征信接口报文处理设计

# 设计目的

由于征信中心数据传输均采用txt报文，需要采用一种较为灵活的方式对报文的解析及生成进行控制。

# 设计思路

为了能够灵活的将txt报文和系统业务实体匹配，需要有一个配置文件对报文的结构和实体映射关系进行维护管理。

应用程序根据配置文件定义的结构使用反射技术将实体和txt报文之间互相转换。

# 详细设计

## 配置文件设计

配置文件样例：

|  |
| --- |
| <ZxMsgFormats>  <ZxFormat name=*"NameNormalMessage"* id=*"com.hengtian.zxjk.entity.message.NameNormalMessage"*>  <ZxNode propertyName=*"FinancialOrganizationCode"* length=*"14"* type=*"AN"* state=*"M"*/>  <ZxNode propertyName=*"MessageRiseTime"* length=*"6"* type=*"AN"* state=*"M"*/>  <ZxNode propertyName=*"FlowNum1"* length=*"3"* type=*"AN"* state=*"M"*/>  <ZxNode propertyName=*"ReportType"* length=*"1"* type=*"AN"* state=*"M"*/>  <ZxNode propertyName=*"FlowNum2"* length=*"3"* type=*"AN"* state=*"M"*/>  </ZxFormat>  </ZxMsgFormats> |

### Element介绍

* **ZxFormat**：每种业务报文对应的配置信息。
* **ZxNode**：具体业务字段。

### Attribute介绍

* **id**：报文对应的业务实体类型名。
* **propertyName**：报文中对应的业务实体属性字段名称。
* **length**: 报文中属性的长度
* **type**：报文中属性的数据类型
* **state**：报文中属性的状态，M表示必选，O表示可选

## 业务实体设计

为了处理简便快速，报文业务实体采用平面型的设计方式。

业务实体中的字段类型均为String，报文中定义的各种字段类型在业务实体后进行处理。

对应3.1节中的样例，业务实体定义如下：

|  |
| --- |
| **package** com.hengtian.zxjk.entity.message;  **public** **class** NameNormalMessage {  /\*\*  \* 金融机构代码(第1～14位)  \*/  **private** String financialOrganizationCode;  /\*\*  \* 第15～20位：表示数据发生年月；  \*/  **private** String messageRiseTime;  /\*\*  \* 第21～23位：表示报文流水序号，由0-9的数字和大写字母A-Z组成  \*/  **private** String flowNum1;  /\*\*  \* 报文类别(第24位)  \*/  **private** String reportType;  /\*\*  \* 流水号(第25～27位),表示报文流水序号,由0-9的数字和大写字母A-Z组成  \*/  **private** String flowNum2;    **public** String getFinancialOrganizationCode() {  **return** financialOrganizationCode;  }  **public** **void** setFinancialOrganizationCode(String financialOrganizationCode) {  **this**.financialOrganizationCode = financialOrganizationCode;  }  **public** String getMessageRiseTime() {  **return** messageRiseTime;  }  **public** **void** setMessageRiseTime(String messageRiseTime) {  **this**.messageRiseTime = messageRiseTime;  }  **public** String getFlowNum1() {  **return** flowNum1;  }  **public** **void** setFlowNum1(String flowNum1) {  **this**.flowNum1 = flowNum1;  }  **public** String getReportType() {  **return** reportType;  }  **public** **void** setReportType(String reportType) {  **this**.reportType = reportType;  }  **public** String getFlowNum2() {  **return** flowNum2;  }  **public** **void** setFlowNum2(String flowNum2) {  **this**.flowNum2 = flowNum2;  }  } |

## 编码解码设计

报文的编解码的产物分别为txt报文和系统业务实体。

编解码都是通过读取配置，使用反射技术来完成的。

### 编码

通过反射获取业务实体对应配置信息

对配置信息进行深度优先遍历

根据各个配置项生成txt报文

获取实体类属性的get方法

获取各个业务属性的值

### 解码

通过反射获取业务实体对应配置信息

对配置信息进行深度优先遍历

使用set方法设置实体中的属性值

根据配置获取txt报文中的String[]

获取实体类属性的set方法

## 异常处理设计

异常处理的目的是帮助开发人员定位报文编解码中的具体错误。

自定义异常异常：

* 待定