**第3章-保险产品**

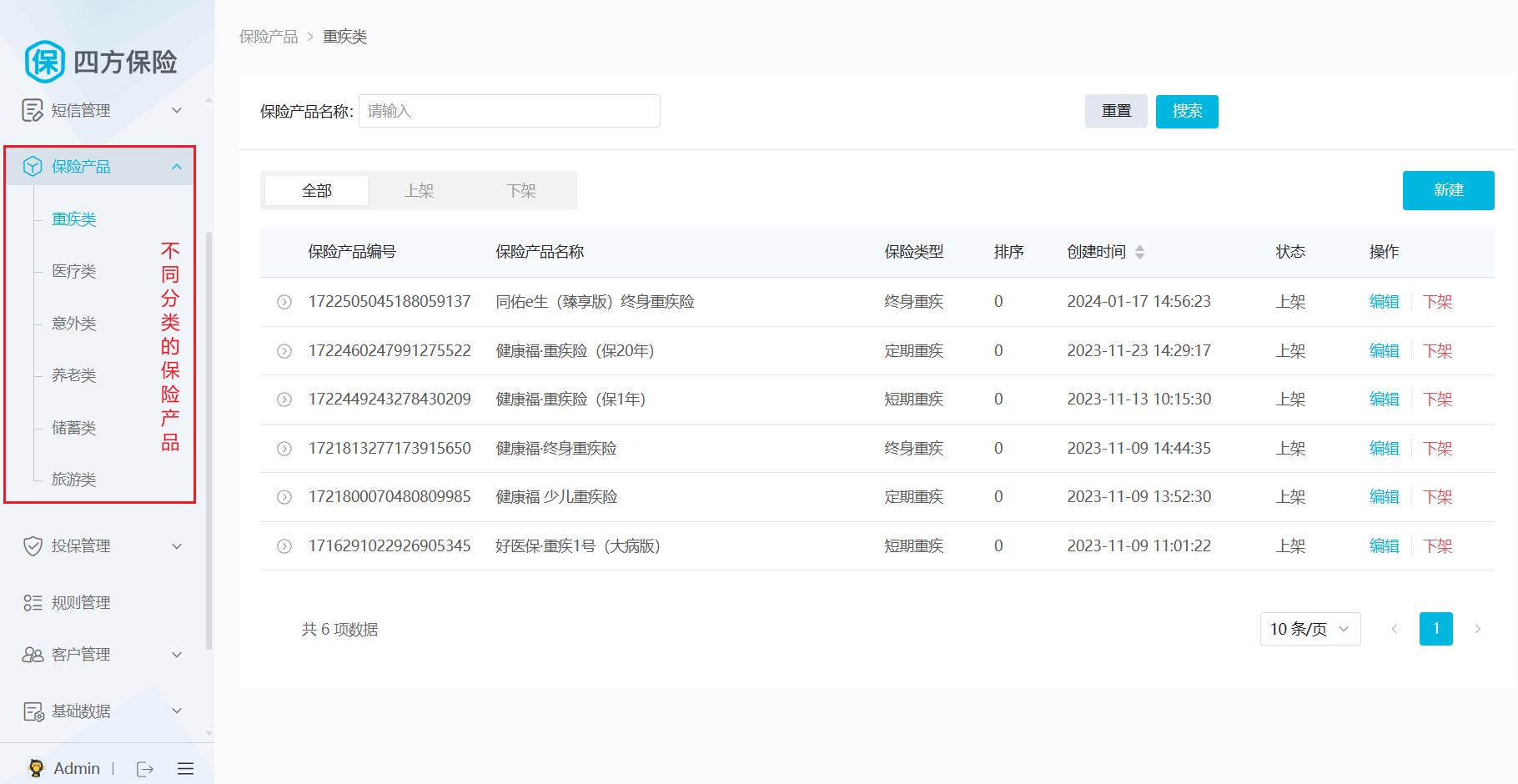
**学习目标**

* 分析《四方保险》保险发布的整体业务，掌握复杂业务的分析方法
* 掌握保险发布-产品信息的关键属性及关系，并完成产品信息的开发
* 掌握保险发布-产品方案的关键属性及关系，并完成产品方案的开发
* 掌握保险发布-产品系数的关键属性及关系，并完成产品系数的开发

**1、保险产品业务说明**

**1.1、保险产品组成**

访问管理后台：http://sf.mgt.itheima.net/#/product/diseases?id=1

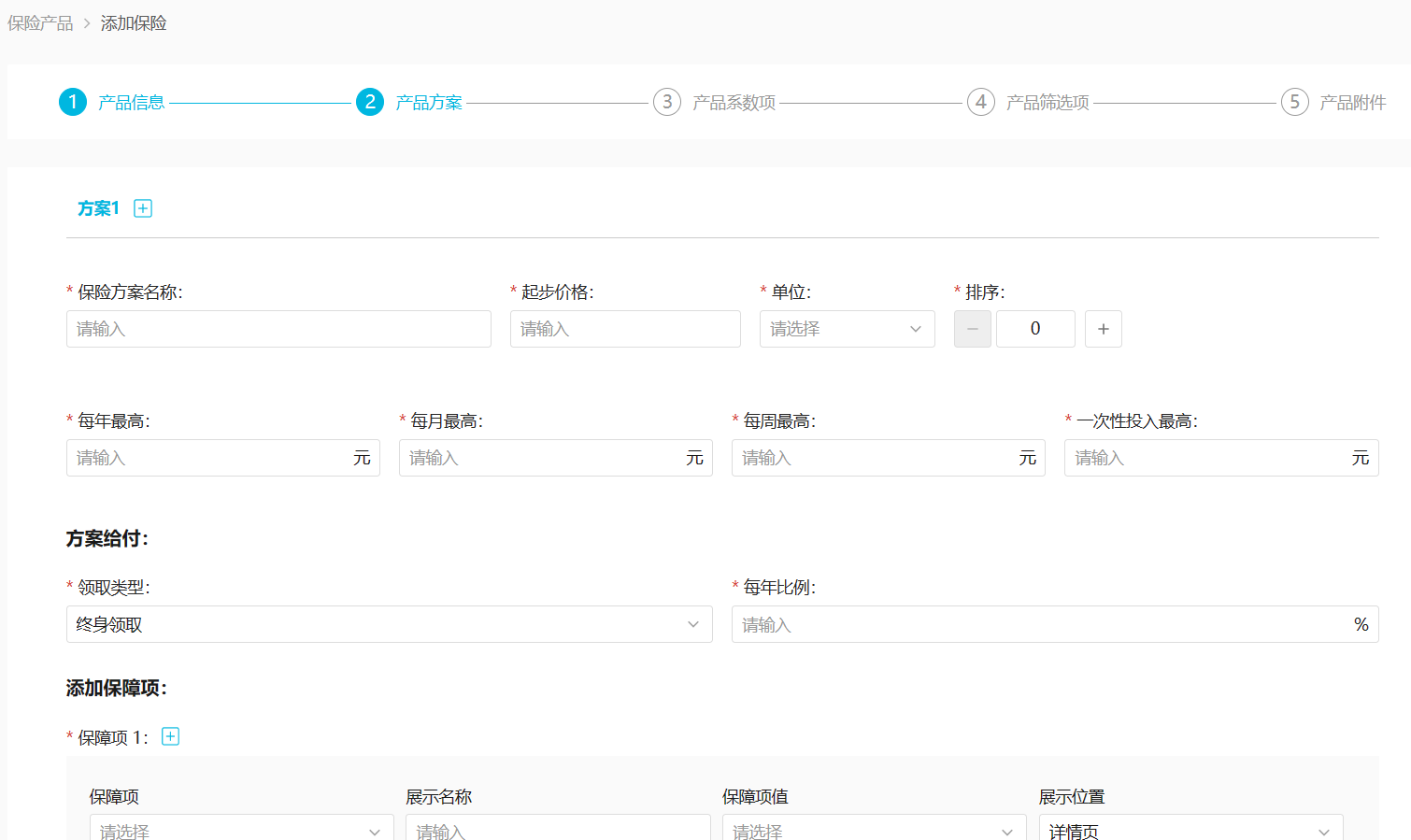


保险发布：可以发布不同分类的保险产品；保险的发布组成部分包括；

* 产品信息



* 产品方案
* 方案保障项
* 方案给付计划（储蓄类、理财类）



* 产品系数项



* 产品筛选项



* 产品附件



上述的 保险产品的组成需要基础数据：分类、保障项、系数项、筛选项的支持。

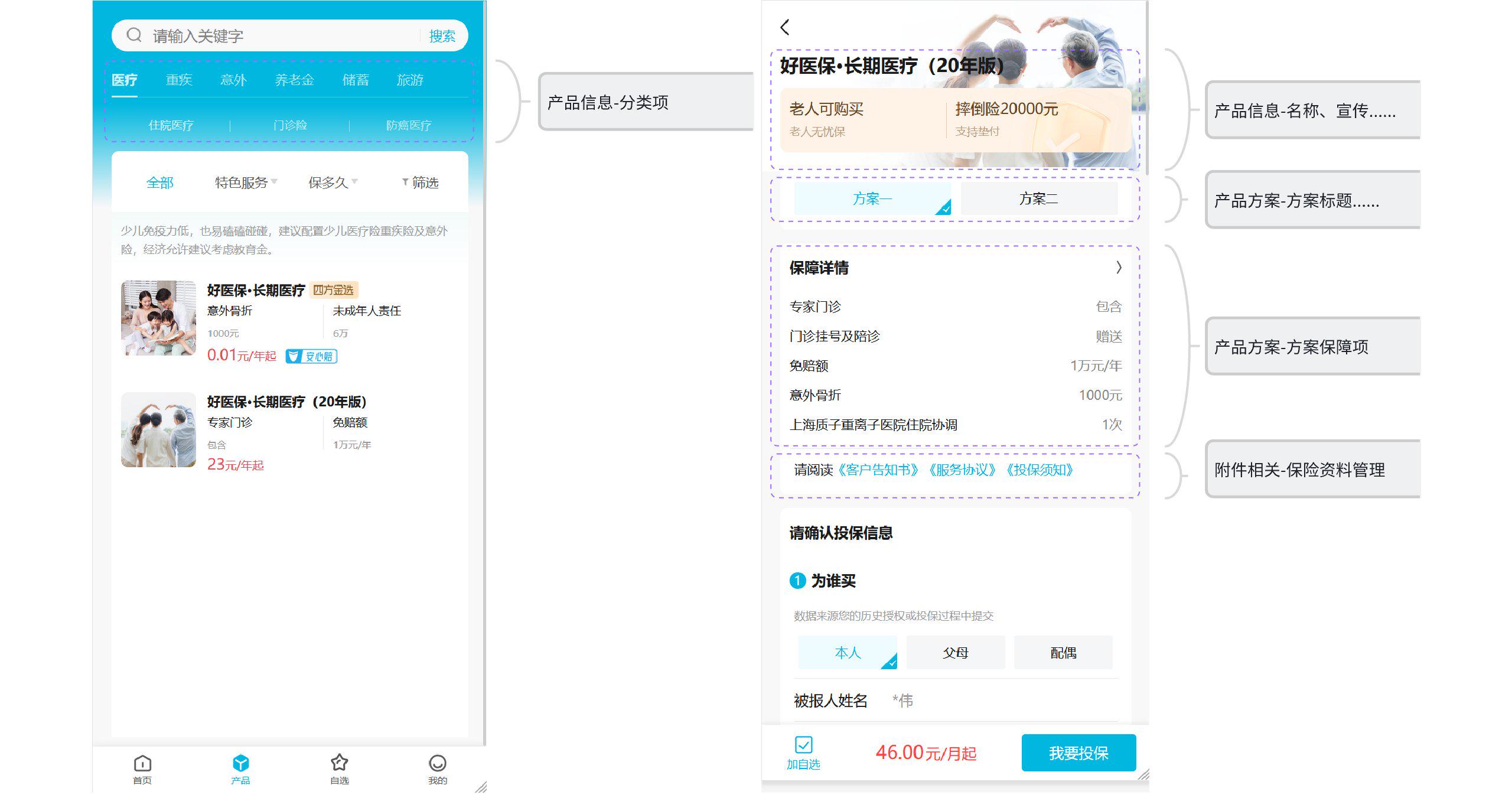
在页面中有 1/2/3/4/5 等分步式的填写表单，逐步逐步的操作，每完成一步之后才能进入下一步，直到引导用户完成整个业务的做法叫做“**分步式表单**”或者“**向导式界面**”。

**1.2、业务流程**

保险产品在后台管理中添加了信息之后；需要上架发布；然后用户才可以在app端中筛选、搜索找到产品；找到产品之后在保险详情页面中选择具体的方案，填写系数确定保险的价格，查看合同生成合同订单、支付；完成投保的整个流程。

**1.2.1、界面操作**

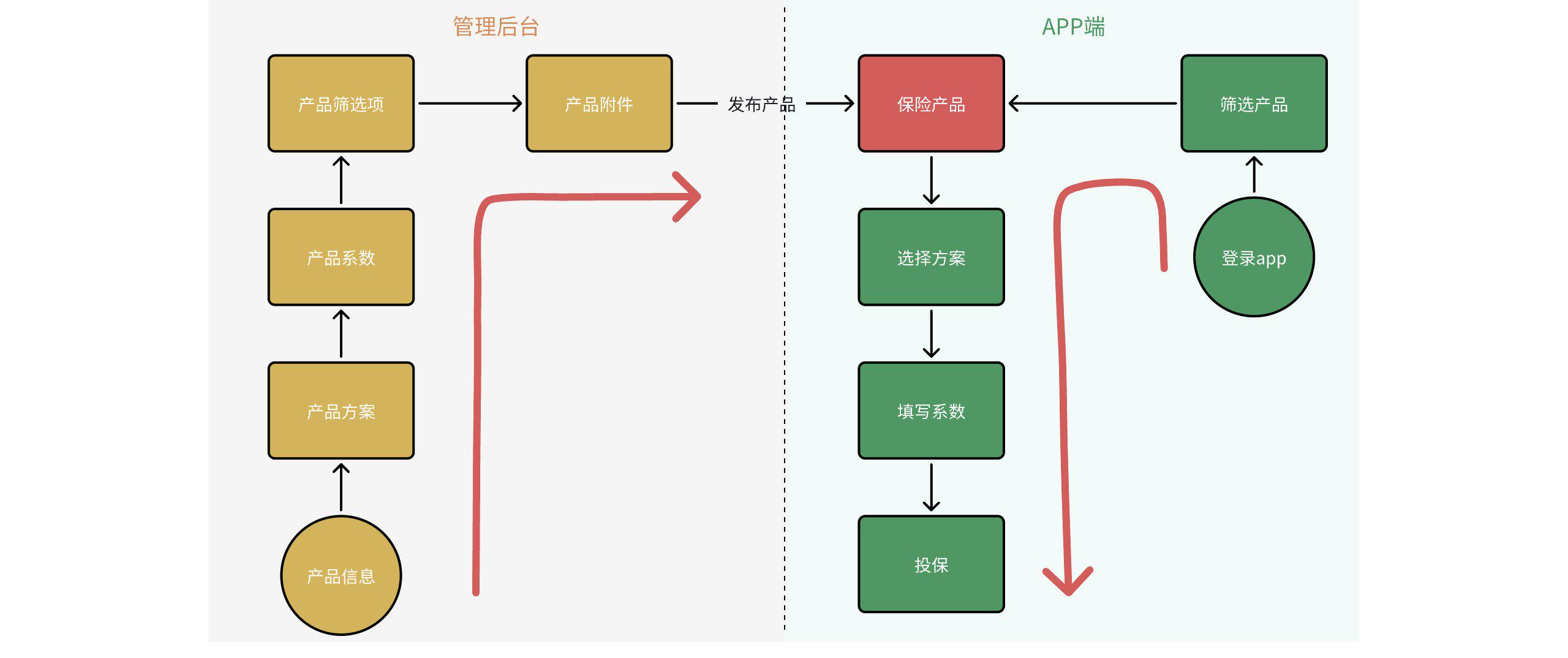
在APP端的页面信息：





**1.2.2、业务流程**

结合后台管理系统与APP端的操作；梳理如下与保险产品相关的流程：

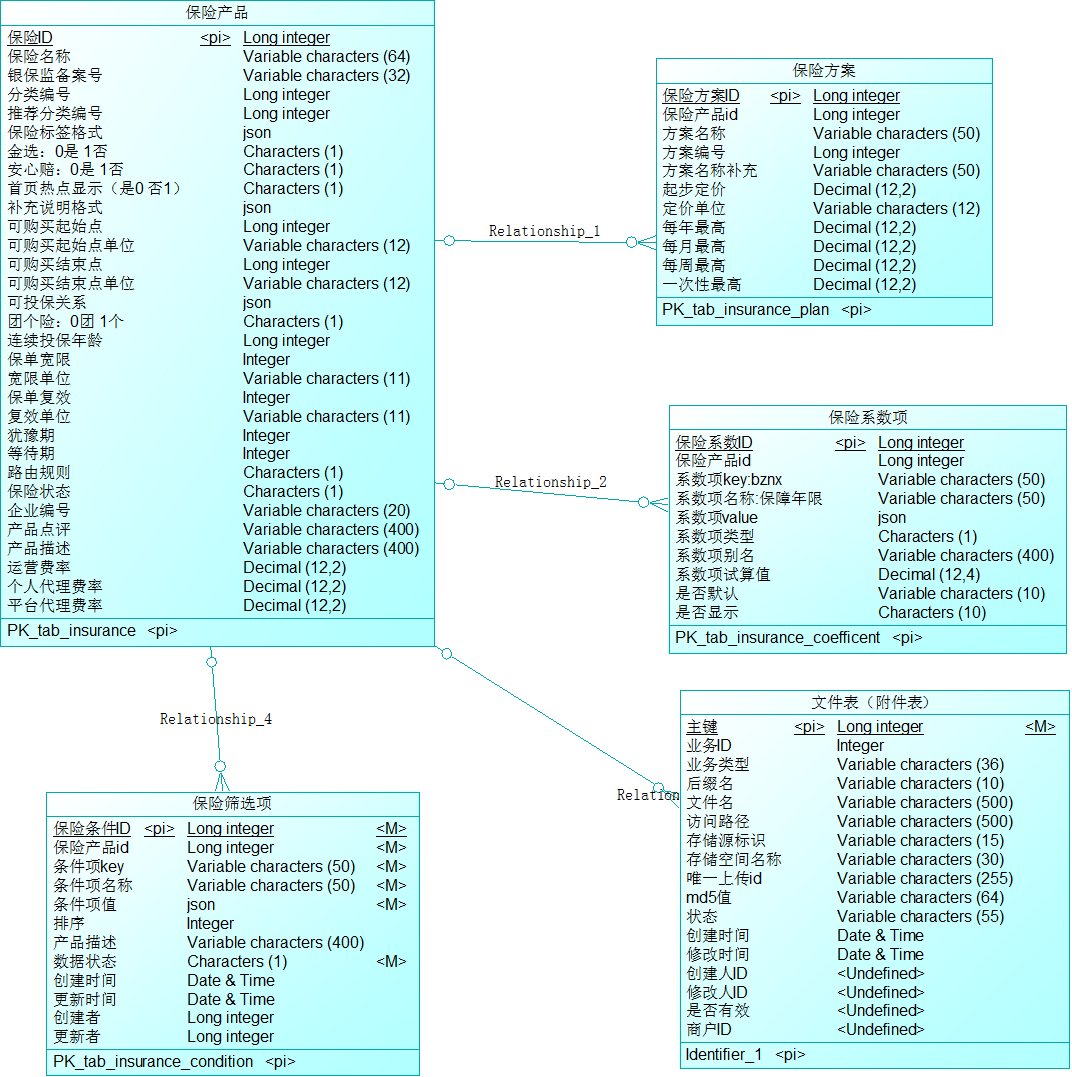


**1.3、产品组成关系**

作为在后台管理中要完成的 保险产品 它与方案、系数项、筛选项、附件的关系分析如下：

* 一个保险产品；可以添加多个**保险方案** ===》一对多
* 一个保险产品；可以配置多个**保险系数项** ===》 一对多
* 一个保险产品；可以配置多个**筛选项** ===》 一对多
* 一个保险产品；可以携带多个**附件** ===》一对多

结合系统原型或前端页面；分析得出具体的保险产品以及方案、系数项、筛选项、附件的数据库概念模型如下：



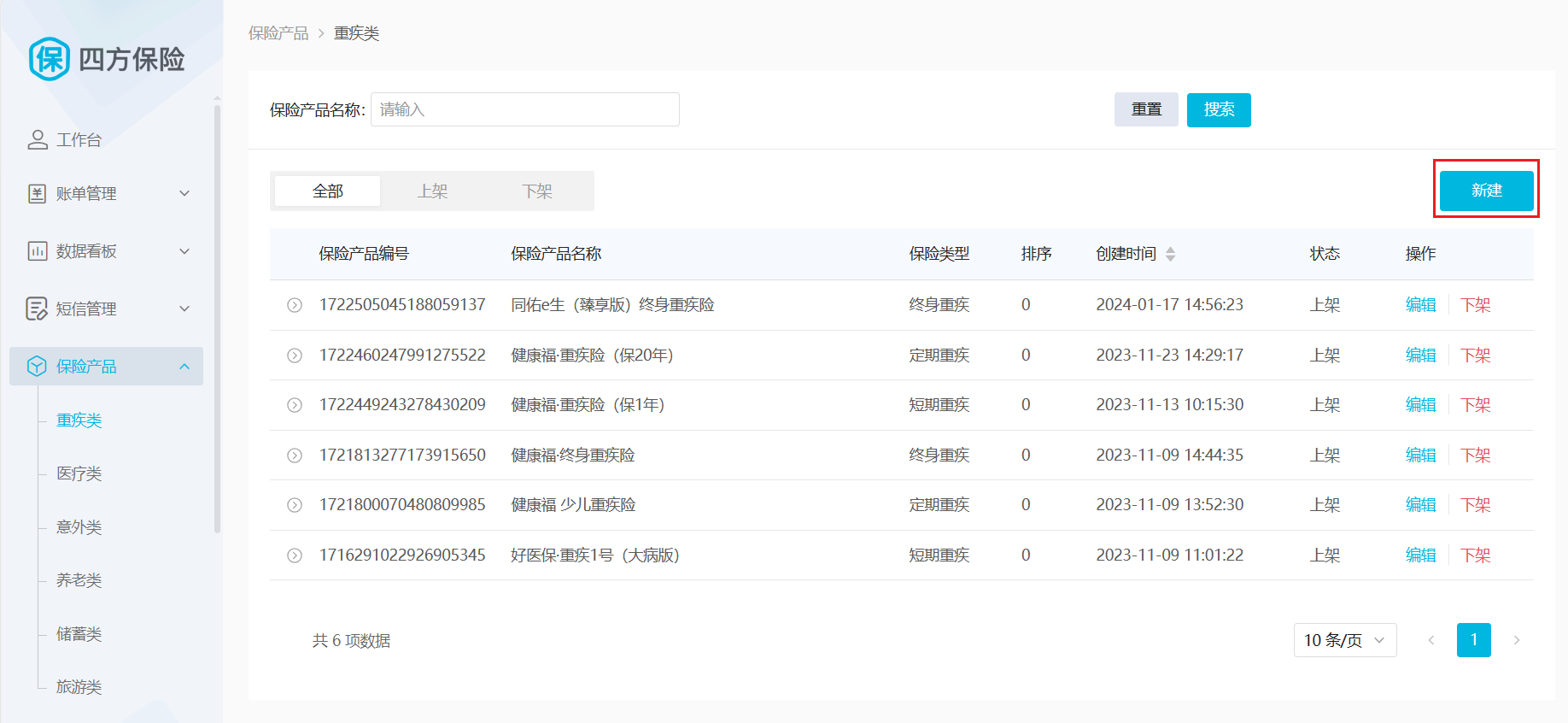
|  |
| --- |
| 基于上述分析的产品关系；如果要删除一个产品和对应关系；那么语句可以参考如下： |

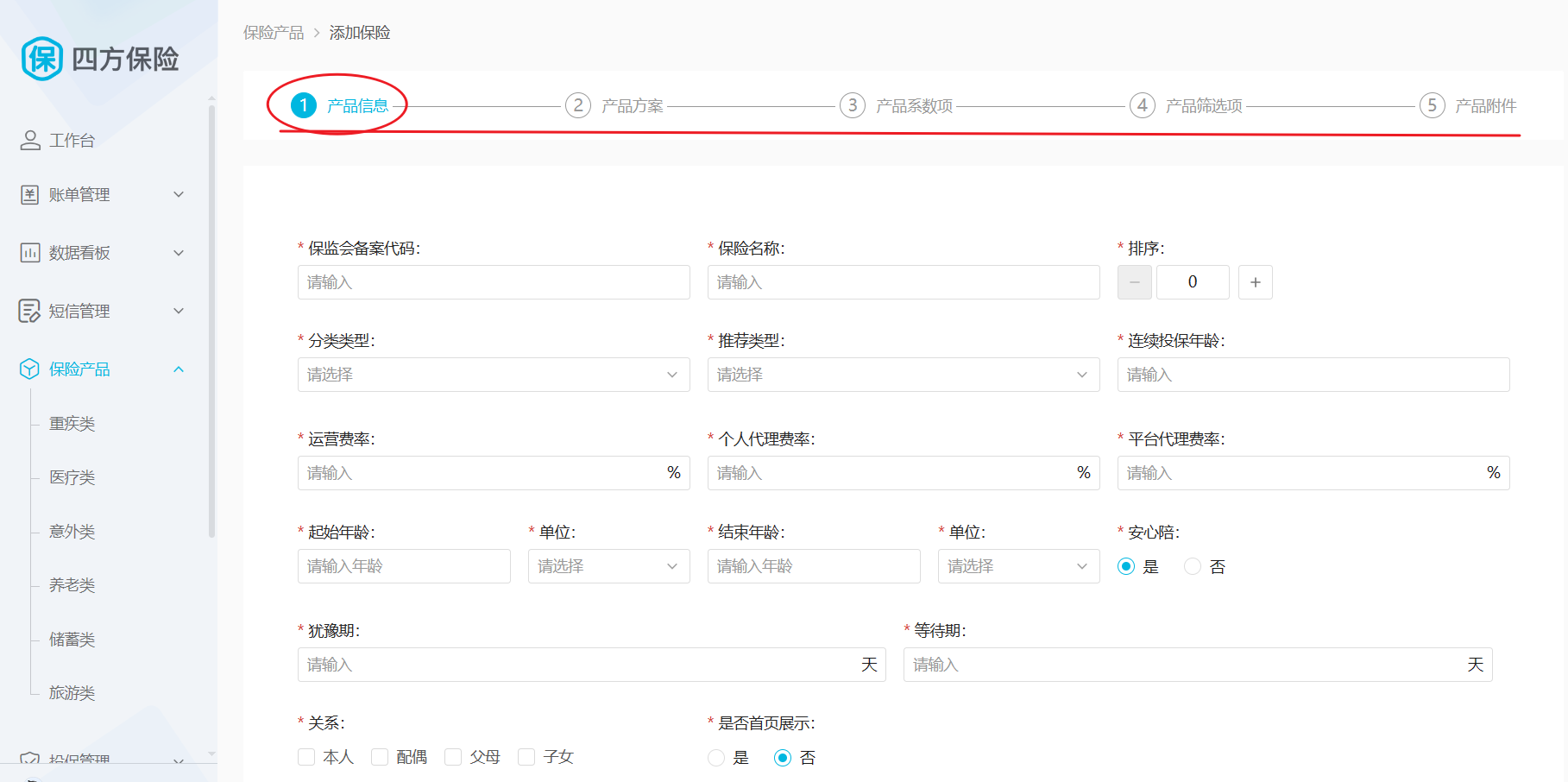
|  |
| --- |
| SQL *-- 删除保险产品相关内容* delete from tab\_insurance\_plan where insurance\_id = 保险产品id ; delete from tab\_insurance\_coefficent where insurance\_id = 保险产品id ; delete from tab\_insurance\_condition where insurance\_id = 保险产品id ; delete from tab\_file where business\_id = 保险产品id ; delete from tab\_insurance where id = 保险产品id ;  -- 删除某个保险产品的所有数据  delete from 方案保障项 where 保险方案ID in （select 保险方案ID from 保险方案 where 保险产品ID = ？） delete from 方案给付 t1, 保险方案 t2 where t1.保险方案ID=t2.保险方案ID and 保险产品ID=? -- 方式三 delete from 方案给付 t1 where exists (select \* from 保险方案 where 保险方案ID=t1.保险方案ID)  delete from 保险方案 where 保险产品ID = ？ delete from 保险筛选项 where 保险产品ID = ？ delete from 保险系数项 where 保险产品ID = ？ delete from 附件 where 业务ID = ？ delete from 保险产品表 where 保险ID = ？ |

**2、产品信息**

**2.1、需求分析**

接下来；我们先完成管理后台中保险产品管理-新建功能；也就是如下页面：



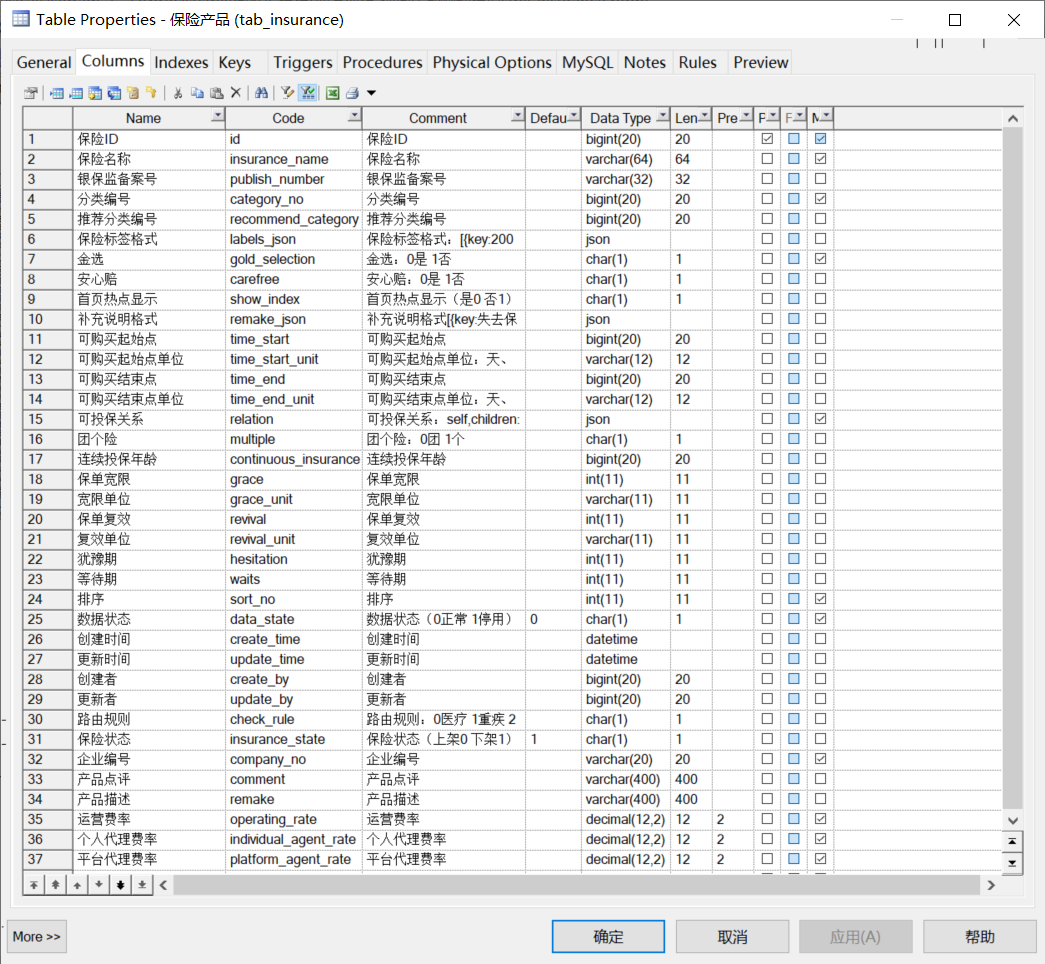


根据界面中内容，逐个输入信息分析；也在过程中熟悉业务概念：

|  |  |
| --- | --- |
| 输入项 | 说明 |
| **保监会备案代码** | 在 中国保险监督管理委员会 进行备案的项目或产品的唯一标识代码。这个代码用于标识和查询已经经过保监会审核并备案的保险产品或服务。通过备案代码，公众可以查询到该产品或服务的详细信息，包括其名称、类型、保险公司、备案时间等。 是一串数值字符串，长度32字符内；可以以字符串存储。 |
| **保险名称** | 保险产品名称；名称要唯一，可区别 |
| **排序** | 产品排序字段；目前暂未使用 |
| **分类类型** | 保险分类类型；属于二级分类中的分类子类；如：重疾类 ，那么可以选择的就是重疾下的子分类 |
| **推荐类型** | app中首页中的推荐类型；可以选择：成年人、少儿、老人 |
| **连续投保年龄** | 在保险合同中规定的，被保险人可以连续投保该保险产品的最高年龄限制（在连续投保保险产品时，被保险人所达到的年龄上限）。这个限制是根据被保险人的生理周期、健康状况以及保险公司的风险控制策略来设定的。  数值类型 |
| **运营费率** | 保险公司为了维持其日常运营和业务开展所支出的费用占保险费用的一定比例。  浮点类型；小数点精确到两位 |
| **个人代理费率** | 指保险公司支付给个人代理人的佣金或费用比例。个人代理人是保险公司的销售人员，他们通过销售保险产品来获取佣金。  浮点类型；小数点精确到两位 |
| **平台代理费率** | 指保险公司支付给合作平台或第三方代理机构的费用比例。  浮点类型；小数点精确到两位 |
| **起始年龄** | 开始购买或参加某种保险产品的最低年龄限制。  数值类型 |
| **起始年龄单位** | 年或者月；字符类型 |
| **结束年龄** | 可以购买或参加某种保险产品的最高年龄限制。  数值类型 |
| **结束年龄单位** | 年或者月；字符类型 |
| **安心赔** | 赔付的帮助方式；是否可以全程在线理赔申请，全程提供协助  字符类型；0是，1否 |
| **犹豫期** | 犹豫期（也称为冷静期）是指投保人在收到保险合同后的一段时间内（通常为10天或15天，具体天数可能因地区或产品类型而异），如果对所购买的保险不满意或后悔购买，可以无条件解除保险合同，并申请全额退还已支付的保费。这个期限是为了给投保人一个考虑和评估保险合同的时间。  多少天；属于数值类型 |
| **等待期** | 等待期（也称为免责期或观察期）是指从保险合同生效之日起的一段时间内（具体天数因产品类型而异），即使发生保险事故，保险公司也不承担保险责任或只承担部分责任。等待期的目的是为了防止投保人明知道将发生保险事故而立即购买保险以获取保险金的行为。在等待期内，保险公司会观察被保险人的健康状况或风险情况，以确保保险合同的有效性。  多少天；属于数值类型 |
| **关系** | 本人、配偶、父母、子女  字符类型 |
| **是否首页展示** | app端业务标识符，是否要在首页展示；目前无用 |
| **标识1** | 在app端详情页面，顶部左边展示的内容标题；**属于变动部分；以后app可能有标识2/3/4等；所以标识及对应的描述；建议存储json结构；这样可扩展性强；**结构如：  [{"key": "住院报销400万", "val": "医保报销后接着赔"}, {"key": "保证续保20年", "val": "理赔停售也不断保"}]  上面的结构中；第一个key就是标识1 ；第一个val就是标识1描述；以此类推 |
| **标识1描述** | 在app端详情页面，顶部左边展示的内容描述 |
| **标识2** | 在app端详情页面，顶部右边展示的内容标题；与标识1一样；与标识1、描述一样；存储在一个json字段中即可；同标识1 |
| **标识2描述** | 同标识1 |
| **补充说明1** | 在app端详情页面；作为补充说明使用，作用和标识是一样的；目前**暂时没有使用到**。其存储与标识1的做法类似；存储结构如：  [{"key": "住院报销400万", "val": "医保报销后接着赔"}, {"key": "保证续保20年", "val": "理赔停售也不断保"}] |
| **补充说明1描述** | 已经在补充说明1中已经说明，统一存储到一个json字符串中 |
| **补充说明2** | 已经在补充说明1中已经说明，统一存储到一个json字符串中 |
| **补充说明2描述** | 已经在补充说明1中已经说明，统一存储到一个json字符串中 |
| **产品点评标题** | 详情页面展示；目前无用；字符串类型 |
| **产品点评内容** | 详情页面展示；目前无用；字符串类型 |
| **金选** | 是否属于金选类型；金选可以在金选过滤的时候显示；0是，1否；字符类型 |
| **团个险标记** | 旅游类产品专属；标记是否团险还是个险；字符类型，0团，1个 |
| **保单宽限** | 存储理财、养老类专属；在这个时间段内，被保险人可以在未支付保费的情况下继续享受保险责任。也就是说，保单到期后，如果未能及时支付保费，被保险人仍然可以在宽限期内享有保险责任。  数值类型 |
| **保单宽限单位** | 存储理财、养老类专属；字符类型，天、月、年 |
| **保单复效** | 存储理财、养老类专属；指停效保单在自停效之日起的时间内，投保人根据保险合同约定，办理有关手续后，使保单恢复效力。具体来说，如果保单因为某些原因（如未支付保费）而失效，投保人在宽限期结束后至两年内可以向保险公司提出保单复效的申请。如果保险公司同意并接受了投保人补交的保费，那么保单就可以恢复效力。  数值类型 |
| **保单复效单位** | 存储理财、养老类专属；字符类型，天、月、年 |
| **其它公共内容** | 保险产品是属于哪个保司的、状态、创建/更新人、时间等 |

**2.2、数据库表设计**

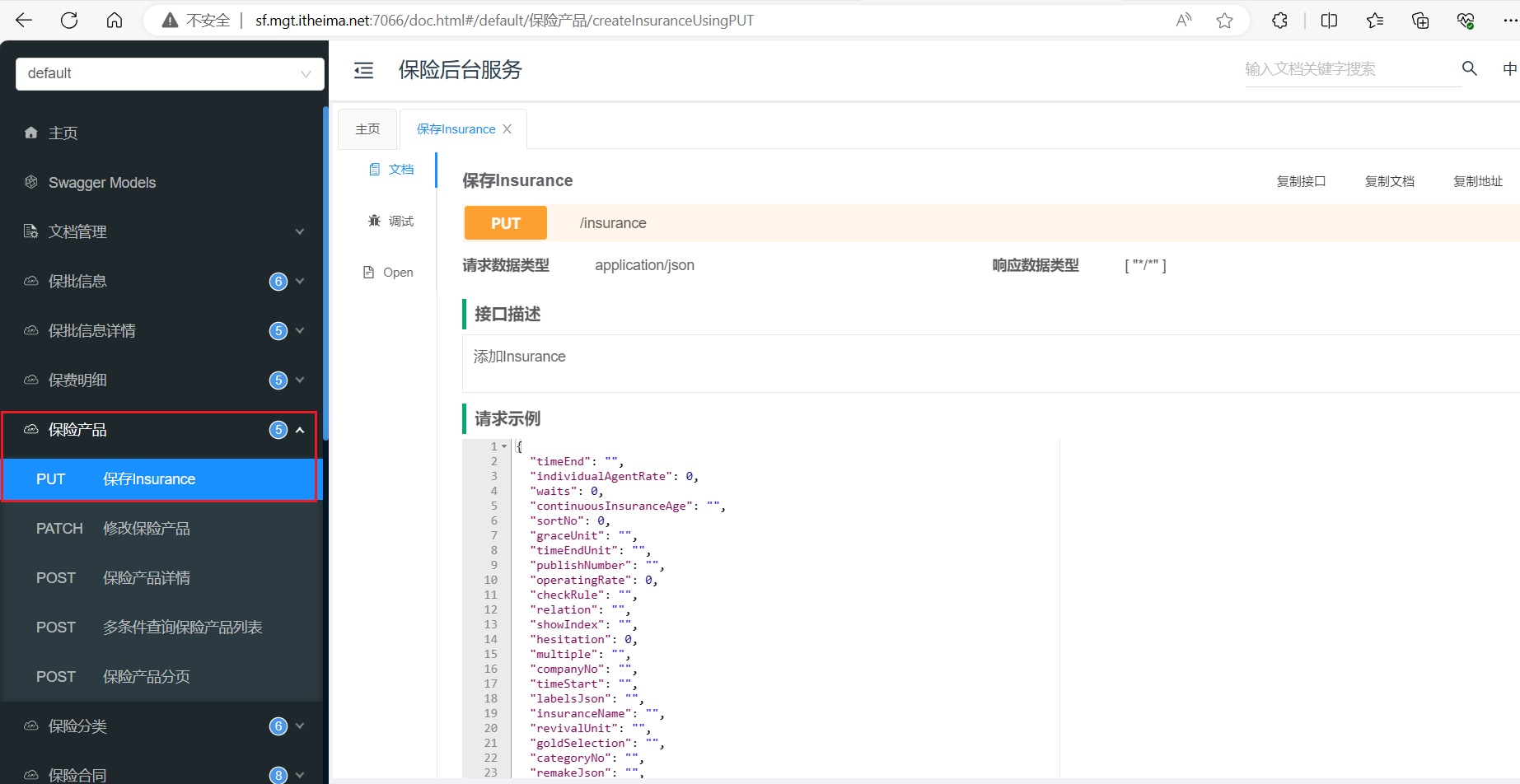
基于在界面中对于输入和在app端需要的产品详情信息的输出展示的需要；设计的数据库表如下：



**2.3、接口分析设计**

我们先查看保存保险产品的接口：

http://sf.mgt.itheima.net:7066/doc.html#/default/%E4%BF%9D%E9%99%A9%E4%BA%A7%E5%93%81/createInsuranceUsingPUT



可以在系统界面中直接添加数据进行测试。也可以使用接口地址进行测试；如果直接接口测试的话，

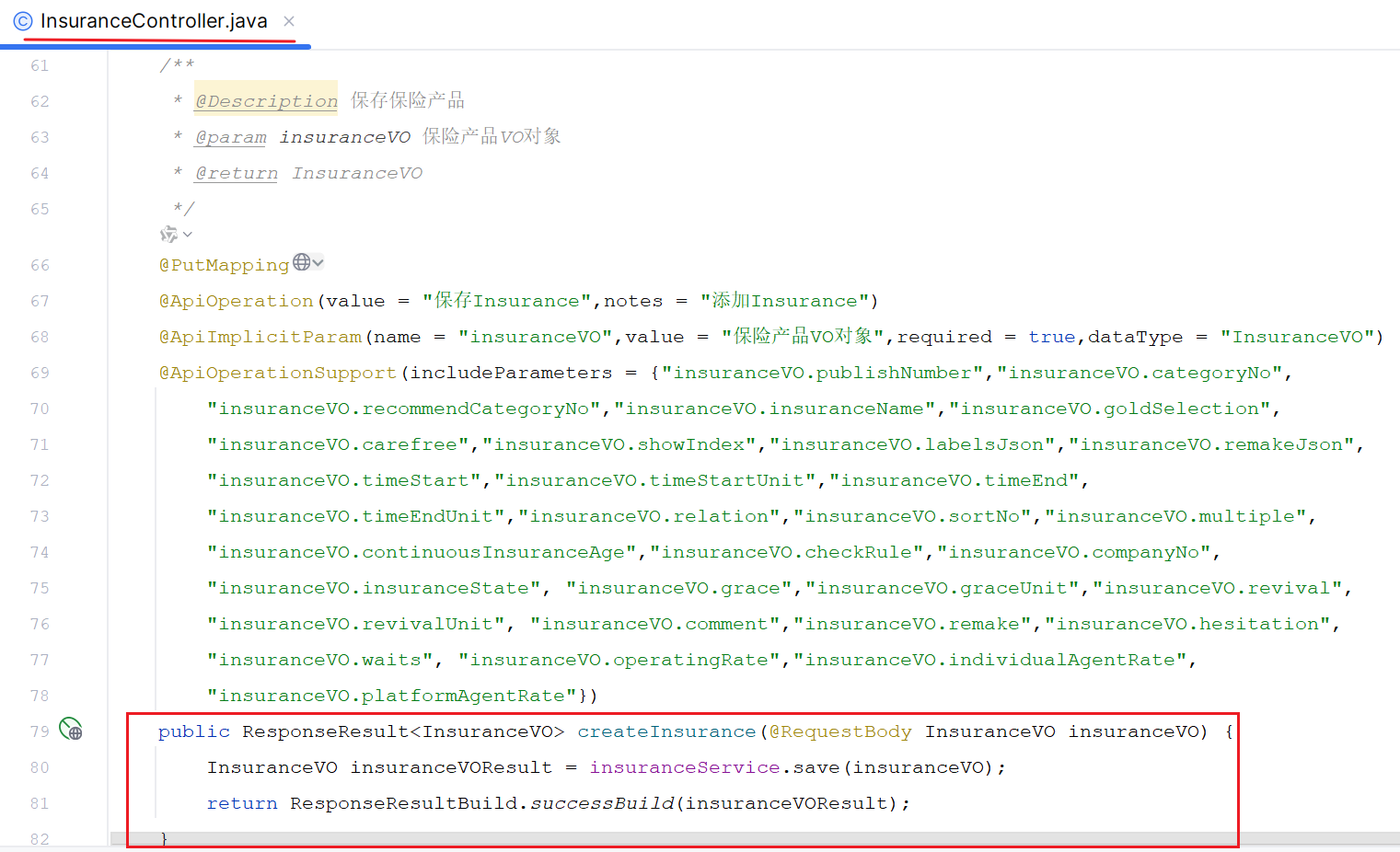
数据结构参考如下：

|  |
| --- |
| JSON {  "publishNumber": "222",  "insuranceName": "定期重疾测试2",  "categoryNo": "100001002002000",  "recommendCategoryNo": "100001008000000",  "checkRule": "1",  "labelsJson": "[{\"key\":\"标识1\",\"val\":\"标识1描述\"},{\"key\":\"标识2\",\"val\":\"标识2描述\"}]",  "remakeJson": "[{\"key\":\"补充说明1\",\"val\":\"补充说明1描述\"},{\"key\":\"22\",\"val\":\"22\"}]",  "timeStart": "23",  "timeStartUnit": "YEAR",  "timeEnd": "45",  "timeEndUnit": "YEAR",  "hesitation": "30",  "waits": "15",  "multiple": "0",  "continuousInsuranceAge": "55",  "relation": "[\"0\"]",  "operatingRate": "12",  "individualAgentRate": "12",  "platformAgentRate": "12",  "goldSelection": "0",  "carefree": "0",  "showIndex": "1",  "sortNo": 1,  "grace": "",  "graceUnit": "",  "revival": "",  "revivalUnit": "",  "remake": "标题",  "comment": "内容" } |

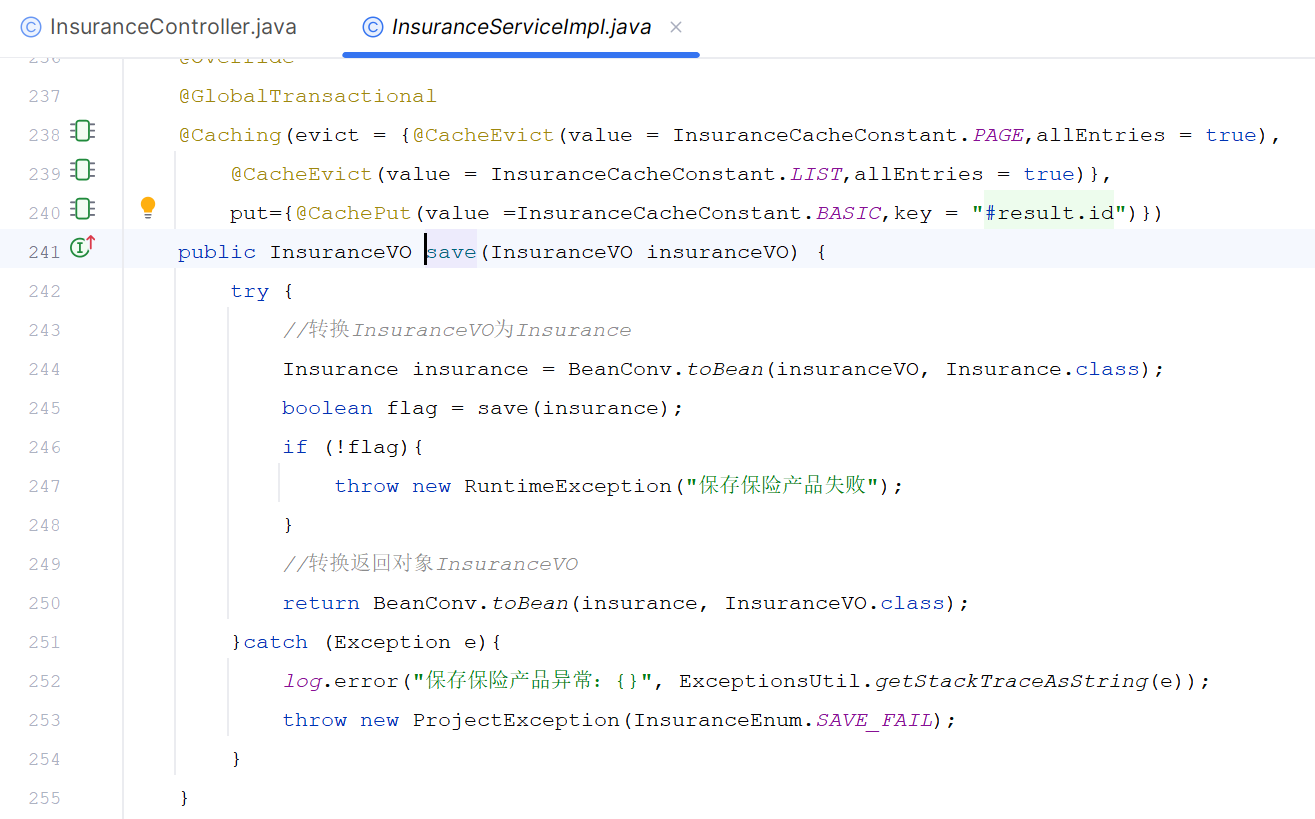
**2.4、开发实现**

由于保存保险产品是比较基础的保存功能；并没有任何特殊处理的部分；参考代码如下：

**处理器代码**：



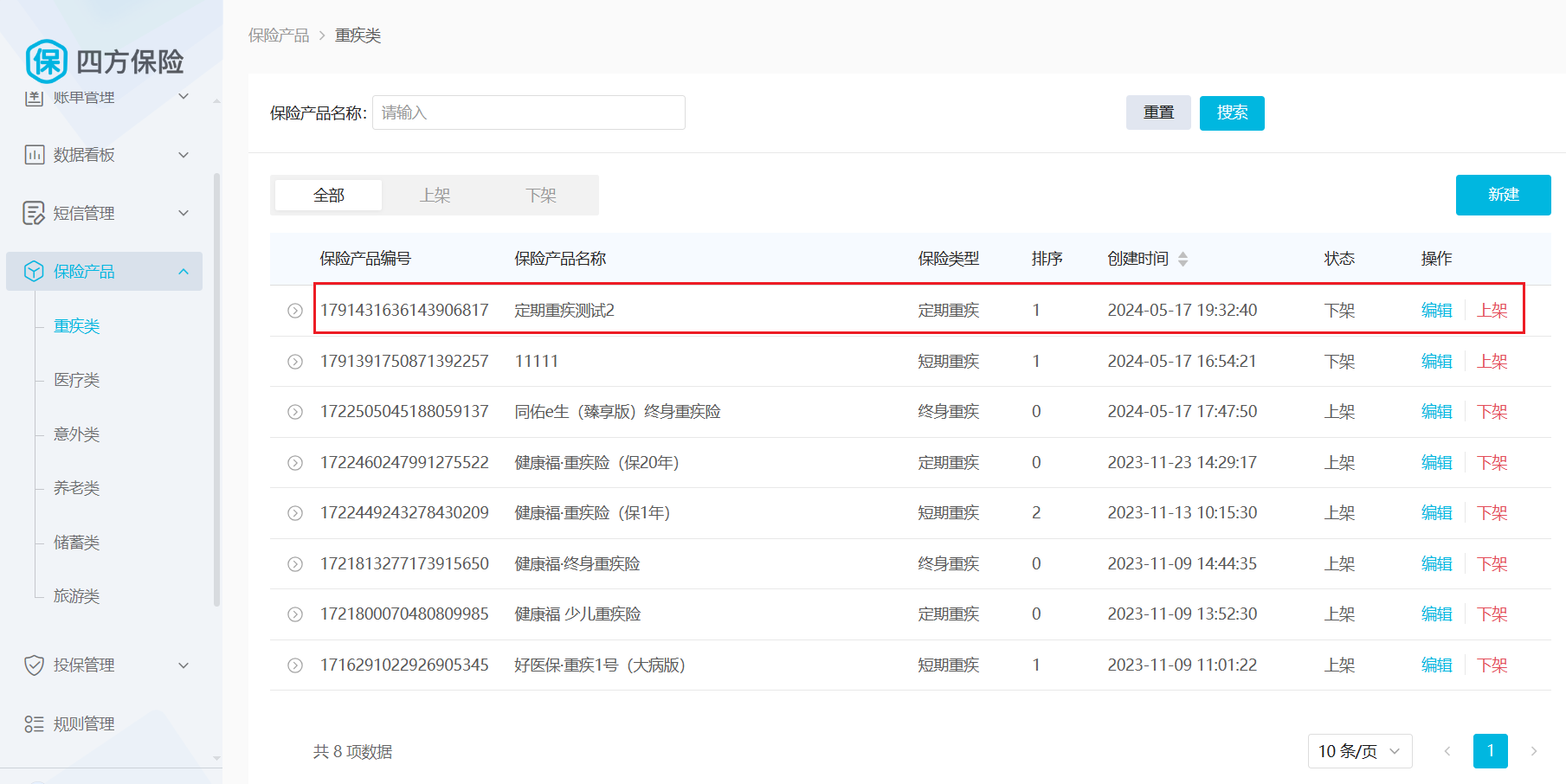
**业务实现**：



**2.5、测试**



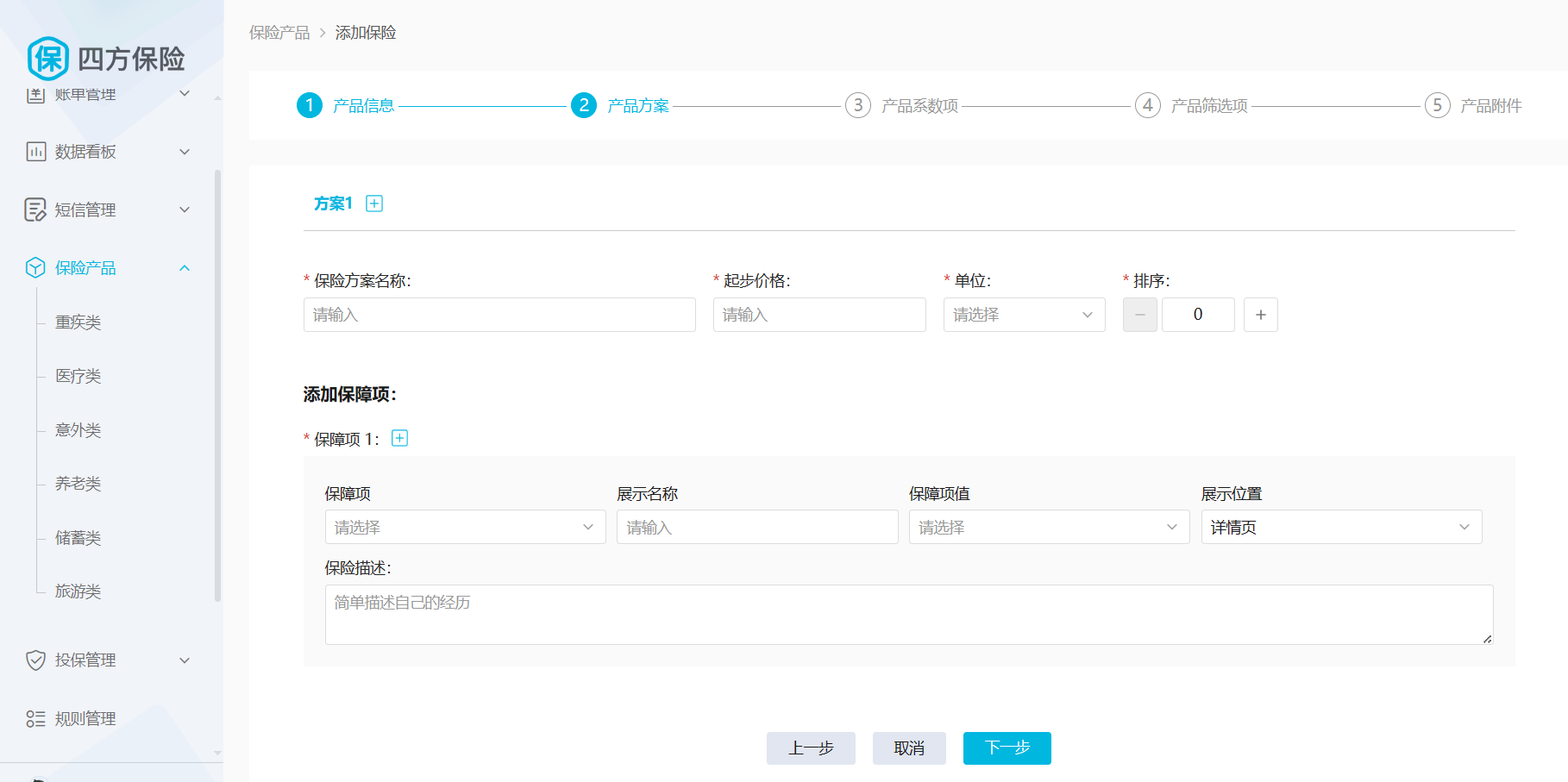
保存完成之后在 保险产品 列表中可以查看到：



**3、产品方案**

**3.1、需求分析**

在保存完 产品信息 之后则跳转到 产品方案；这一界面需求如下：



界面上的输入项分析如下：

|  |  |
| --- | --- |
| 输入项 | 说明 |
| **保险方案名称** | 保险方案名称；将会在app端查看保险产品详情的时候展示；  字符串类型 |
| **起步价格** | 买这个保险产品对应的方案的价格；（这个价格不是最终的，还与系数有关）  数值类型 |
| **单位** | 字符类型；元/天，元/月，元/年 |
| **排序** | 方案的排序值，暂无用；数值类型 |
| **每年最高** | 养老、储蓄类专属；每年最高投入的总金额 |
| **每月最高** | 养老、储蓄类专属；每月最高投入的总金额 |
| **每周最高** | 养老、储蓄类专属；每周最高投入的总金额 |
| **一次性投入最高** | 养老、储蓄类专属；一次最高投入的金额上限 |
| **领取类型** | 养老、储蓄类专属；内容为：终身领取0，固定领取1，字符类型  **终身领取**是指从保险合同规定的领取年龄起，保险公司会按终身领取的基本保险金额，每年给付一次年金，直到被保险人身故，合同终止。  **固定领取**则是指在一个特定的时间段内，每年都可以领取到固定比例的养老年金。 |
| **每年比例** | 只有当领取类型为**终身领取**的时候才有；百分比；浮点类型 |
| **领取总期数** | 只有当领取类型为**固定领取**的时候才有；数值类型 |
| **第1...n期** | 根据不同的总期数，自动生成，每期都可以设置领取额度百分比；浮点类型  由于这个期数也是不固定的，那么可以作为一个json字段；每期都作为其属性存储；可以无限期数 |
| **保障项** | 保障项名称；每个方案都可以有很多保障项，所以可以将保障项编号也保存 |
| **展示名称** | 展示在app端展示位置（详情页面）中的 保障详情里面显示的：保障项名字（也就是可以对原有的保障项起别名并在详情页面中展示） |
| **保障项值** | 具体保障项的值 |
| **展示位置** | 这个保障项在哪些位置中展示 |
| **保险描述** | 这个保障项对应的保险产品方案信息描述 |
| **其它公共内容** | 状态、创建/更新人、时间等 |

**3.2、数据库表设计**

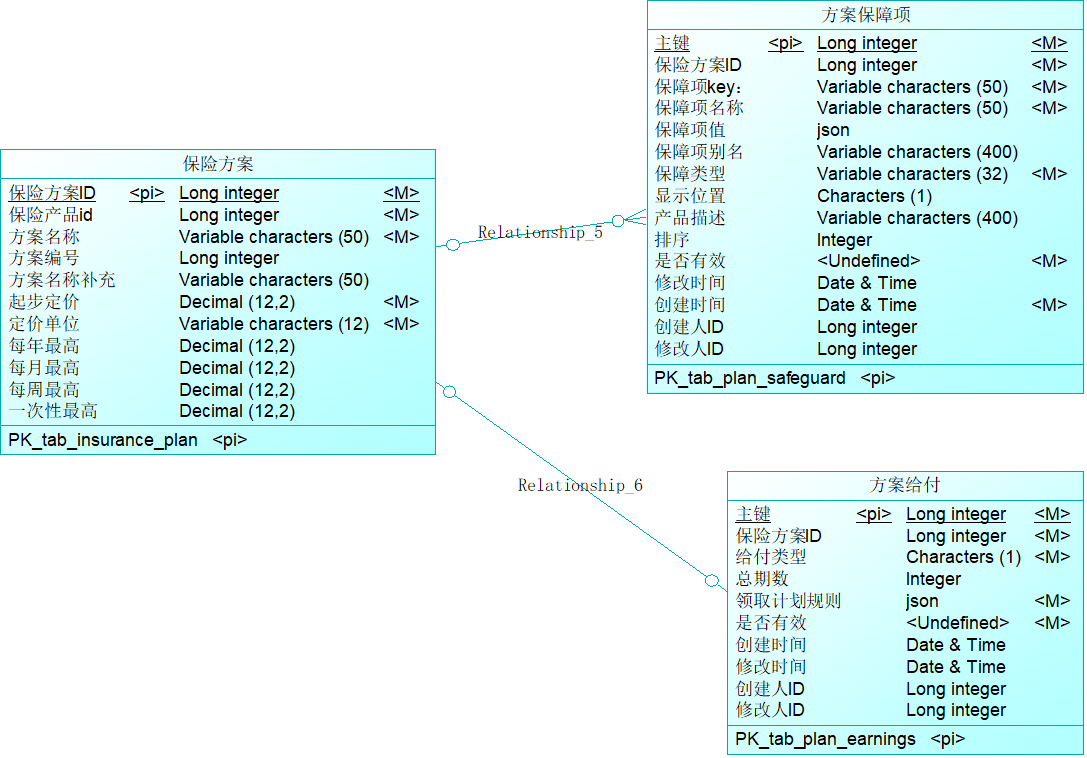
根据需求分析；得出保险方案与 方案保障项、方案给付的关系为：

* 一个保险方案可以有多个方案保障项 ===》一对多
* 一个保险方案可以有一个方案给付 ===》一对一

|  |
| --- |
| 一对一关系；可以不分两张表；也可以将方案给付分开保存到另外一种从表；  四方保险采用从表方案，因为里面有一个给付计划json大字段且不常用字段；所以单独创建一张从表保存方案给付计划 |

**3.2.1、保险方案概念模型**

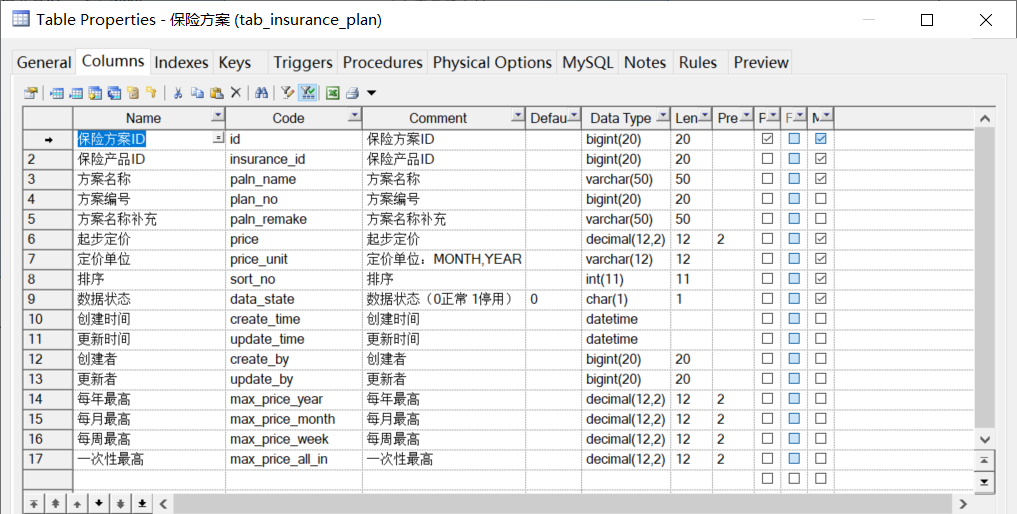
与保险方案相关的概念模型如下：



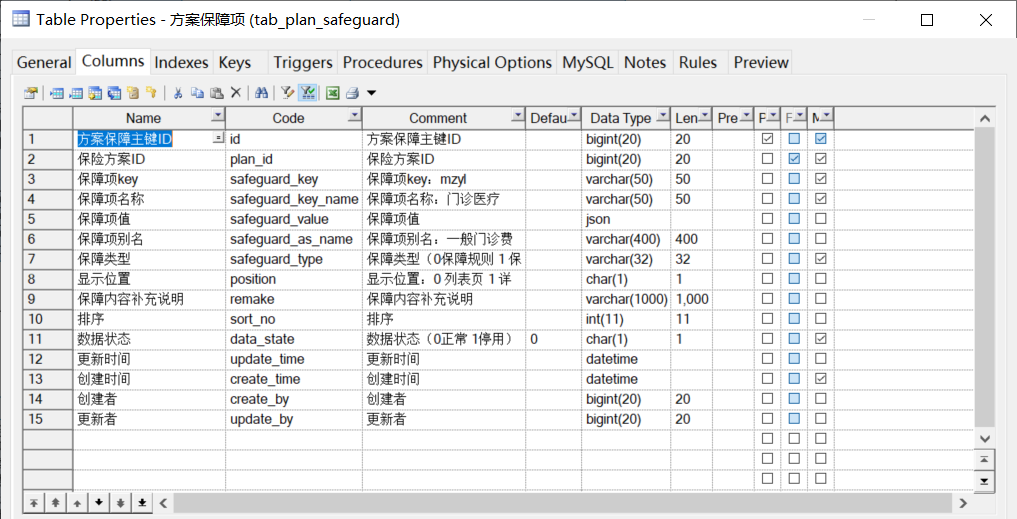
**3.2.2、保险方案物理模型**



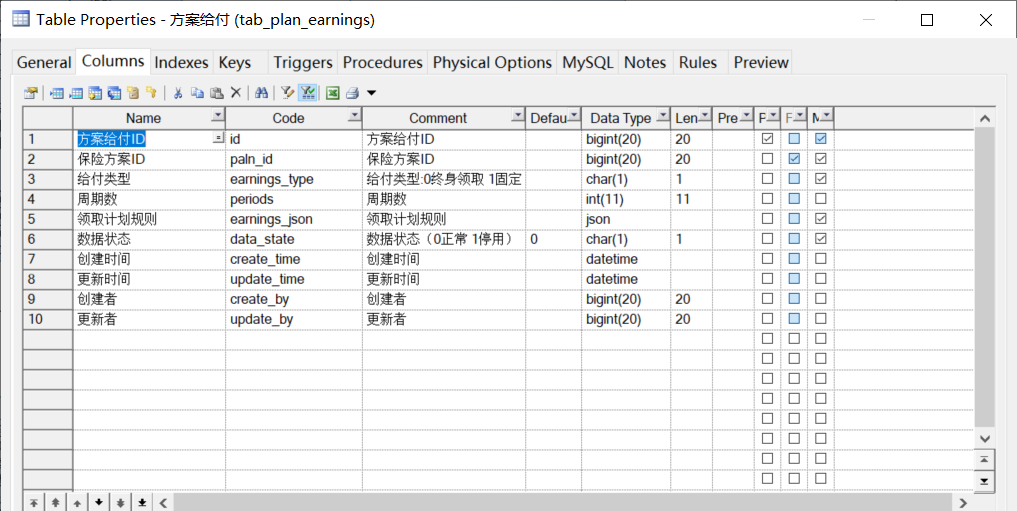
**1）保险方案表**



**2）方案保障项表**

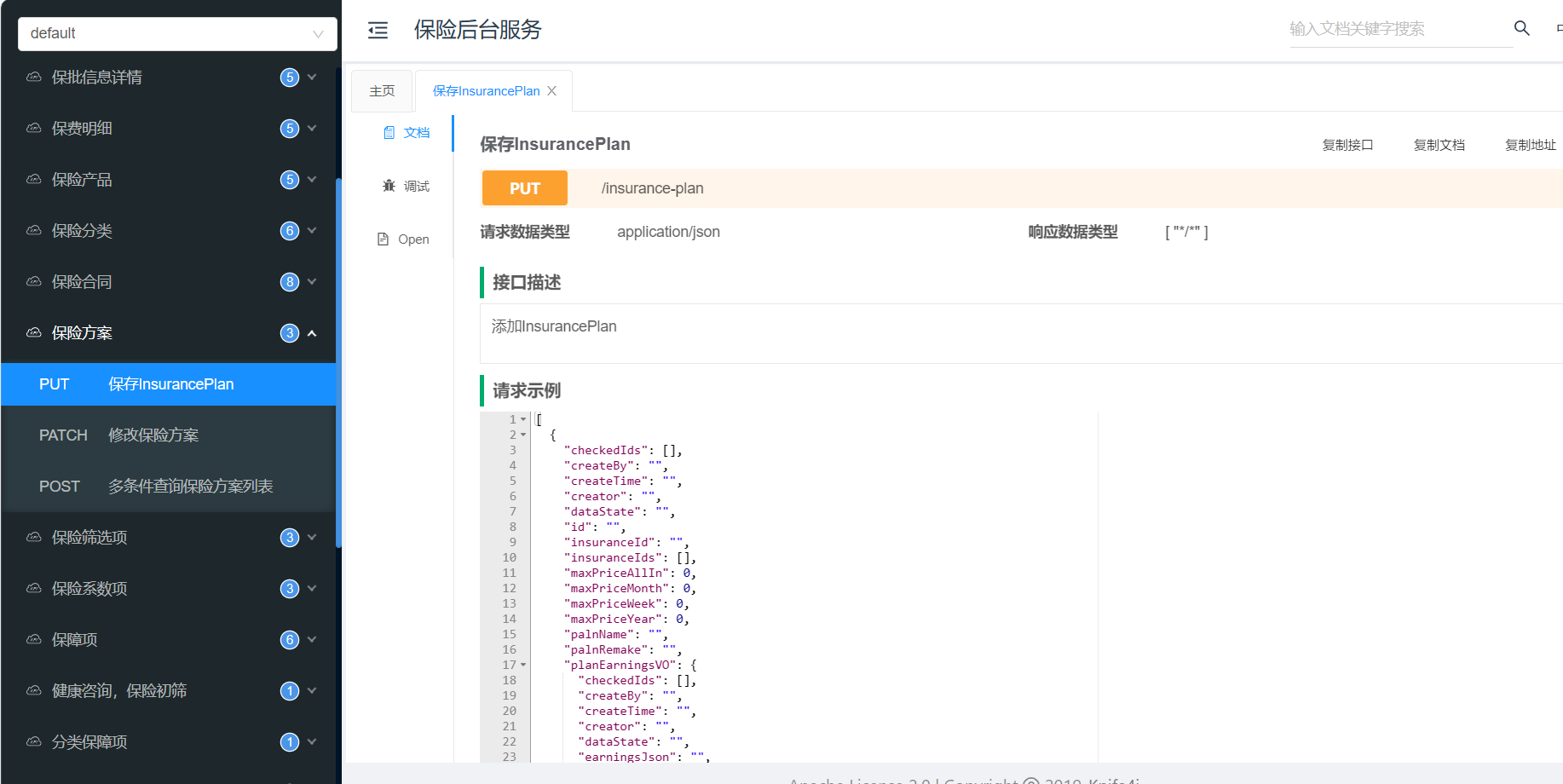


**3）方案给付表**



**3.3、接口分析设计**

保存 保险产品方案的接口地址：http://sf.mgt.itheima.net:7066/doc.html#/default/%E4%BF%9D%E9%99%A9%E6%96%B9%E6%A1%88/createInsurancePlansUsingPUT



在接口测试时；提交到后端的保险产品方案数据结构，参考大概结构与数据如下：

|  |
| --- |
| JSON [  {  "id": "1791464440806252548",  "createTime": "2024-05-17 21:43:01",  "updateTime": "2024-05-17 21:43:01",  "createBy": "1371500419615895553",  "updateBy": "1371500419615895553",  "dataState": "0",  "creator": null,  "insuranceId": "1791431636143906817",  "palnName": "方案一",  "planNo": 1791464440806252500,  "palnRemake": "",  "price": 1000,  "priceUnit": "YEAR",  "maxPriceYear": null,  "maxPriceMonth": null,  "maxPriceWeek": null,  "maxPriceAllIn": null,  "sortNo": 0,  "checkedIds": null,  "planSafeguardVOs": [  {  "createTime": "2024-05-17 21:43:06",  "updateTime": "2024-05-17 21:43:06",  "createBy": "1371500419615895553",  "updateBy": "1371500419615895553",  "dataState": "0",  "creator": null,  "planId": "1791464440806252548",  "safeguardKey": "BZ009",  "safeguardKeyName": "恶性肿瘤特效药",  "safeguardValue": "{\"val\":\"0\",\"name\":\"赠送\",\"score\":\"0\"}",  "safeguardVOs": null,  "safeguardAsName": "特效药",  "safeguardType": "0",  "position": "2",  "remake": "",  "sortNo": null,  "checkedIds": null,  "safeguardOptions": [  {  "val": "0",  "name": "赠送",  "score": "0"  }  ]  },  {  "createTime": "2024-05-17 21:43:06",  "updateTime": "2024-05-17 21:43:06",  "createBy": "1371500419615895553",  "updateBy": "1371500419615895553",  "dataState": "0",  "creator": null,  "planId": "1791464440806252548",  "safeguardKey": "BZ001",  "safeguardKeyName": "一般医疗保险金",  "safeguardValue": "{\"val\":\"200\",\"name\":\"200万\",\"score\":\"200\"}",  "safeguardVOs": null,  "safeguardAsName": "一般医疗保险金",  "safeguardType": "0",  "position": "2",  "remake": "",  "sortNo": null,  "checkedIds": null,  "safeguardOptions": [  {  "val": "200",  "name": "200万",  "score": "200"  }  ]  }  ],  "insuranceIds": null  },  {  "id": "1791464440873361410",  "createTime": "2024-05-17 21:43:01",  "updateTime": "2024-05-17 21:43:01",  "createBy": "1371500419615895553",  "updateBy": "1371500419615895553",  "dataState": "0",  "creator": null,  "insuranceId": "1791431636143906817",  "palnName": "方案2",  "planNo": 1791464440806252500,  "palnRemake": "",  "price": 800,  "priceUnit": "MONTH",  "maxPriceYear": null,  "maxPriceMonth": null,  "maxPriceWeek": null,  "maxPriceAllIn": null,  "sortNo": 0,  "checkedIds": null,  "planSafeguardVOs": [  {  "createTime": "2024-05-17 21:43:06",  "updateTime": "2024-05-17 21:43:06",  "createBy": "1371500419615895553",  "updateBy": "1371500419615895553",  "dataState": "0",  "creator": null,  "planId": "1791464440873361410",  "safeguardKey": "BZ005",  "safeguardKeyName": "特定药品费用医疗保险金",  "safeguardValue": "{\"val\":\"200\",\"name\":\"200万\",\"score\":\"200\"}",  "safeguardVOs": null,  "safeguardAsName": "保险金",  "safeguardType": "0",  "position": "2",  "remake": "",  "sortNo": null,  "checkedIds": null,  "safeguardOptions": [  {  "val": "200",  "name": "200万",  "score": "200"  }  ]  },  {  "createTime": "2024-05-17 21:43:06",  "updateTime": "2024-05-17 21:43:06",  "createBy": "1371500419615895553",  "updateBy": "1371500419615895553",  "dataState": "0",  "creator": null,  "planId": "1791464440873361410",  "safeguardKey": "BZ006",  "safeguardKeyName": "门诊挂号及陪诊",  "safeguardValue": "{\"val\":\"0\",\"name\":\"赠送\",\"score\":\"0\"}",  "safeguardVOs": null,  "safeguardAsName": "陪诊",  "safeguardType": "0",  "position": "2",  "remake": "",  "sortNo": null,  "checkedIds": null,  "safeguardOptions": [  {  "val": "0",  "name": "赠送",  "score": "0"  }  ]  }  ],  "insuranceIds": null  } ] |

**3.4、开发实现**

**3.4.1、处理器**



**3.4.2、业务实现**

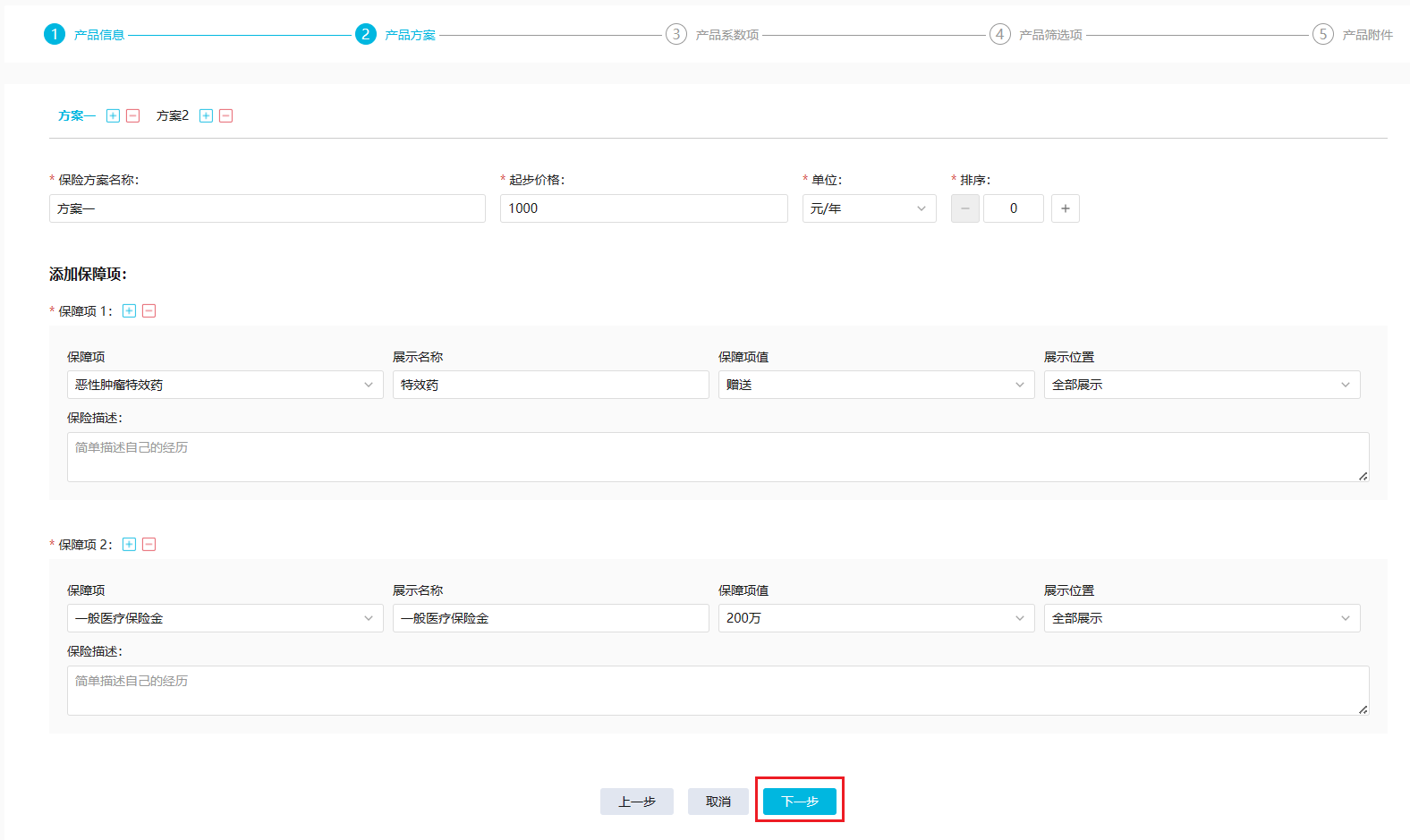
完成 com.itheima.sfbx.insurance.service.impl.InsurancePlanServiceImpl#save

完善（删除后重写）业务代码，实现保险方案的保存功能：

|  |
| --- |
| Java @Override @Transactional @Caching(evict = {@CacheEvict(value = InsurancePlanCacheConstant.*PAGE*,allEntries = true),  @CacheEvict(value = InsurancePlanCacheConstant.*LIST*,allEntries = true)}) public Boolean save(List<InsurancePlanVO> insurancePlanVOs) {  try {  *//最终操作结果* boolean flag = false;  *//1、批量保存保险方案*  *//给所有的保险方案设置方案编号* insurancePlanVOs.forEach(insurancePlan->{  insurancePlan.setPlanNo(identifierGenerator.nextId(insurancePlan).longValue());  });  *//转换vo为实体* List<InsurancePlan> insurancePlans = BeanConv.*toBeanList*(insurancePlanVOs, InsurancePlan.class);  *//保存* flag = saveBatch(insurancePlans);   *//因为保存方案成功后，方案的id在insurancePlans里面；insurancePlanVOs的每个方案是没有方案id的；*  *// 所以可以将方案编号与方案id放到map中对应起来*  *//Map<方案编号,方案id>* Map<Long, Long> planNoAndIdMap = insurancePlans.stream()  .collect(Collectors.*toMap*(InsurancePlan::getPlanNo, InsurancePlan::getId));   *//2、批量保存每个方案的保障项*  *//给方案保障项设置方案编号* insurancePlanVOs.forEach(insurancePlanVO->{  insurancePlanVO.getPlanSafeguardVOs().forEach(planSafeguardVO->{  planSafeguardVO.setPlanId(planNoAndIdMap.get(insurancePlanVO.getPlanNo()));  });  List<PlanSafeguard> planSafeguardList = BeanConv.*toBeanList*(insurancePlanVO.getPlanSafeguardVOs(), PlanSafeguard.class);  *//批量保存保障项* planSafeguardService.saveBatch(planSafeguardList);  });   *//3、批量保存每个方案的给付计划* insurancePlanVOs.forEach(insurancePlanVO->{  *//只有养老、储蓄类型的才有给付计划* if (insurancePlanVO.getPlanEarningsVO() != null) {  insurancePlanVO.getPlanEarningsVO().setPalnId(planNoAndIdMap.get(insurancePlanVO.getPlanNo()));  planEarningsService.save(insurancePlanVO.getPlanEarningsVO());  }  });   return flag;  }catch (Exception e){  *log*.error("保存保险方案异常：{}", ExceptionsUtil.*getStackTraceAsString*(e));  throw new ProjectException(InsurancePlanEnum.*SAVE\_FAIL*);  } } |

**3.5、测试**

在如下的界面中点击 下一步 的时候；如能跳转到第三步 产品系数项 说明保存保险产品方案成功。



点击 下一步 之后；再次查看这个产品，产品方案如果有数据的话说明保存成功。

**4、产品系数项**

**4.1、需求分析**

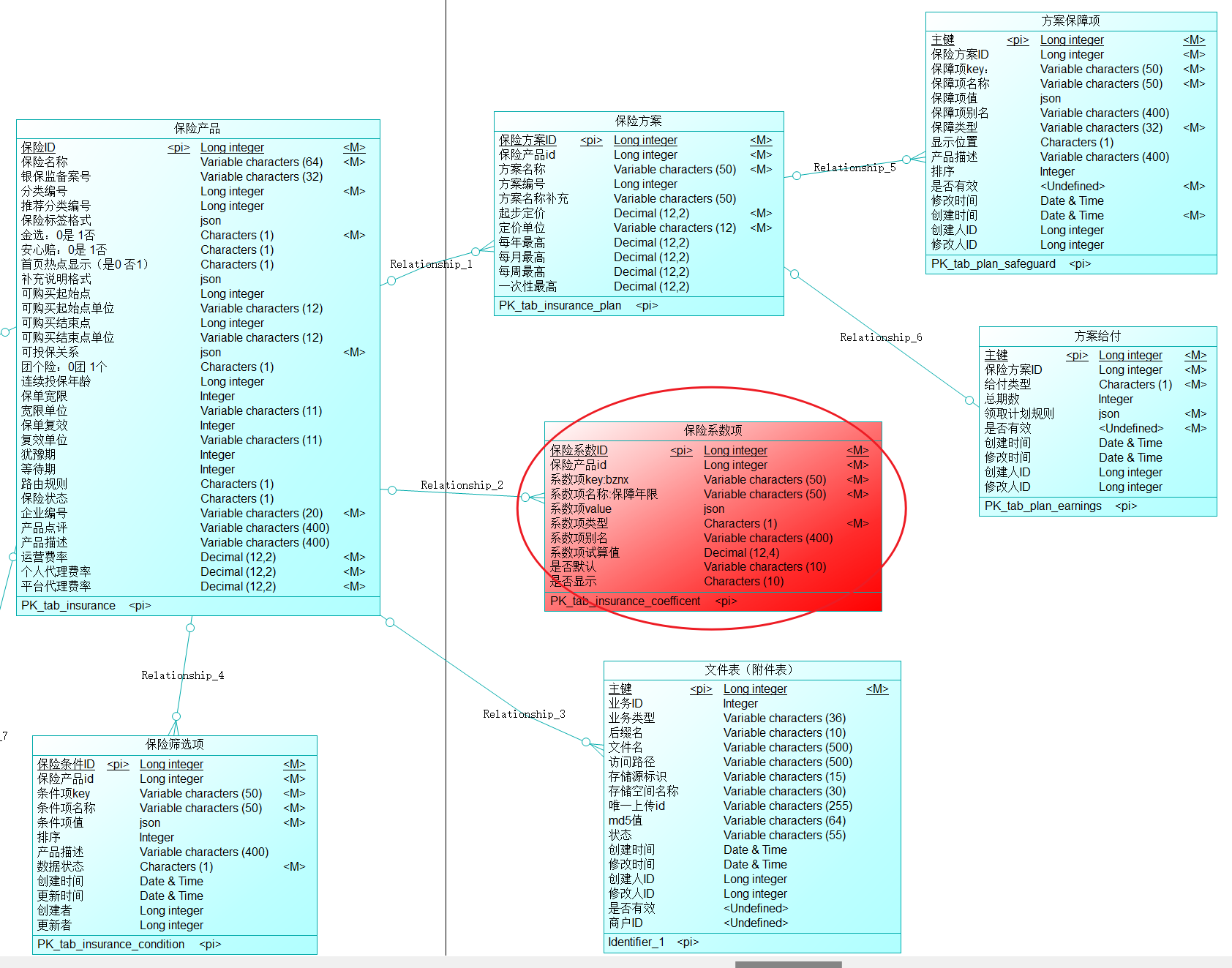
继续刚才的业务；在保存完产品方案之后；跳转到第三步：产品系数项；界面如下：



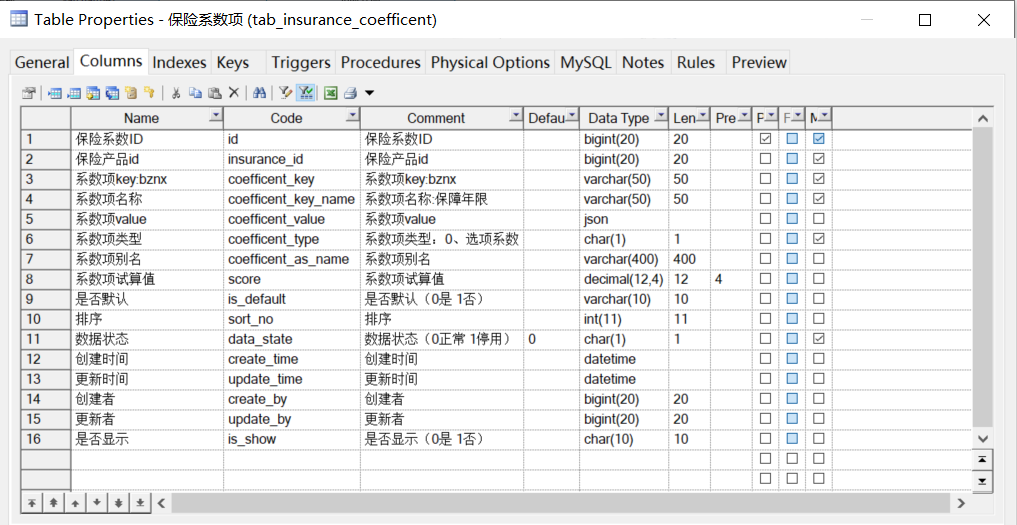
|  |  |
| --- | --- |
| 输入项 | 说明 |
| **保险系数** | 系数名称；字符串；为了以后通过系数查询方便，也同时需要记录系数编号。  小窍门：基本上出现下拉框的时候，一般都是记录对应的id或者编号，为了方便查询可以记录中文选项值 |
| **保险系数别名** | 系数对应显示时候的名字；字符串 |
| **是否展示** | 是否要在详情页面中展示这个系数；字符类型，0是，1否 |
| **系数值** | 属于某个系数项对应的值；类型不定，数量不定；适合将其及试算值、是否默认 保存的一个json中；方便扩展，结构灵活 |
| **系数试算值** | 在产品详情页面中，选择了某个系数值的时候，对应保险价格要乘于这个试算值才是最终的保险价格  浮点类型 |
| **是否默认** | 在产品详情页面中，展示系数项的时候；当前这个系数项值是否为默认选中  字符类型；0是1否 |
| **其它公共内容** | 状态、创建/更新人、时间等 |

**4.2、数据库表设计**

按照界面中分析；保存 保险系数项，需要设计一个独立的表保存；那么保险系数项在整个保险产品中的关系概念图如下：



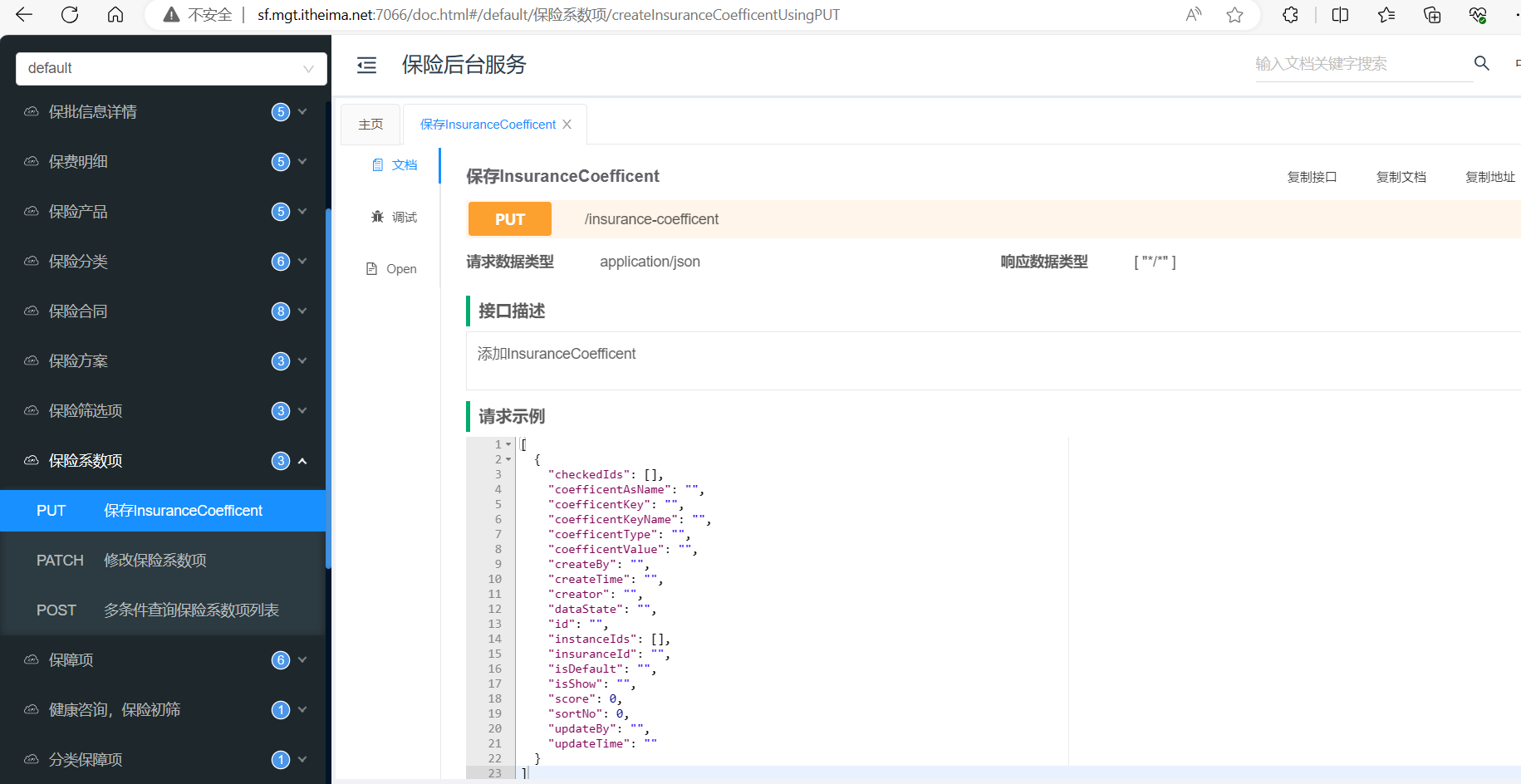
保险系数项的具体数据库表结构如下：



**4.3、接口分析设计**

保存 保险系数项的接口地址：

http://sf.mgt.itheima.net:7066/doc.html#/default/%E4%BF%9D%E9%99%A9%E7%B3%BB%E6%95%B0%E9%A1%B9/createInsuranceCoefficentUsingPUT



在测试时；提交到后台的数据结构参考如下：

|  |
| --- |
| JSON [  {  "coefficentKey": "XS010",  "coefficentKeyName": "自动续保",  "isShow": "0",  "coefficentAsName": "自动续保",  "coefficentType": "0",  "activeKeyOptions": [  {  "val": "0",  "name": "是"  },  {  "val": "1",  "name": "否"  }  ],  "coefficentValue": "{\"val\":\"0\",\"name\":\"是\"}",  "score": "0.9",  "isDefault": "1",  "insuranceId": "1791431636143906817"  },  {  "coefficentKey": "XS010",  "coefficentKeyName": "自动续保",  "isShow": "0",  "coefficentAsName": "自动续保",  "coefficentType": "0",  "activeKeyOptions": [  {  "val": "0",  "name": "是"  },  {  "val": "1",  "name": "否"  }  ],  "coefficentValue": "{\"val\":\"1\",\"name\":\"否\"}",  "score": "1",  "isDefault": "1",  "insuranceId": "1791431636143906817"  },  {  "coefficentKey": "XS008",  "coefficentKeyName": "投入时长",  "isShow": "0",  "coefficentAsName": "投保时长",  "coefficentType": "2",  "activeKeyOptions": [  {  "val": "15",  "name": "15年",  "unit": "YEAR",  "calculatedVal": "15"  },  {  "val": "10",  "name": "10年",  "unit": "YEAR",  "calculatedVal": "10"  },  {  "val": "5",  "name": "5年",  "unit": "YEAR",  "calculatedVal": "5"  },  {  "val": "1",  "name": "1年",  "unit": "YEAR",  "calculatedVal": "1"  },  {  "val": "3",  "name": "3年",  "unit": "YEAR",  "calculatedVal": "3"  },  {  "val": "5",  "name": "5年",  "unit": "YEAR",  "calculatedVal": "5"  },  {  "val": "8",  "name": "8年",  "unit": "YEAR",  "calculatedVal": "8"  },  {  "val": "65",  "name": "65岁",  "unit": "AGE",  "calculatedVal": "65"  }  ],  "coefficentValue": "{\"val\":\"1\",\"name\":\"1年\",\"unit\":\"YEAR\",\"calculatedVal\":\"1\"}",  "score": "1",  "isDefault": "1",  "insuranceId": "1791431636143906817"  },  {  "coefficentKey": "XS008",  "coefficentKeyName": "投入时长",  "isShow": "0",  "coefficentAsName": "投保时长",  "coefficentType": "2",  "activeKeyOptions": [  {  "val": "15",  "name": "15年",  "unit": "YEAR",  "calculatedVal": "15"  },  {  "val": "10",  "name": "10年",  "unit": "YEAR",  "calculatedVal": "10"  },  {  "val": "5",  "name": "5年",  "unit": "YEAR",  "calculatedVal": "5"  },  {  "val": "1",  "name": "1年",  "unit": "YEAR",  "calculatedVal": "1"  },  {  "val": "3",  "name": "3年",  "unit": "YEAR",  "calculatedVal": "3"  },  {  "val": "5",  "name": "5年",  "unit": "YEAR",  "calculatedVal": "5"  },  {  "val": "8",  "name": "8年",  "unit": "YEAR",  "calculatedVal": "8"  },  {  "val": "65",  "name": "65岁",  "unit": "AGE",  "calculatedVal": "65"  }  ],  "coefficentValue": "{\"val\":\"5\",\"name\":\"5年\",\"unit\":\"YEAR\",\"calculatedVal\":\"5\"}",  "score": "0.9",  "isDefault": "0",  "insuranceId": "1791431636143906817"  },  {  "coefficentKey": "XS006",  "coefficentKeyName": "保障期限",  "isShow": "0",  "coefficentAsName": "保障期限",  "coefficentType": "2",  "activeKeyOptions": [  {  "val": "20",  "name": "20年",  "unit": "YEAR",  "calculatedVal": "23"  },  {  "val": "30",  "name": "30年",  "unit": "YEAR",  "calculatedVal": "30"  },  {  "val": "15",  "name": "15年",  "unit": "YEAR",  "calculatedVal": "15"  },  {  "val": "10",  "name": "10年",  "unit": "YEAR",  "calculatedVal": "10"  },  {  "val": "5",  "name": "5年",  "unit": "YEAR",  "calculatedVal": "5"  },  {  "val": "1",  "name": "1年",  "unit": "YEAR",  "calculatedVal": "1"  },  {  "val": "2",  "name": "保2天",  "unit": "DAY",  "calculatedVal": "2"  },  {  "val": "7",  "name": "保7天",  "unit": "DAY",  "calculatedVal": "7"  },  {  "val": "15",  "name": "保15天",  "unit": "DAY",  "calculatedVal": "15"  },  {  "val": "30",  "name": "保30天",  "unit": "DAY",  "calculatedVal": "30"  }  ],  "coefficentValue": "{\"val\":\"10\",\"name\":\"10年\",\"unit\":\"YEAR\",\"calculatedVal\":\"10\"}",  "score": "1",  "isDefault": "1",  "insuranceId": "1791431636143906817"  },  {  "coefficentKey": "XS006",  "coefficentKeyName": "保障期限",  "isShow": "0",  "coefficentAsName": "保障期限",  "coefficentType": "2",  "activeKeyOptions": [  {  "val": "20",  "name": "20年",  "unit": "YEAR",  "calculatedVal": "23"  },  {  "val": "30",  "name": "30年",  "unit": "YEAR",  "calculatedVal": "30"  },  {  "val": "15",  "name": "15年",  "unit": "YEAR",  "calculatedVal": "15"  },  {  "val": "10",  "name": "10年",  "unit": "YEAR",  "calculatedVal": "10"  },  {  "val": "5",  "name": "5年",  "unit": "YEAR",  "calculatedVal": "5"  },  {  "val": "1",  "name": "1年",  "unit": "YEAR",  "calculatedVal": "1"  },  {  "val": "2",  "name": "保2天",  "unit": "DAY",  "calculatedVal": "2"  },  {  "val": "7",  "name": "保7天",  "unit": "DAY",  "calculatedVal": "7"  },  {  "val": "15",  "name": "保15天",  "unit": "DAY",  "calculatedVal": "15"  },  {  "val": "30",  "name": "保30天",  "unit": "DAY",  "calculatedVal": "30"  }  ],  "coefficentValue": "{\"val\":\"20\",\"name\":\"20年\",\"unit\":\"YEAR\",\"calculatedVal\":\"23\"}",  "score": "0.9",  "isDefault": "0",  "insuranceId": "1791431636143906817"  },  {  "coefficentKey": "XS003",  "coefficentKeyName": "付款方式",  "isShow": "0",  "coefficentAsName": "付款方式",  "coefficentType": "0",  "activeKeyOptions": [  {  "val": "MONTH",  "name": "按月支付",  "calculatedVal": ""  },  {  "val": "YEAR",  "name": "按年支付",  "calculatedVal": ""  }  ],  "coefficentValue": "{\"val\":\"MONTH\",\"name\":\"按月支付\",\"calculatedVal\":\"\"}",  "score": "1",  "isDefault": "1",  "insuranceId": "1791431636143906817"  },  {  "coefficentKey": "XS003",  "coefficentKeyName": "付款方式",  "isShow": "0",  "coefficentAsName": "付款方式",  "coefficentType": "0",  "activeKeyOptions": [  {  "val": "MONTH",  "name": "按月支付",  "calculatedVal": ""  },  {  "val": "YEAR",  "name": "按年支付",  "calculatedVal": ""  }  ],  "coefficentValue": "{\"val\":\"YEAR\",\"name\":\"按年支付\",\"calculatedVal\":\"\"}",  "score": "0.8",  "isDefault": "0",  "insuranceId": "1791431636143906817"  } ] |

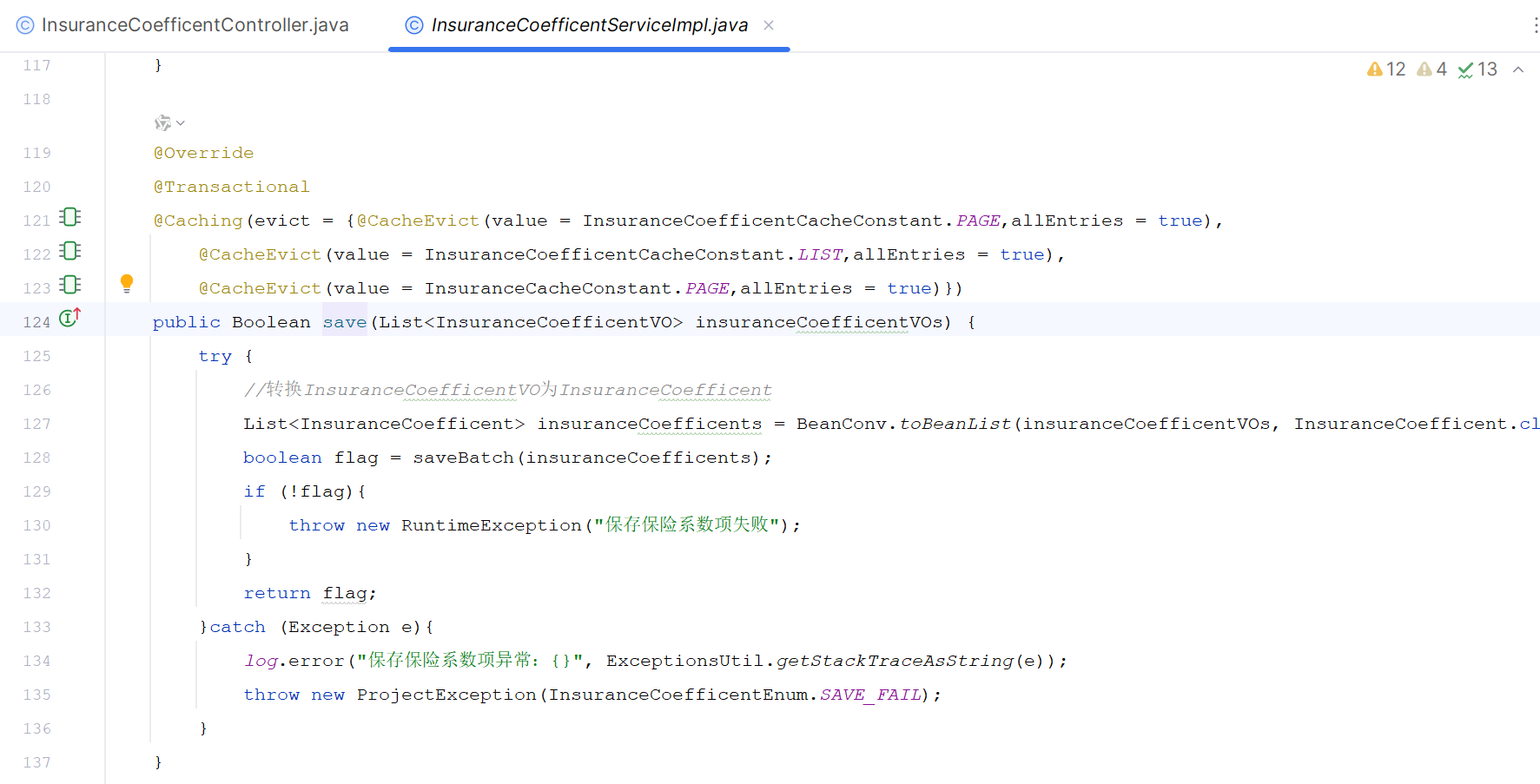
**4.4、开发实现**

由于较为基础；这里代码带着大家查阅如下：

处理器：



业务实现：



**4.5、测试**

在如下的界面中点击 下一步 的时候；如能跳转到第四步 产品筛选项 说明保存产品系数项成功。



点击 下一步 之后；再次查看这个产品，产品系数项如果有数据的话说明保存成功。

**5、作业-产品筛选项**

**5.1、需求分析**

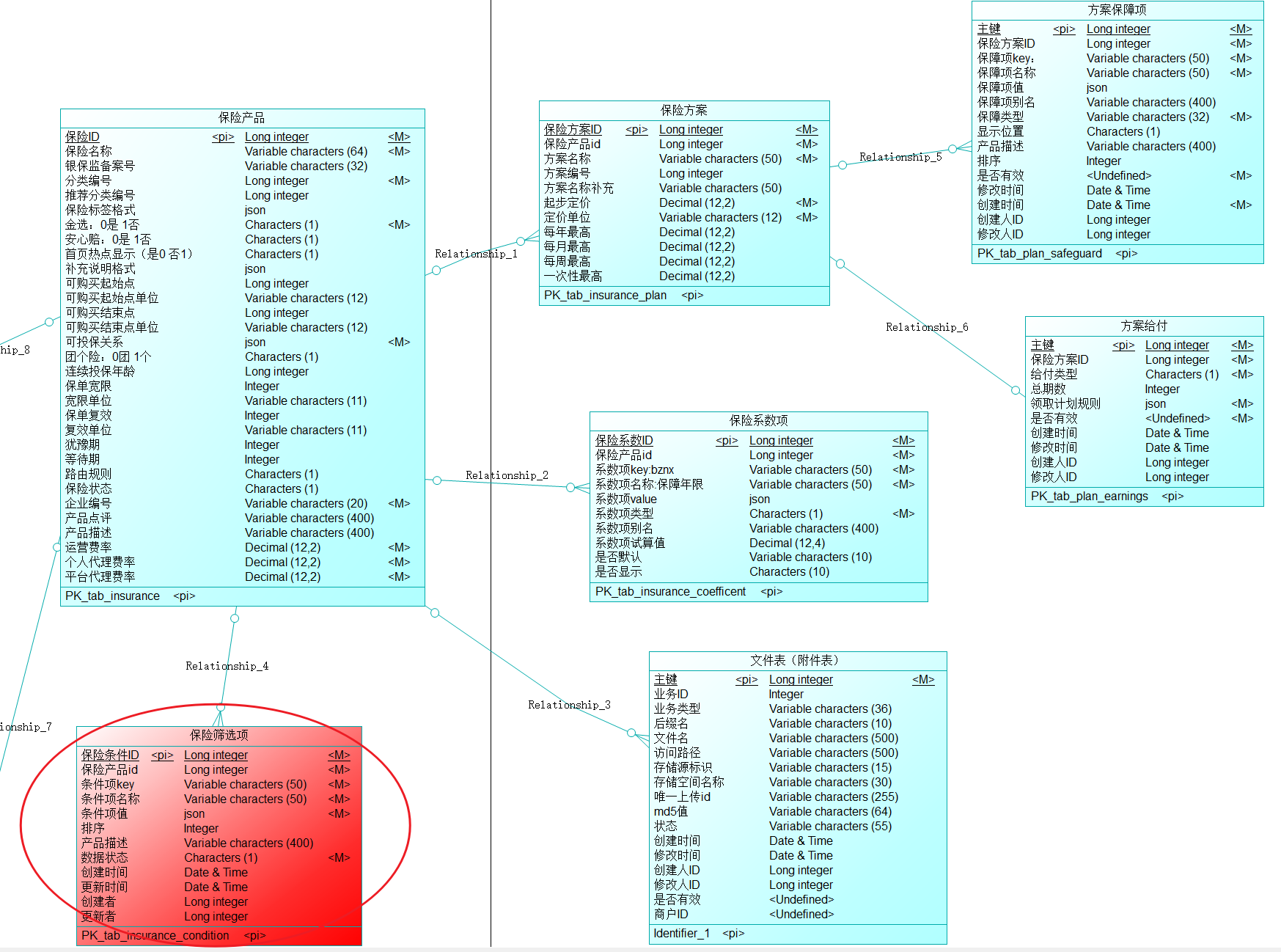
继续课上的业务；在保存完产品系数项之后；跳转到第四步：产品筛选项；界面如下：



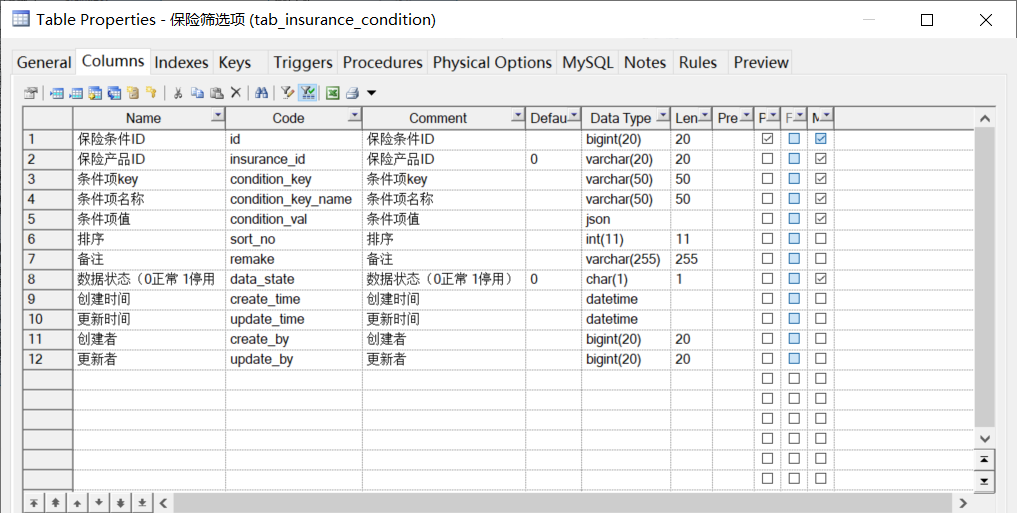
|  |  |
| --- | --- |
| 输入项 | 说明 |
| **条件筛选项** | 名称；字符串；为了以后通过系数查询方便，也同时需要记录系数编号。  小窍门：基本上出现下拉框的时候，一般都是记录对应的id或者编号，为了方便查询可以记录中文选项值 |
| **系数值** | 属于某个系数项对应的值；类型不定，数量不定；适合将其及试算值、是否默认保存的一个json中；方便扩展，结构灵活 |
| **筛选项值** | 在APP端可以通过某些筛选值找到这个产品；那么一个产品可以通过好多筛选项的值都能找到的话；这个值又是不固定的；比如：为谁买；可以在选择本人或者配偶等都可以找到。 所以存储的也应该是结构数量不固定的，类型可以为json；存储的值例如：["BZ", "ZN", "PO", "FM"]  json |
| **其它公共内容** | 状态、创建/更新人、时间等 |

**5.2、数据库表设计**

按照界面中分析；保存 保险筛选项，需要设计一个独立的表保存；那么保险筛选项在整个保险产品中的关系概念图如下：



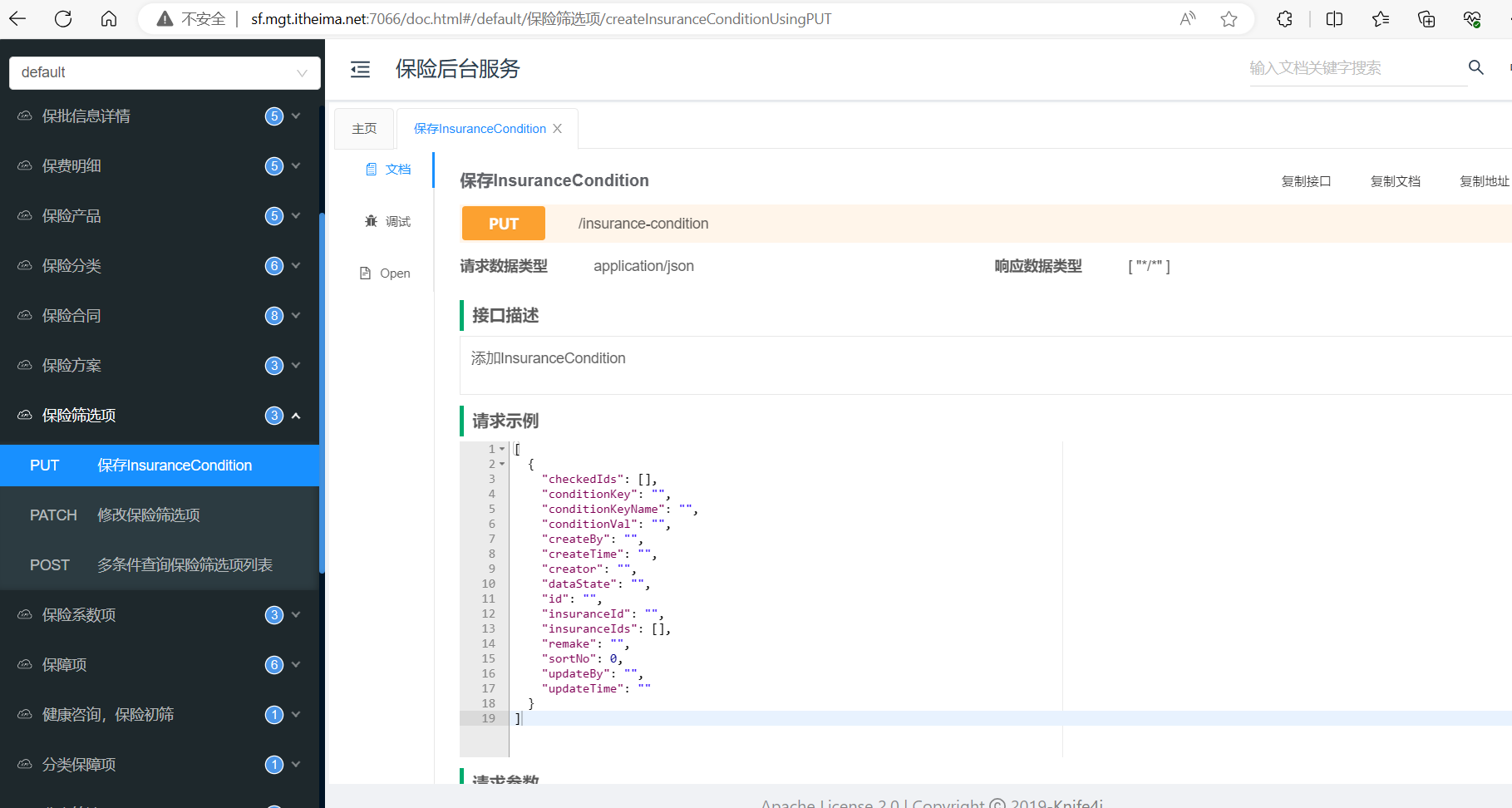
保险筛选项的具体数据库表结构如下：



**5.3、接口分析设计**

保存 保险筛选项的接口地址：

http://sf.mgt.itheima.net:7066/doc.html#/default/%E4%BF%9D%E9%99%A9%E7%AD%9B%E9%80%89%E9%A1%B9/createInsuranceConditionUsingPUT



在测试时；提交到后台的数据结构参考如下：

|  |
| --- |
| JSON [  {  "createTime": "2023-11-07 16:04:42",  "updateTime": "2023-11-07 16:04:42",  "createBy": "1371500419615895553",  "updateBy": "1371500419615895553",  "dataState": "0",  "creator": null,  "categoryNo": "100001002002000",  "conditionKey": "SX002",  "conditionKeyName": "健康状况",  "sortNo": null,  "remake": null,  "checkedIds": null,  "conditionOptions": [  {  "val": "SX00201",  "name": "有其他慢性病"  },  {  "val": "SX00202",  "name": "健康"  },  {  "val": "SX00203",  "name": "有糖尿病"  },  {  "val": "SX00204",  "name": "有三高"  },  {  "val": "SX00205",  "name": "有心脑血管疾病"  }  ],  "conditionValList": [  "SX00201"  ],  "conditionVal": "[\"SX00201\"]",  "insuranceId": "1791431636143906817"  },  {  "createTime": "2023-11-07 16:04:42",  "updateTime": "2023-11-07 16:04:42",  "createBy": "1371500419615895553",  "updateBy": "1371500419615895553",  "dataState": "0",  "creator": null,  "categoryNo": "100001002002000",  "conditionKey": "SX006",  "conditionKeyName": "保多久",  "sortNo": null,  "remake": null,  "checkedIds": null,  "conditionOptions": [  {  "val": "SX00601",  "name": "1天"  },  {  "val": "SX00602",  "name": "2天"  },  {  "val": "SX00603",  "name": "3天"  },  {  "val": "SX00604",  "name": "4天"  },  {  "val": "SX00605",  "name": "5天"  },  {  "val": "SX00606",  "name": "6天"  },  {  "val": "SX00607",  "name": "7天"  }  ],  "conditionValList": [  "SX00602",  "SX00605"  ],  "conditionVal": "[\"SX00602\",\"SX00605\"]",  "insuranceId": "1791431636143906817"  },  {  "createTime": "2023-11-07 16:04:42",  "updateTime": "2023-11-07 16:04:42",  "createBy": "1371500419615895553",  "updateBy": "1371500419615895553",  "dataState": "0",  "creator": null,  "categoryNo": "100001002002000",  "conditionKey": "SX001",  "conditionKeyName": "为谁买",  "sortNo": null,  "remake": null,  "checkedIds": null,  "conditionOptions": [  {  "val": "ZN",  "name": "子女"  },  {  "val": "BZ",  "name": "本人"  },  {  "val": "PO",  "name": "配偶"  },  {  "val": "FM",  "name": "父母"  }  ],  "conditionValList": [  "ZN",  "BZ"  ],  "conditionVal": "[\"ZN\",\"BZ\"]",  "insuranceId": "1791431636143906817"  } ] |

**5.4、开发实现**

**5.4.1、处理器**

处理器部分已经实现如下：



**5.4.2、业务实现**

业务代码需要大家完成（重写业务代码）。比较基础的批量保存代码；做完之后最重要的还是需要梳理一整个保险产品的概念模型；CDM模型搞清楚了那今天也就通关了。加油！

**5.5、测试**

在如下的界面中点击 下一步 的时候；如能跳转到第五步 产品附件 说明保存产品筛选项项成功。



点击 下一步 之后；再次查看这个产品，产品筛选项如果有数据的话说明保存成功。

**6、午间问题**

* 能说一下你开发的保险产品的业务流程吗
* 什么叫 “分步式表单”或者“向导式界面”？
* 保险产品的组成有什么
* 保险产品与方案的关系；保险方案是怎么样形成的？
* 保险产品的系数有什么作用？那筛选项呢？
* **简历职责-候选**：独立完成保险产品的业务分析、关联数据库表设计，分步表单的方式实现保险产品基本信息、多方案、不同系数、筛选和附件的处理；
* 保险产品的业务流程
* 保险产品涉及到哪些表；有几张
* 分步表单怎么做的
* 开发这个业务功能时遇到什么问题？业务、技术上