**逸社区**

**结算系统接口文档**

作者：况其富

编写日期：2016-12-22

版本号：0.1

版本修订历史

|  |  |  |
| --- | --- | --- |
| 内容 | 修改者 | 版本编号 |
| 初始版本 | 况其富 | 0．1 |

1. **总体说明**

结算系统作为公司比较重要的系统，涉及到各个子系统，需要兼容安全性、开发效率等，结算规则和结算查询这两块内容，业务系统直接引用Dll调用；预结算和结算管理核心业务处理，由结算系统完成；业务系统推送预结算到结算系统，采用TCP通信，具体实现都做了封装，只需要调用方面。

直接调用Dll方法或者调用WCF服务，结算系统Dll引用路径：https://ysqsvn.yun-kai.com/svn/CDYK\_CODE/basics/01\_基础架构/Trunk/lib.service/Settlement；

配置信息（Web.config或App.Config）配置信息

<connectionStrings>

<!--结算数据库连接配置-->

<add name="SettlementConnection" providerName="MySql.Data.MySqlClient" connectionString="server=localhost;port=3306;database=SettlementMaster;uid=root;password=123456" />

</connectionStrings>

<appSettings>

<!--结算服务调用地址-->

<add key="SettlementServiceAddress" value="net.tcp://localhost:8085/SettlementService" />

<add key="SettlementDBPreName" value="Settlement" />

</appSettings>

1. **结算规则管理**
   1. **新增结算规则**

调用方式：Dll调用

类名称：SettlementRuleContext

方法名称：AddSettlementRule

功能描述：新增通用结算规则(物业、停车场、商户都适用)

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 结算规则 | settlementRuleConfig | SettlementRuleConfig | 停车场传递[停车场结算规则](#停车场规则配置)；  物业传递[物业结算规则](#物业费结算配置)；  商城传递[商户结算规则](#商户结算配置)； |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<bool> | 见[方法调用通用结果](#方法调用通用结果)返回 |

调用示例：

SettlementRuleContext settlementRuleContext = new SettlementRuleContext();

var result = settlementRuleContext.AddSettlementRule(new ParkingSettlementRuleConfig());

* 1. **修改结算规则**

调用方式：Dll调用

类名称：SettlementRuleContext

方法名称：UpdateSettlementRule

功能描述：修改通用结算规则 (物业、停车场、商户都适用)

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 结算规则 | settlementRuleConfig | SettlementRuleConfig | 停车场传递[停车场结算规则](#停车场规则配置)；  物业传递[物业结算规则](#物业费结算配置)；  商城传递[商户结算规则](#商户结算配置)； |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<int> | 见[方法调用通用结果](#方法调用通用结果)返回,data为改变后的主键值 |

调用示例：

SettlementRuleContext settlementRuleContext = new SettlementRuleContext();

var result = settlementRuleContext.UpdateSettlementRule (new ParkingSettlementRuleConfig());

* 1. **删除停车场结算规则**

调用方式：Dll调用

类名称：SettlementRuleContext

方法名称：DeleteParkingSettlementRule

功能描述：删除停车场结算规则

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 结算规则Id | settlementRuleId | int |  |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<bool> | 见[方法调用通用结果](#方法调用通用结果)返回 |

调用示例：

SettlementRuleContext settlementRuleContext = new SettlementRuleContext();

var result = settlementRuleContext.DeleteParkingSettlementRule(5);

* 1. **删除物业结算规则**

调用方式：Dll调用

类名称：SettlementRuleContext

方法名称：DeletePropertySettlementRule

功能描述：删除物业结算规则

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 结算规则Id | settlementRuleId | int |  |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<bool> | 见[方法调用通用结果](#方法调用通用结果)返回 |

调用示例：

SettlementRuleContext settlementRuleContext = new SettlementRuleContext();

var result = settlementRuleContext.DeletePropertySettlementRule(5);

* 1. **删除商户结算规则**

调用方式：Dll调用

类名称：SettlementRuleContext

方法名称：DeleteShopSettlementRule

功能描述：删除商户结算规则

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 结算规则Id | settlementRuleId | int |  |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<bool> | 见[方法调用通用结果](#方法调用通用结果)返回 |

调用示例：

SettlementRuleContext settlementRuleContext = new SettlementRuleContext();

var result = settlementRuleContext.DeleteShopSettlementRule(5);

* 1. **新增商品规则**

调用方式：Dll调用

类名称：SettlementRuleContext

方法名称：AddGoodSettlementRule

功能描述：新增商品结算规则

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 结算规则 | goodSettlementRuleConfig | GoodSettlementRuleConfig | 见[商品结算规则配置](#商品结算配置)； |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<bool> | 见[方法调用通用结果](#方法调用通用结果)返回 |

调用示例：

SettlementRuleContext settlementRuleContext = new SettlementRuleContext();

var result = settlementRuleContext.AddGoodSettlementRule(new GoodSettlementRuleConfig());

* 1. **修改商品规则**

调用方式：Dll调用

类名称：SettlementRuleContext

方法名称：UpdateGoodSettlementRule

功能描述：新增商品结算规则

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 结算规则 | goodSettlementRuleConfig | GoodSettlementRuleConfig | 见[商品结算规则配置](#商品结算配置)； |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<int> | 见[方法调用通用结果](#方法调用通用结果)返回,data为改变后的主键值 |

调用示例：

SettlementRuleContext settlementRuleContext = new SettlementRuleContext();

var result = settlementRuleContext.UpdateGoodSettlementRule(new GoodSettlementRuleConfig());

* 1. **删除商户结算规则**

调用方式：Dll调用

类名称：SettlementRuleContext

方法名称：DeleteGoodSettlementRule

功能描述：删除商品结算规则

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 结算规则Id | goodSettlementRuleId | int |  |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<bool> | 见[方法调用通用结果](#方法调用通用结果)返回 |

调用示例：

SettlementRuleContext settlementRuleContext = new SettlementRuleContext();

var result = settlementRuleContext.DeleteGoodSettlementRule(5);

* 1. **根据商品Id获取商品结算配置集合**

调用方式：Dll调用

类名称：SettlementRuleContext

方法名称：GetUsedGoodSettlementRuleByCommodityId

功能描述：根据商品Id获取商品结算配置集合

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 商品Id | commodityId | int |  |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| IEnumerable<GoodSettlementRuleConfig> | 见[商品规则信息](#商品结算配置)描述 |

调用示例：

SettlementRuleContext settlementRuleContext = new SettlementRuleContext();

var result = settlementRuleContext.GetUsedGoodSettlementRuleByCommodityId(1);

1. **推送预结算信息**

都是以调用Wcf服务，WCF调用细节已经做了封装，只需要调用封装类的方法

* 1. **创建物业预结算**

调用方式：WCF调用

类名称：BusinessCallClient

方法名称：CreatePropertyPreSettlement

功能描述：创建物业预结算

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 预结算源 | prePropertySettlementMaterial | PrePropertySettlementMaterial |  |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<bool> | 见[方法调用通用结果](#方法调用通用结果)返回 |

调用示例：

BusinessCallClient preSettlementClient= new BusinessCallClient();

var result = preSettlementClient.CreatePropertyPreSettlement(new PrePropertySettlementMaterial);

* 1. **创建停车场预结算**

调用方式：WCF调用

类名称：BusinessCallClient

方法名称：CreateParkingPreSettlement

功能描述：创建物业预结算

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 预结算源 | preParkingSettlementMaterial | PreParkingSettlementMaterial |  |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<bool> | 见[方法调用通用结果](#方法调用通用结果)返回 |

调用示例：

BusinessCallClient preSettlementClient= new BusinessCallClient();

var result = preSettlementClient.CreateParkingPreSettlement(new PreParkingSettlementMaterial);

* 1. **创建商城预结算**

调用方式：WCF调用

类名称：BusinessCallClient

方法名称：CreateShopPreSettlement

功能描述：创建商城预结算

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数中文名称 | 参数名称 | 参数类型 | 说明 |
| 预结算源 | preShopSettlementMaterial | PreShopSettlementMaterial |  |

返回参数

|  |  |
| --- | --- |
| 参数类型 | 说明 |
| SettlementCallResult<bool> | 见[方法调用通用结果](#方法调用通用结果)返回 |

调用示例：

BusinessCallClient preSettlementClient= new BusinessCallClient();

var result = preSettlementClient.CreateShopPreSettlement(new PreShopSettlementMaterial);

**附录**

**方法调用通用结果**

[DataContract]

/// <summary>

/// 结算客户端调用结果

/// </summary>

public class SettlementCallResult<T>

{

[DataMember]

/// <summary>

/// 方法调用是否成功

/// </summary>

public bool IsSuccess { get; set; }

[DataMember]

/// <summary>

/// 方法调用返回值

/// </summary>

public string Message { get; set; }

[DataMember]

/// <summary>

/// 方法调用返回值

/// </summary>

public T Data { get; set; }

}

**结算配置基类**

/// <summary>

/// 结算配置基类

/// </summary>

public class SettlementRuleConfig

{

/// <summary>

/// 主键

/// </summary>

public int Id { get; set; }

/// <summary>

/// 是否参与结算

/// </summary>

public bool IsSettlement { get; set; }

/// <summary>

/// 是否配置完整

/// </summary>

public bool IsFullConfig { get; set; }

/// <summary>

/// 结算账户类型

/// </summary>

public ESettlementAccountType SettlementAccountType { get; set; }

/// <summary>

/// 结算账户(400)

/// </summary>

public string SettlementAccount { get; set; }

/// <summary>

/// 结算账户名称(400)

/// </summary>

public string SettlementAccountName { get; set; }

/// <summary>

/// 是否自动结算

/// </summary>

public bool IsAutoSettlement { get; set; }

/// <summary>

/// 结算周期

/// </summary>

public int SettlementCycle { get; set; }

/// <summary>

/// 手续费承担方

/// </summary>

public EHandlingChargeOwner HandlingChargeOwner { get; set; }

/// <summary>

/// 手续费率(10,4)

/// </summary>

public decimal HandlingChargeRate { get; set; }

/// <summary>

/// 是否启用

/// </summary>

public bool IsUsed { get; set; }

}

**停车场规则配置**

/// <summary>

/// 停车收费结算配置

/// </summary>

public class ParkingSettlementRuleConfig : SettlementRuleConfig

{

/// <summary>

/// 停车场Id

/// </summary>

public Guid ParkingId { get; set; }

/// <summary>

/// 结算比例(10,4)

/// </summary>

public decimal SettlementScale { get; set; }

}

**物业费结算配置**

/// <summary>

/// 物业费结算配置

/// </summary>

public class PropertySettlementRuleConfig : SettlementRuleConfig

{

/// <summary>

/// 小区Id

/// </summary>

public int CommunityDeptId { get; set; }

/// <summary>

/// 结算比例(10,4)

/// </summary>

public decimal SettlementScale { get; set; }

/// <summary>

/// 物业费付款方式(30)

/// </summary>

public string PropertyPayType { get; set; }

}

**商户结算配置**

/// <summary>

/// 商户结算配置

/// </summary>

public class ShopSettlementRuleConfig : SettlementRuleConfig

{

/// <summary>

/// 商户Id

/// </summary>

public int MerchantId { get; set; }

/// <summary>

/// 扫码结算比例

/// </summary>

public decimal ScanCodePayScale { get; set; }

}

**商品结算配置**

/// <summary>

/// 商品结算配置

/// </summary>

public class GoodSettlementRuleConfig

{

/// <summary>

/// 主键

/// </summary>

public int Id { get; set; }

/// <summary>

/// 商品Id

/// </summary>

public int CommodityId { get; set; }

/// <summary>

/// 商户结算价格(10,2)

/// </summary>

public decimal MerchantPrice { get; set; }

/// <summary>

/// 物业结算价格(10,2)

/// </summary>

public decimal PropertyPrice { get; set; }

/// <summary>

/// 小区Code集合(max)

/// </summary>

public string CommunityCodes { get; set; }

/// <summary>

/// 是否启用

/// </summary>

public bool IsUsed { get; set; }

}

**预结算源**

[DataContract]

/// <summary>

/// 预结算源

/// </summary>

public class PreSettlementMaterial

{

[DataMember]

/// <summary>

/// 物业和停车场为缴费单号，商城为订单号

/// </summary>

public string OriginId { get; set; }

[DataMember]

/// <summary>

/// 费用的二级分类(费用可能为多张表)

/// </summary>

public string ChargeType { get; set; }

[DataMember]

/// <summary>

/// 发生金额（10,2）

/// </summary>

public decimal OccurMoney { get; set; }

[DataMember]

/// <summary>

/// 物业和停车场传递小区DeptID，商城传递CommunityCode

/// </summary>

public string CommunityIdentity { get; set; }

[DataMember]

/// <summary>

/// 业务表单生成时间

/// </summary>

public DateTime BusinessTime { get; set; }

}

**物业预结算源**

[DataContract]

/// <summary>

/// 物业预结算源

/// </summary>

public class PrePropertySettlementMaterial: PreSettlementMaterial

{

}

**停车场预结算源**

[DataContract]

/// <summary>

/// 停车场预结算源

/// </summary>

public class PreParkingSettlementMaterial : PreSettlementMaterial

{

[DataMember]

/// <summary>

/// -停车场Id

/// </summary>

public Guid ParkingId { get; set; }

[DataMember]

/// <summary>

/// -停车场名称(200)

/// </summary>

public string ParkingName { get; set; }

}

**商城预结算源**

[DataContract]

/// <summary>

/// 商城预结算源

/// </summary>

public class PreShopSettlementMaterial : PreSettlementMaterial

{

[DataMember]

/// <summary>

/// 商城支付类型

/// </summary>

public EOrderPayType OrderPayType { get; set; }

[DataMember]

/// <summary>

/// 商户Id

/// </summary>

public int MerchantId { get; set; }

[DataMember]

/// <summary>

/// 商户名称(100)

/// </summary>

public string MerchantName { get; set; }

[DataMember]

/// <summary>

/// 商品Id集合(max)

/// </summary>

public string CommodityIds { get; set; }

[DataMember]

/// <summary>

/// 商品价格集合(max)

/// </summary>

public string CommodityPrices { get; set; }

[DataMember]

/// <summary>

/// 订单金额

/// </summary>

public decimal OrderMoney { get; set; }

}