根据counponRule和couponRulePo的statusCode判断请求是否合法  4.检测试图修改的属性是否存在且合法  5.设置couponRule必要属性  6.向数据库更新couponRule  7.向前端返回更新后的couponRulePo |
| 预期输出 | 错误信息：“该优惠券规则上架中，不可修改” |
| 实际输出 | 错误信息：“该优惠券规则上架中，不可修改” |

|  |  |
| --- | --- |
| 测试用例编号 | Test05 |
| 测试项目 | 管理员根据couponRuleId修改一条优惠券规则内容 |
| 输入 | couponRulePo，其statusCode和原本的状态statusCode不同 |
| 操作步骤 | 1.检查couponRuleId是否存在  2.尝试从数据库取出当前id的couponRule  3.根据counponRule和couponRulePo的statusCode判断请求是否合法  4.检测试图修改的属性是否存在且合法  5.设置couponRule必要属性  6.向数据库更新couponRule  7.向前端返回更新后的couponRulePo |
| 预期输出 | 错误信息：“不可以对优惠券进行上下架操作” |
| 实际输出 | 错误信息：“不可以对优惠券进行上下架操作” |

|  |  |
| --- | --- |
| 测试用例编号 | Test06 |
| 测试项目 | 管理员根据couponRuleId下架优惠券规则 |
| 输入 | couponRuleId |
| 操作步骤 | 1.查询couponRuleId对应的couponRule是否存在  2.将该优惠券规则的所有优惠券取出并设置为失效  3.将该优惠券规则设置为失效 |
| 预期输出 | responseUtil.ok() |
| 实际输出 | responseUtil.ok() |

|  |  |
| --- | --- |
| 测试用例编号 | Test07 |
| 测试项目 | 管理员新增一条优惠券规则 |
| 输入 | couponRulePo |
| 操作步骤 | 1.将couponRulePo装配到CouponRule类型的实例对象couponRule中  2.检查准备插入的couponRule的属性是否合法  3.装填couponRule的必要属性  4.将couponRule插入数据库  5.返回couponRulePo |
| 预期输出 | responseUtil.ok(couponRulePo) |
| 实际输出 | responseUtil.ok(couponRulePo) |

|  |  |
| --- | --- |
| 测试用例编号 | Test08 |
| 测试项目 | 管理员查看一条优惠券规则 |
| 输入 | couponRuleId |
| 操作步骤 | 1.根据couponRuleId从数据库读取对应coupoRule  2.检查couponRule是否存在  3.将couponRule装填入couponRulePo  4.返回couponRulePo |
| 预期输出 | responseUtil.ok(couponRulePo) |
| 实际输出 | responseUtil.ok(couponRulePo) |

|  |  |
| --- | --- |
| 测试用例编号 | Test09 |
| 测试项目 | 管理员删除一条优惠券规则 |
| 输入 | couponRuleId |
| 操作步骤 | 1.删除id为couponRuleId的优惠券规则 |
| 预期输出 | responseUtil.ok() |
| 实际输出 | responseUtil.ok() |

|  |  |
| --- | --- |
| 测试用例编号 | Test10 |
| 测试项目 | （内部接口）订单模块设置优惠券状态为未使用或已使用 |
| 输入 | 优惠券couponId,操作类型type  （1表示设置为未使用；-1表示设置为已使用） |
| 操作步骤 | 1.根据couponId从数据库取出对应的coupon  2.检查coupon是否存在  3.若type为1，设置优惠券为未使用；若type为-1，设置优惠券为已使用 |
| 预期输出 | responseUtil.ok() |
| 实际输出 | responseUtil.ok() |

|  |  |
| --- | --- |
| 测试用例编号 | Test10 |
| 测试项目 | （内部接口）根据order计算总折扣 |
| 输入 | 订单模块提交的order |
| 操作步骤 | 1.设置order中每个orderItem的类型为普通商品  2.判断order是否使用优惠券，不使用则按普通订单处理  3.使用优惠券，根据order中的couponId从数据库读取coupon，检查是否合法  4.分配优惠券折扣到适用优惠券的orderItem上  5.分配返点折扣到所有orderItem上  6.生成payments  7.返回order |
| 预期输出 | responseUtil.ok(order) |
| 实际输出 | responseUtil.ok(order) |

|  |  |
| --- | --- |
| 测试用例编号 | Test11 |
| 测试项目 | 用户在商场首页查看可领取的优惠券规则 |
| 输入 | 页数page(≥1)且每页数据条数limit(≥1) |
| 操作步骤 | 1.判断page和limit是否合法  2.获取所有没删除的、上架的优惠券规则couponRuleList  3.筛选出在有效期内的优惠券规则resultCouponRuleList  4.返回resultCouponRuleList |
| 预期输出 | responseUtil.ok(resultCouponRuleList) |
| 实际输出 | responseUtil.ok(resultCouponRuleList) |

|  |  |
| --- | --- |
| 测试用例编号 | Test12 |
| 测试项目 | 根据用户userId、优惠券状态showType、页数page和页内数据数limit，返回用户拥有的对应类型的优惠券 |
| 输入 | userId , showType; page; limit |
| 操作步骤 | 1.判断page和limit是否合法  2.获取所有符合条件(userId和showType)的优惠券  3.组装为couponRulePoList并返回 |
| 预期输出 | responseUtil.ok(couponRulePoList) |
| 实际输出 | responseUtil.ok(couponRulePoList) |

|  |  |
| --- | --- |
| 测试用例编号 | Test13 |
| 测试项目 | 根据用户userId和优惠券规则信息领取一张优惠券 |
| 输入 | 用户ID：userId、优惠券信息：couponPo |
| 操作步骤 | 1.根据couponPo中的优惠券规则ID从数据库读取所属的优惠券规则couponRule  2.判断couponRule是否合法  3.检查userId是否已经拥有该规则下的优惠券  4.开始组装新的优惠券coupon  5.向数据库插入新的优惠券  6.返回优惠券couponPo |
| 预期输出 | responseUtil.ok(couponPo) |
| 实际输出 | responseUtil.ok(couponPo) |

|  |  |
| --- | --- |
| 测试用例编号 | Test14 |
| 测试项目 | 根据用户userId和购物车cartItemList列表返回可用的优惠券列表 |
| 输入 | 用户ID：userId、优惠券信息：cartItemList |
| 操作步骤 | 1.从cartItemList中读取goodsIdList列表  2.根据用户userId和购物车goodsIdList列表返回可用的优惠券列表couponList  3.返回优惠券列表couponList |
| 预期输出 | responseUtil.ok(couponList) |
| 实际输出 | responseUtil.ok(couponList) |