**estimation.service 模块测试**

测试方法：

* 保证所有的语句、分支被覆盖
* 参考等价类划分方法
* 参考边界值分析方法
* 用例分析与设计
* 测试脚本使用Java语言实现，并且和驱动（Drive）代码、桩（Stub）代码构成一个可执行系统
* 覆盖率分析使用IDEA工具
* 某函数的缺陷被修正后必须回归与该函数相关的所有单元测试用例

1. **ApiController 类测试函数（方法）参考表**

|  |  |  |
| --- | --- | --- |
| 标识符 | 名称 | 代码行（LOC） |
| BACK\_001\_API\_001 | List<Sensor> getSeneors(Integer groupId) |  |
| BACK\_001\_API\_002 | Map<String,List<Sensor>> getSensorDividedByType(Long groupId) |  |
| BACK\_001\_API\_003 | List<SensorGroup> addNewGroup(@RequestBody SensorGroup group) |  |
| BACK\_001\_API\_004 | List<SensorGroup> getAllGroup() |  |

getSeneors测试分析与设计

1）标识符定义：BACK\_001\_API\_001

2）被测特性：

* groupId在数据库中不存在时返回空列表
* groupId在数据库中存在返回一个List<Sensor>,执行成功

3）测试方法：

* 考虑groupId在数据库中是否存在，即是否返回空列表，分为返回空列表和返回List<Sensor>

4）测试项标志

|  |  |  |
| --- | --- | --- |
| 测试项目标识 | 测试项描述 | 优先级 |
| BACK\_001\_API\_001\_01 | getSeneors返回空列表 | 高 |
| BACK\_001\_API\_001\_02 | getSeneors返回类型为List<Sensor> | 高 |

5）测试通过\失败标准

所有的用例都必须被执行，且没有发现错误

6）对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_API\_001\_01 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | getSeneors返回空列表 | | | | | |
| 前置条件 | 数据库中所有groupId为1——99 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 边界值 | groupId= “100” | null | null | 无 | / |
| 002 | 边界值 | groupId= “0” | null | null | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_API\_001\_02 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | getSeneors返回类型为List<Sensor> | | | | | |
| 前置条件 | 数据库中所有groupId为1——99 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 边界值 | groupId= “50” | 返回groupID=”50”对应数据的List<Sensor> | 返回groupID=”50”对应数据的List<Sensor> | 无 | / |
| 002 | 边界值 | groupId= “1” | 返回groupID=”1”对应数据的List<Sensor> | 返回groupID=”1”对应数据的List<Sensor> | 无 | / |
| 003 | 边界值 | groupId= “99” | 返回groupID=”99”对应数据的List<Sensor> | 返回groupID=”99”对应数据的List<Sensor> | 无 | / |
| 004 | 边界值 | groupId= “2” | 返回groupID=”2”对应数据的List<Sensor> | 返回groupID=”2”对应数据的List<Sensor> | 无 | / |
| 005 | 边界值 | groupId= “98” | 返回groupID=”98”对应数据的List<Sensor> | 返回groupID=”98”对应数据的List<Sensor> | 无 | / |

getSeneorsDivideByType测试分析与设计

1）标识符定义：BACK\_001\_API\_002

2）被测特性：

* groupId在数据库中不存在时返回空哈希表
* groupId在数据库中存在返回一个Map<String,List<Sensor>>,执行成功

3）测试方法：

* 考虑groupId在数据库中是否存在，即是否返回空哈希表，分为返回空哈希表和返回Map<String,List<Sensor>>

4）测试项标志

|  |  |  |
| --- | --- | --- |
| 测试项目标识 | 测试项描述 | 优先级 |
| BACK\_001\_API\_002\_01 | getSensorDividedByType返回空哈希表 | 高 |
| BACK\_001\_API\_002\_02 | getSensorDividedByType返回类型为Map<String,List<Sensor>> | 高 |

5）测试通过\失败标准

所有的用例都必须被执行，且没有发现错误

6）对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_API\_002\_01 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | getSensorDividedByType返回空哈希表 | | | | | |
| 前置条件 | 数据库中所有groupId为1——99 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 边界值 | groupId= “0” | null | null | 无 | / |
| 002 | 边界值 | groupId= “100” | null | null | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_API\_002\_02 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | getSensorDividedByType返回类型为Map<String,List<Sensor>> | | | | | |
| 前置条件 | 数据库中所有groupId为1——99 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 边界值 | groupId= “1” | 返回groupID=”1”对应数据的List<Sensor> | 返回groupID=”1”对应数据的List<Sensor> | 无 | / |
| 002 | 边界值 | groupId= “2” | 返回groupID=”2”对应数据的List<Sensor> | 返回groupID=”2”对应数据的List<Sensor> | 无 | / |
| 003 | 边界值 | groupId= “50” | 返回groupID=”50”对应数据的List<Sensor> | 返回groupID=”50”对应数据的List<Sensor> | 无 | / |
| 004 | 边界值 | groupId= “99” | 返回groupID=”99”对应数据的List<Sensor> | 返回groupID=”99”对应数据的List<Sensor> | 无 | / |
| 005 | 边界值 | groupId= “100” | 返回groupID=”100”对应数据的List<Sensor> | 返回groupID=”100”对应数据的List<Sensor> | 无 | / |

addNewGroup测试分析与设计

1）标识符定义：BACK\_001\_API\_003

2）被测特性：

* group中缺失参数会触发插入错误，执行失败
* group中参数完整，插入成功，执行成功

3）测试方法：

* group参数使用数据不完整的SensorGroup对象，预期插入失败
* group参数使用数据完整的SensorGroup对象，预期插入成功

4）测试项标志

|  |  |  |
| --- | --- | --- |
| 测试项目标识 | 测试项描述 | 优先级 |
| BACK\_001\_API\_003\_01 | addNewGroup插入数据失败 | 高 |
| BACK\_001\_API\_003\_02 | addNewGroup插入数据成功，返回List<SensorGroup> | 高 |

5）测试通过\失败标准

所有的用例执行结果与预期一致

6）对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_API\_003\_01 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | addNewGroup插入数据失败 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 数据不完整的group | 插入失败，报错 | 插入失败，报错 | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_API\_003\_02 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | addNewGroup插入数据成功，返回List<SensorGroup> | | | | | |
| 前置条件 | 无 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 数据完整的group | 插入成功，返回所有数据 | 插入成功，返回所有数据 | 无 | / |

getAllGroup测试分析与设计

1）标识符定义：BACK\_001\_API\_004

2）被测特性：

* 数据库中无数据，返回空值
* 数据库中有数据，返回SensorGroup中的所有数据，执行成功

3）测试方法：

* 数据库无对应数据时查询，返回空值
* 数据库中有对应数据时查询，返回List<SensorGroup>，执行成功

4）测试项标志

|  |  |  |
| --- | --- | --- |
| 测试项目标识 | 测试项描述 | 优先级 |
| BACK\_001\_API\_004\_01 | 对空数据库查询，返回空值 | 高 |
| BACK\_001\_API\_004\_02 | 数据库中存在数据，返回List<SensorGroup> | 高 |

5）测试通过\失败标准

所有的用例执行结果与预期一致

6）对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_API\_004\_01 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 对空数据库查询，返回空值 | | | | | |
| 前置条件 | 数据库相关表中无数据 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 无 | null | null | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_API\_004\_02 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 数据库中存在数据，返回List<SensorGroup> | | | | | |
| 前置条件 | 数据库中相关表有数据 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 无 | 返回包含全部数据的List<SensorGroup> | 返回包含全部数据的List<SensorGroup> | 无 | / |

1. **MsgReciever类测试函数（方法）参考表**

|  |  |  |
| --- | --- | --- |
| 标识符 | 名称 | 代码行（LOC） |
| BACK\_001\_MSG\_001 | Object sendMsg(String msg) | 27 |

sendMsg测试分析与设计

1）标识符定义：BACK\_001\_MSG\_001

2）被测特性：

* 当msg包含“/”时，会对msg做处理，把msg里的字符串进行包装，并添加到msgList里发送出去
* 当msg不包含“/”时，不发送信息

3）测试方法：

* 传入带“/”的msg，检查是否有信息被发送
* 传入不带“/”的msg，检查是否有信息被发送

4）测试项标志

|  |  |  |
| --- | --- | --- |
| 测试项目标识 | 测试项描述 | 优先级 |
| BACK\_001\_MSG\_001\_01 | 传入带“/”的msg，检查是否有信息被发送 | 低 |
| BACK\_001\_MSG\_001\_02 | 传入不带“/”的msg，检查是否有信息被发送 | 高 |

5）测试通过\失败标准

所有的用例都必须被执行，且没有发现错误

6）对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_MSG\_001\_01 | | | | | |
| 优先级 | 低 | | | | | |
| 测试项描述 | 传入带“/”的msg，检查是否有信息被发送 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 带“/”的msg参数 | 检测到信息发送 | 检测到信息发送 | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_MSG\_001\_02 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 传入不带“/”的msg，检查是否有信息被发送 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 不带“/”的msg参数 | 未检测到信息发送 | 未检测到信息发送 | 无 | / |

1. **OrderReciever类测试函数（方法）参考表**

|  |  |  |
| --- | --- | --- |
| 标识符 | 名称 | 代码行（LOC） |
| BACK\_001\_ORD\_001 | void excuteOrder(String order) | 27 |

excuteOrder测试分析与设计

1）标识符定义：BACK\_001\_ORD\_001

2）被测特性：

* 传入字符串带有“state”，根据携带信息改变Sensor的开启状态
* 传入字符串带有“msg”，更新ClientMsg
* 传入字符串带有“create”，创建新的Clients

3）测试方法：

* 传入带state/true的字符串，检查SensorState是否为true
* 传入带state/false的字符串，检查SensorState是否为false
* 传入带msg的字符串，检查ClientMsg是否更新
* 传入带create的字符串，检查是否有新的Client被创建

4）测试项标志

|  |  |  |
| --- | --- | --- |
| 测试项目标识 | 测试项描述 | 优先级 |
| BACK\_001\_ORD\_001\_01 | 传入带state/true的字符串 | 低 |
| BACK\_001\_ORD\_001\_02 | 传入带state/false的字符串 | 高 |
| BACK\_001\_ORD\_001\_03 | 传入带msg的字符串 | 高 |
| BACK\_001\_ORD\_001\_04 | 传入带create的字符串 |  |

5）测试通过\失败标准

所有的用例都必须被执行，且没有发现错误

6）对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_ORD\_001\_01 | | | | | |
| 优先级 | 低 | | | | | |
| 测试项描述 | 传入带state/true的字符串 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 边界值 | 带state/true的字符串 | SensorState=true | SensorState=true | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_ORD\_001\_02 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 传入带state/false的字符串 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 带state/false的字符串 | SensorState=false | SensorState=flase | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_ORD\_001\_03 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 传入带msg的字符串 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 带msg的字符串 | ClientMsg更新 | ClientMsg更新 | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_ORD\_001\_04 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 传入带create的字符串 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 带create的字符串 | 有新的Client被创建 | 有新的Client被创建 | 无 | / |

1. **StorageReciever类测试函数（方法）参考表**

|  |  |  |
| --- | --- | --- |
| 标识符 | 名称 | 代码行（LOC） |
| BACK\_001\_STO\_001 | void storage(string str) | 27 |

storage测试分析与设计

1）标识符定义：BACK\_001\_STO\_001

2）被测特性：

* str参数如果被‘/’分割为2部分，将str处理后与当前时间一起插入数据库
* str参数如果被‘/’分割后的数组长度不为2，不做处理，不插入数据

3）测试方法：

* 传入带一个“/”字符的str参数，检查数据库是否插入数据
* 传入不带或带多个“/”字符的str参数，检查数据库是否插入数据

4）测试项标志

|  |  |  |
| --- | --- | --- |
| 测试项目标识 | 测试项描述 | 优先级 |
| BACK\_001\_STO\_001\_01 | 传入带一个“/”字符的str参数 | 高 |
| BACK\_001\_STO\_001\_02 | 传入不带或带多个“/”字符的str参数 | 高 |

5）测试通过\失败标准

所有的用例都必须被执行，且没有发现错误

6）对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_STO\_001\_01 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 传入带一个“/”字符的str参数 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 带一个“/”字符的str参数 | 数据库插入新数据，这个新数据由str与当前时间组成 | 数据库插入新数据，这个新数据由str与当前时间组成 | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | BACK\_001\_STO\_001\_01 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 类型为ILF的情况 | | | | | |
| 前置条件 | 传入类型ILF | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 不带“/”字符的str参数 | 数据库没有插入新数据 | 数据库没有插入新数据 | 无 | / |
| 002 | 等价类 | 带多个“/”字符的str参数 | 数据库没有插入新数据 | 数据库没有插入新数据 | 无 | / |

1. **SensorClientThread类测试函数（方法）参考表**

|  |  |  |
| --- | --- | --- |
| 标识符 | 名称 | 代码行（LOC） |
| BACK\_001\_SEN\_001 | void run() | 27 |

run测试分析与设计

1）标识符定义：BACK\_001\_SEN\_001

2）被测特性：

* 执行函数，线程启动，执行成功。
* 执行函数，报错，执行失败。

3）测试方法：

* 执行函数，查看线程是否启动成功

4）测试项标志

|  |  |  |
| --- | --- | --- |
| 测试项目标识 | 测试项描述 | 优先级 |
| BACK\_001\_SEN\_001\_01 | 添加线程，执行run函数 | 低 |

5）测试通过\失败标准

所有的用例都必须被执行，且没有发现错误

6）对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UT\_TC\_001\_002\_001 | | | | | |
| 优先级 | 低 | | | | | |
| 测试项描述 | 添加线程，执行run函数 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例序号 | 测试方法 | 输入 | 期望结果 | 真实结果 | 是否有bug | 是否解决 |
| 001 | 等价类 | 无 | 线程启动成功 | 线程启动成功 | 无 | / |

用例执行及测试报告：