

本科毕业论文（设计）

概要设计说明书

|  |  |
| --- | --- |
| **学 生 姓 名** | **吴光宇** |
| **学号** | **2014051016** |
| **专业** | **计算机科学与技术** |
| **年级班级** | **2014级计算机（应用）1班** |
| **指导教师** | **张欢（讲师）** |
| **所在学院** | **计算机学院** |
| **提交日期** | **2018年5月16日** |

2018 年 5 月

成都信息工程大学 计算机学院

目录

[1 引言 - 1 -](#_Toc514868358)

[1.1 编写目的 - 1 -](#_Toc514868359)

[1.2 背景 - 1 -](#_Toc514868360)

[1.3 参考资料 - 1 -](#_Toc514868361)

[2 总体设计 - 2 -](#_Toc514868362)

[2.1 系统体系结构 - 2 -](#_Toc514868363)

[2.2 系统功能结构 - 2 -](#_Toc514868364)

[2.3 运行环境 - 3 -](#_Toc514868365)

[2.3.1 硬件环境 - 3 -](#_Toc514868366)

[2.3.2 软件环境 - 3 -](#_Toc514868367)

[2.4 系统的关键技术 - 4 -](#_Toc514868368)

[2.4.1 Spring - 4 -](#_Toc514868369)

[2.4.2 Spring Security - 4 -](#_Toc514868370)

[2.4.3 Hbase - 5 -](#_Toc514868371)

[3 功能模块设计说明 - 5 -](#_Toc514868372)

[3.1 功能模块列表 - 5 -](#_Toc514868373)

[3.2 数据导入 - 6 -](#_Toc514868374)

[3.2.1 模块编号和功能描述 - 6 -](#_Toc514868375)

[3.2.2 操作者 - 6 -](#_Toc514868376)

[3.2.3 与本模块相关的码表和表 - 6 -](#_Toc514868377)

[3.2.4 界面设计与说明 - 7 -](#_Toc514868378)

[3.2.5 输入信息 - 7 -](#_Toc514868379)

[3.2.6 输出信息 - 7 -](#_Toc514868380)

[3.2.7 处理流程 - 7 -](#_Toc514868381)

[3.2.8 类设计 - 8 -](#_Toc514868382)

[3.3 数据可视化 - 10 -](#_Toc514868383)

[3.3.1 模块编号和功能描述 - 10 -](#_Toc514868384)

[3.3.2 操作者 - 10 -](#_Toc514868385)

[3.3.3 与本模块相关的码表和表 - 10 -](#_Toc514868386)

[3.3.4 界面设计与说明 - 11 -](#_Toc514868387)

[3.3.5 输入信息 - 12 -](#_Toc514868388)

[3.3.6 输出信息 - 12 -](#_Toc514868389)

[3.3.7 处理流程 - 12 -](#_Toc514868390)

[3.3.8 类设计 - 13 -](#_Toc514868391)

[4 内部接口设计 - 14 -](#_Toc514868392)

[4.1 Format - 14 -](#_Toc514868393)

# 引言

## 编写目的

通过之前的需求分析，得出了系统得基本需求，以及基本的数据处理流程，要实现整个系统，还需要根据已得出的需求，进行设计。概要设计主要是利用比较抽象的语言对整个需求进行概况，确定对系统的物理配置，确定整个系统的处理流程和系统的数据结构，接口设计，实现对系统的初步设计。预期读者为系统开发人员。

## 背景

1. 系统名称：基于Hadoop的自适应光学分析
2. 提出者、开发者：吴光宇
3. 系统应用范围：实验数据存储与可视化

## 参考资料

1. 薄奇,许林英.Spring框架中IoC的实现[J].微处理机,2008年,第1期
2. 百度百科：面向切面编程[EB/OL]. https://baike.baidu.com/item/面向切面编程
3. 张国平,万仲保,刘高原.Spring Aop框架在J2EE中的应用[J].微计算机信息,2007,第23卷，第12-3期
4. Craig Walls. Manning.Spring.in.Action[M].4th.Edition.北京市丰台区成寿寺路11号：人民邮电出版社。
5. 蒋丛萃,史卓.Spring Security在电子商务中的设计应用[J].电脑知识与技术,2016，第5x期
6. 维基百科：Apache Hbase[EB/OL]. <https://zh.wikipedia.org/wiki/Apache_HBase>
7. 张叶,许国艳,花青.基于HBase的矢量空间数据存储与访问优化[J].计算机应用,2015,35(11):3102-3105
8. Chao-Hsien Lee, Yu-Lin Zheng, ‘ Automatic SQL-to-NoSQL Schema Transformation over the MySQL and HBase Databases’, 2015International Conference on Consumer Electronics

# 总体设计

## 系统体系结构

|  |
| --- |
| F:\迅雷下载\组织结构.png |
| 图2‑1 系统体系结构图 |

## 系统功能结构

|  |
| --- |
|  |

图2‑2 系统功能结构图

## 运行环境

### 硬件环境

1）服务器：

内存：16G或更多

处理器：E3-1230 V3

硬盘：1T

2）客户端：

处理器：Intel Pentium或以上

内存：2G以上

### 软件环境

1) 操作系统：Windows2000/XP或更新版本、Centos6.5或更新版本

2) 数据库系统：HDP2.6或更新版本

3) 开发平台及工具：Intellij Idea

4) 应用服务器：Tomcat7或更高

5) 其他软件：Centos7、Chorme

## 系统的关键技术

### Spring

支撑spring的两大核心原理，一是控制反转(依赖注入)，在传统的java程序中，系统模块之间必然存在这各种各样的依赖关系，过强的耦合会使软件结构复杂、维护困难[1]。而spring的控制反转(依赖注入)则可以解决这个问题。另一个是面向切面编程，面向切面编程主要实现的目的是针对业务处理过程中的切面进行提取，它所面对的是处理过程中的某个步骤或阶段，以获得逻辑过程中各部分之间低耦合性的隔离效果[2]。这些切面，也叫关注点，一个关注点，就是一个特定的目的，或者说一个系统需要实现的目标[3]，而这个关注点可能横跨几个模块。

在spring中实现控制反转(依赖注入)的方式是jvm级别维护一个Ioc容器，该Ioc容器通过配置文件来生成、管理、销毁类实例，而在程序中需要获取类实例都在Ioc容器中进行获取，即组件之间的依赖关系由容器在运行期间决定，形象的说，就是就是容器在运行期间将对象实例注入到组件中，而组件只需要关心自己的业务逻辑，而不需要关心使用的资源来自何方。

而面向切面编程主要解决代码重复和关注点分散的问题，面向切面编程比较常见的例子就是系统的日志接口，几乎每个类都需要使用日志接口，如果将日志功能单独抽象为一个模块，然后再每个类中都对其进行实例化，这样会造成所有模块都和日志模块耦合，而如果将日志功能的代码嵌入到每个需要使用的地方则会造成该段代码重复率极高，而对于原本的类来说，日志功能的实现却不是该模块应该关注的内容，为了解决这些问题，spring提供了面向切面编程的功能，通过声明的方式将功能模块应用到需要他们的模块中去，进而改变他们的行为。

Spring是基于配置来实例化对象、应用对象、销毁对象，随着项目规模的不断扩大，spring需要的配置文件越来越繁杂，以至于维护配置信息变得极其困难，spring boot的出现就是为了解决这个问题，spring boot默认习惯优于配置，即spring默认配置好了符合大多数项目的配置文件，而使用者无需关心，直接使用即可，而调整默认配置也变得极其简单。

### Spring Security

Spring security是为基于spring的应用程序提供声明式安全保护的安全性框架[4]。

认证和权限控制是Spring Security要处理的两个主要领域[5]。认证需要处理这样一些问题：

1.用户具有哪些权限

2.访问对应资源需要什么权限

3.用户是否具有访问该资源的权限

授权需要解决这些问题：

1.该系统权限种类

2.如何授予用户权限

Spring security支持广泛的认证模型，其中大部分既支持标准机构发布的模型，也支持第三方发布的模型。

Spring security的权限认证主要集中在以下三个方面：

1.WEB请求的认证

2.方法执行的认证

3.独立域资源的授权

### Hbase

Hbase是一个开源的非关系型分布式数据库（NoSQL），它参考了google的bigTable[6]建模，实现的语言为java[7]。

Hbase适合于存储大数据，支持单条记录的快速查询，在任意指定位置单条或批量添加，删除数据[8]。Hbase将数据作为字节块存储在分布式文件系统中，如果列值为空，将不会被存储，这将大大节省存储空间。而行键按LSM结构进行存储，进而支持单条记录的快速查询。

分布式的非SQL数据库提供了人们十分需要的可伸缩特性。这使得横向扩展变得容易，从而能够存储更多的数据。

# 功能模块设计说明

## 功能模块列表

表3‑1 功能模块列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块**  **编号** | **模块**  **名称** | **对应需求**  **功能编号** | **所对应**  **需求功能** | **实现**  **优先级** |
| DS\_YXGL01 | 数据导入 | SRS\_SJGL01 | 数据导入 | 高 |
| DS\_YXGL02 | 数据存储 | SRS\_SJGL02 | 数据存储 | 高 |
| DS\_YXGL03 | 数据统计 | SRS\_SJGL03 | 数据统计 | 高 |
| DS\_YXGL04 | 数据可视化 | SRS\_SJGL04 | 数据可视化 | 高 |
| DS\_YXGL05 | 角色管理 | SRS\_SJGL05 | 角色管理 | 中 |
| DS\_YXGL06 | 用户管理 | SRS\_SJGL06 | 用户管理 | 高 |
| DS\_YXGL07 | 系统日志管理 | SRS\_SJGL07 | 系统日志管理 | 低 |
| DS\_YXGL08 | 操作日志管理 | SRS\_SJGL08 | 操作日志管理 | 低 |

## 数据导入

### 模块编号和功能描述

编号：DS\_YXGL01

主要功能：通过前端预留的接口，用户可以上传本地的文件到服务器端。

### 操作者

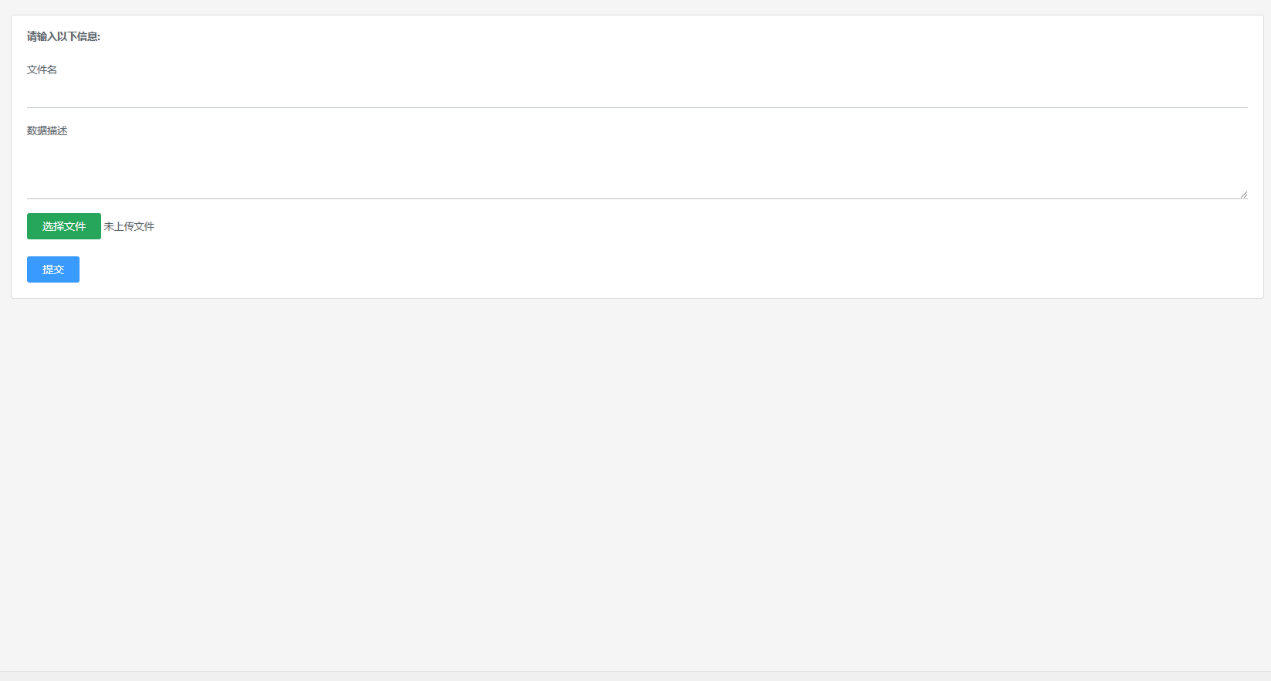
系统所有角色。

### 与本模块相关的码表和表

表3‑2 模块功能表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **中文注释** | **类型** | | **作用** |
| **码表** | **表** |
| graduation\_data | 通用数据表 |  | 是 | 记录导入数据的相关信息 |
| graduation\_data\_all | 详细数据表 |  | 是 | 记录导入文件的详细信息 |

### 界面设计与说明

界面应该如下图所示，包含文件名、数据描述的输入框和文件上传的按钮。

### 输入信息

文件名：String

数据描述：String

Ocam-gv格式文件，文件一帧数据长度为2048个字节，每帧前1024字节为斜率数据，后1024个字节为电压数据，每两个字节为一个有效数据。

### 输出信息

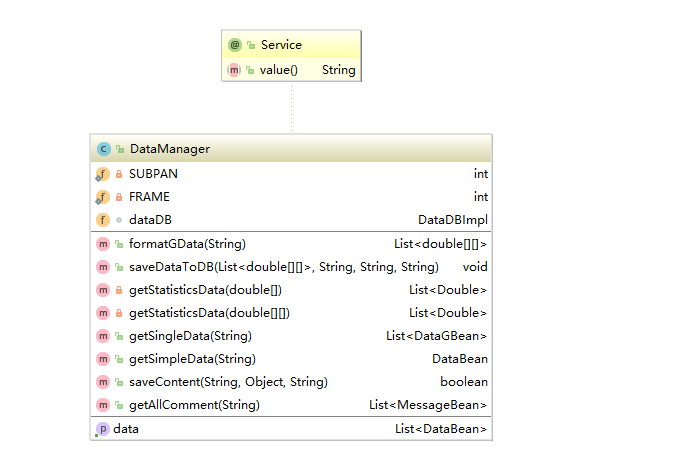
文件上传状态

### 处理流程

### 类设计

#### 类图

|  |
| --- |
|  |
|  |

图3‑1 类图

#### 类说明

（一）DataController类说明：

1. 功能：负责响应前端请求
2. 主要方法：

upload()：接受前端文件上传请求

（二）DataController类说明：

1. 功能：数据导入逻辑处理
2. 主要方法：up

format()：解析数据

saveToDB()：将数据存入数据库

getStatistics()：分析数据的统计信息

## 数据可视化

### 模块编号和功能描述

编号：DS\_YXGL04

主要功能：实现数据可视化。

### 操作者

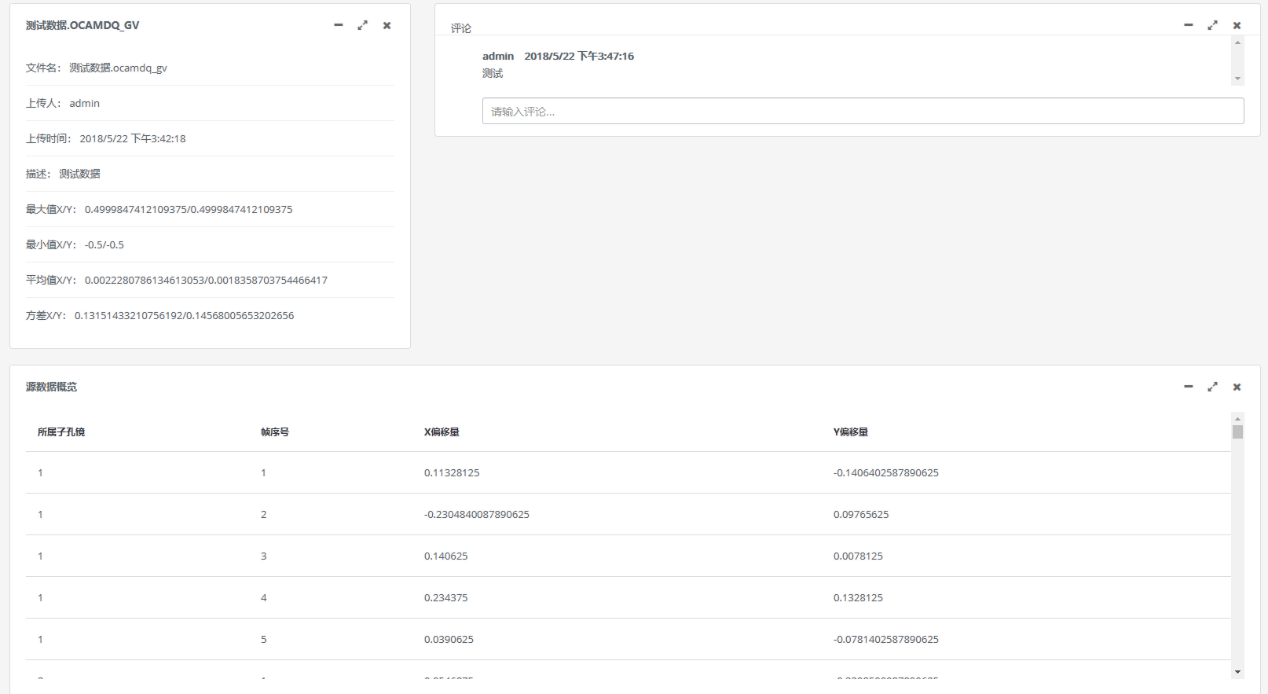
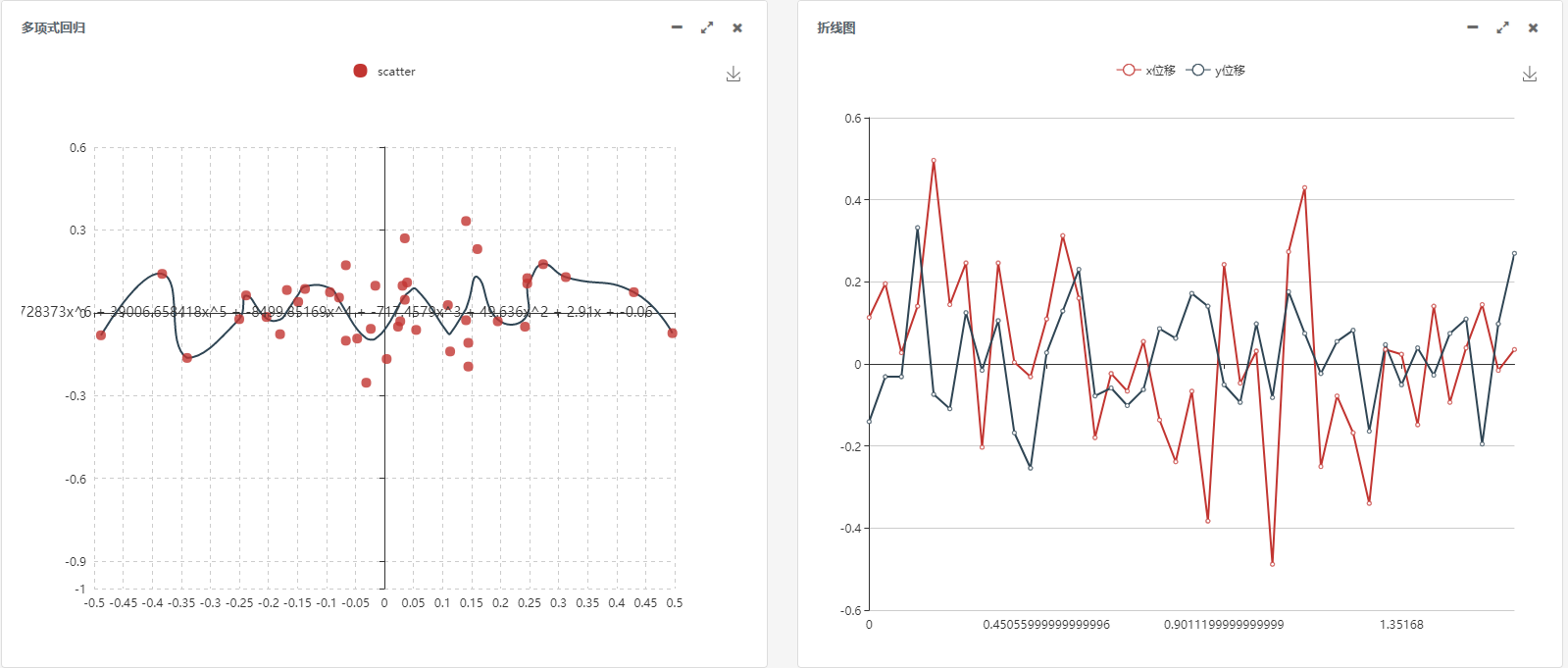
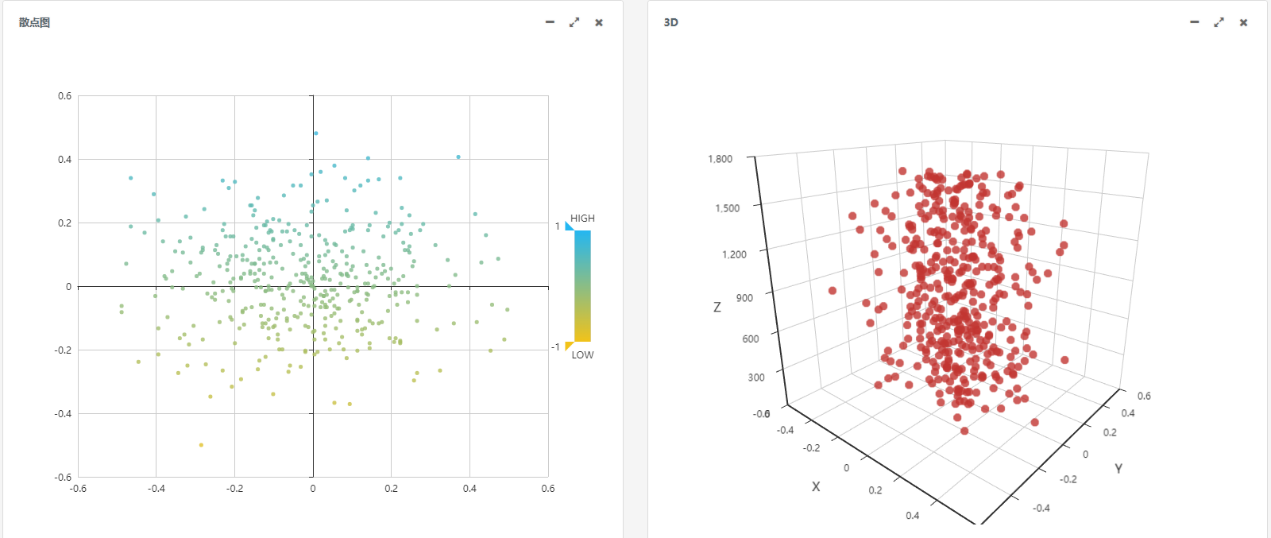
系统所有角色。

### 与本模块相关的码表和表

表3‑2 模块功能表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **中文注释** | **类型** | | **作用** |
| **码表** | **表** |
| graduation\_data | 通用数据表 |  | 是 | 记录导入数据的相关信息 |
| graduation\_data\_all | 详细数据表 |  | 是 | 记录导入文件的详细信息 |
| graduation\_comment | 评论表 |  | 是 | 记录评论 |

### 界面设计与说明

界面如下所示：

### 输入信息

获取详细信息的请求。

### 输出信息

数据可视化视图

### F:\迅雷下载\流程图2.png处理流程

### 类设计

#### 类图

图3‑2 类图

#### 类说明

（一）DataController类说明：

1. 功能：负责响应前端请求
2. 主要方法：

getData()：获取详细数据

getSingleData()：获取统计数据

getAllComment()：获取所有评论

（二）DataController类说明：

1. 功能：数据导入逻辑处理
2. 主要方法：up

getStatistics()：获取统计数据

saveToDB()：将数据存入数据库

getStatistics()：分析数据的统计信息

# 内部接口设计

表4‑1 构件接口列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块名称** | **接口编号** | **接口名称** | **接口类型** | **说明** |
| 数据处理 | 1 | Format | 内部 | 解析数据 |
| 2 | SaveToDB | 内部 | 将数据存储进数据库 |

## Format

1. 接口属性设计

表4‑2 format接口说明

|  |  |
| --- | --- |
| **接口编号** | 1 |
| **接口名称** | Format |
| **接口说明** | 解析数据 |
| **数据来源** | 文件 |
| **调用者** | Datacontroller |
| **输入** | 文件路径 |
| **输出** | 解析结果 |
| **处理流程** | 获取数据、解析数据、保存数据、返回状态码 |

1. 类设计

表4‑3 Datamanager类

|  |  |
| --- | --- |
| **类名称** | Datamanager |
| **分类** | Logic |
| **描述** | 处理数据相关逻辑 |
| **使用到的其他类** | DataDBImpl |