|  |
| --- |
|  |
| MDDRequirements |
| Sanjel项目 |
|  |
| ©MetaShare Inc. |
| **9/7/2018** |

|  |
| --- |
| 本文记录在MDD与Sanjel团队协作过程中的需求及输出定义等。 |

目录

[1 引言 1](#_Toc530158021)

[1.1 目的 1](#_Toc530158022)

[2 需求 1](#_Toc530158023)

[2.1 目标 1](#_Toc530158024)

[2.2 建议 1](#_Toc530158025)

[3 输出定义 2](#_Toc530158026)

[3.1 一个可运行的项目 2](#_Toc530158027)

[3.1.1 Solution的结构 2](#_Toc530158028)

[3.2 自动部署一个自定义域名的应用 2](#_Toc530158029)

[自定义页面显示的字段 3](#_Toc530158030)

[3.3 3](#_Toc530158031)

[3.4 生成系统包括的功能 4](#_Toc530158032)

[3.4.1 对MasterData的基本操作 4](#_Toc530158033)

[3.4.2 关联的实现 5](#_Toc530158034)

[3.4.3 版本控制的实现 6](#_Toc530158035)

[3.5 WebApi 7](#_Toc530158036)

# 引言

## 目的

记录在MDD与Sanjel团队协作过程中的需求及输出定义等。

# 需求

## 目标

使用MDD生成一个可运行的项目：MasterDataMangementSystem。在目前MDD生成系统的基础上，需要完成以下功能：

1. 关联的前台实现，（包括聚合关系）——预估60人时
2. 版本控制，需要对所有实体的修改记录进行保存，每个修改的版本都可以找到——因为需要修改CommonCore底层代码，涉及修改地方比较多，预估80人时
3. 提供WebAPI，需要新增按时间戳查询，新增版本控制后需对原来已经实现的WebAPI进行修改——预估24人时
4. 权限控制（可选）
5. 升级.net core 2.0（可选）
6. 日志功能（可选）

其中功能1-3是9月份的目标。

交付日期：9月21日前需要确定下来生成后的数据表结构。月底前完成开发交付测试。

## 建议

1. 域名设置。Metashare的Namespace定义，应只限于MDD本身和Metashare的核心库。生成的应用中的Namespace，不应有Metashare Namespace出现。在创建应用时，应该允许用户输入起始根域，这样用户可以与自己的应用进行结合。

（关于域名。在需求文档2.2中我们讨论的域名对应的是Namespace，而在3.2中你把它变成了互联网域名，对应的英文是Domain Name。基于这个歧义，我把文档中的Namespace意思的“域名“改为“命名空间”，请更新相应的规范文档。）——已完成

1. 版本控制的系统属性命名，可以参考ESERVICE中的相关命名，并在此基础上进行添加——已完成
2. 目标应用接口，可以参照Eservice中最终对reference data的使用，提供合理的集成路径。——已完成
3. MDD中一定有许多命字的生成是按你们已经存在的规范生成的，如表名，字段名，类名，属性名等。可能与EService中的命名不完全一致。对于这种情况，请提出来讨论，不要为了迎合Eservice的命名规范而自设障碍，增加实现难度。
4. 在高层需求中的第8项：Logging framework implementation to save logging data in separate database.- 日志框架的实现，以在单独的数据库中保存日志数据。你们应当认真考虑尽快加入到你们架构中去。因为MDD生成的应用，如果没有详细的日志，客户应用中出了问题会很难诊断。出了严重问题，对公司声誉的影响而产生的损失会远大于向用户收取的费用。

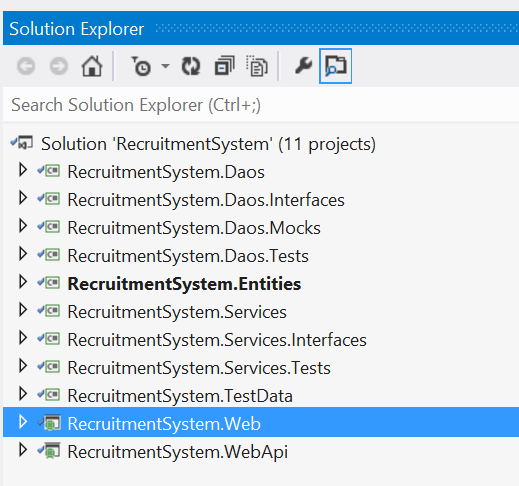
# 输出定义

以下以BlendRecipe、BlendChemical、Product为例进行说明

## 一个可运行的项目

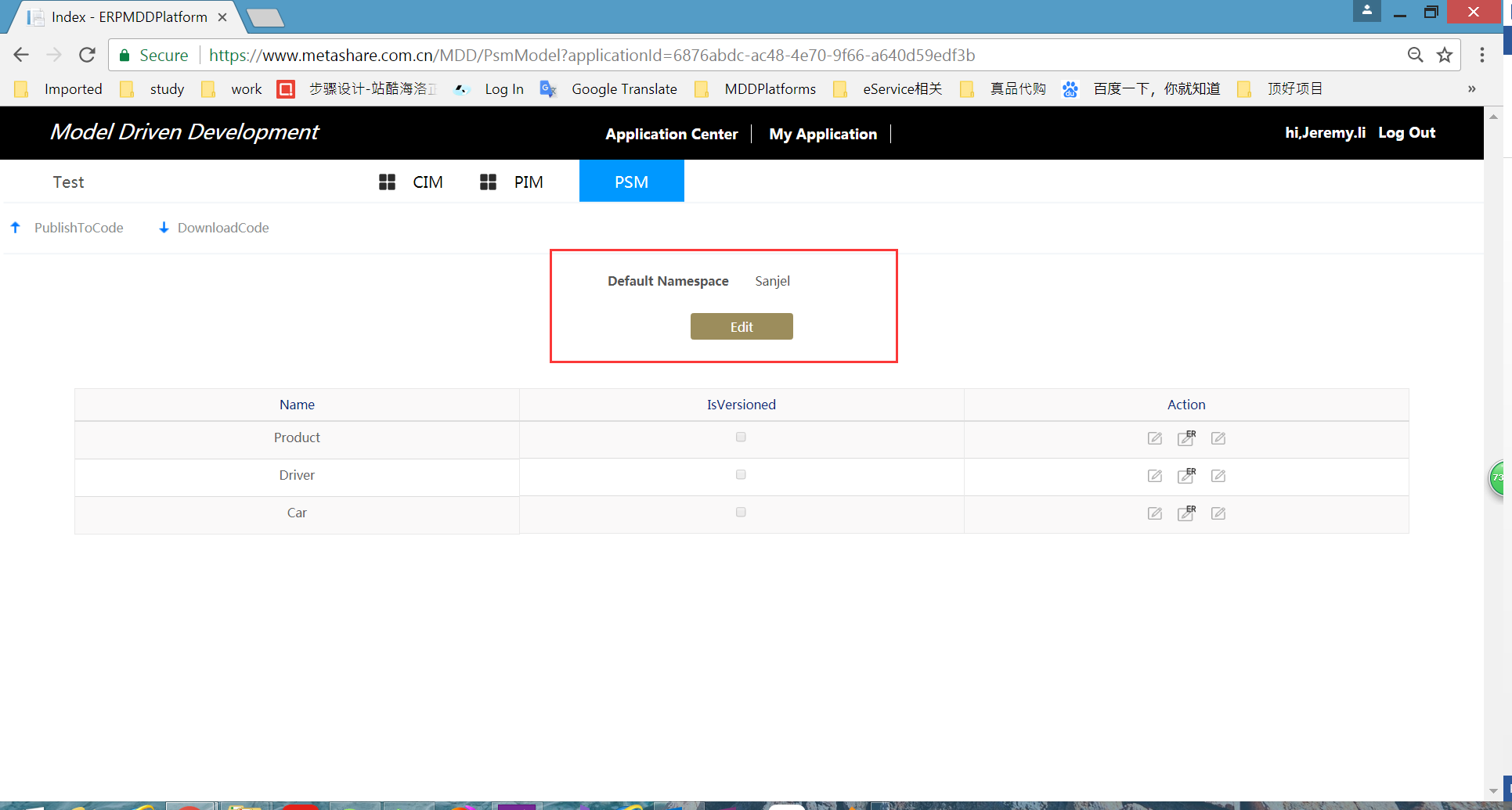
### Solution的结构

生成的项目结构 ：



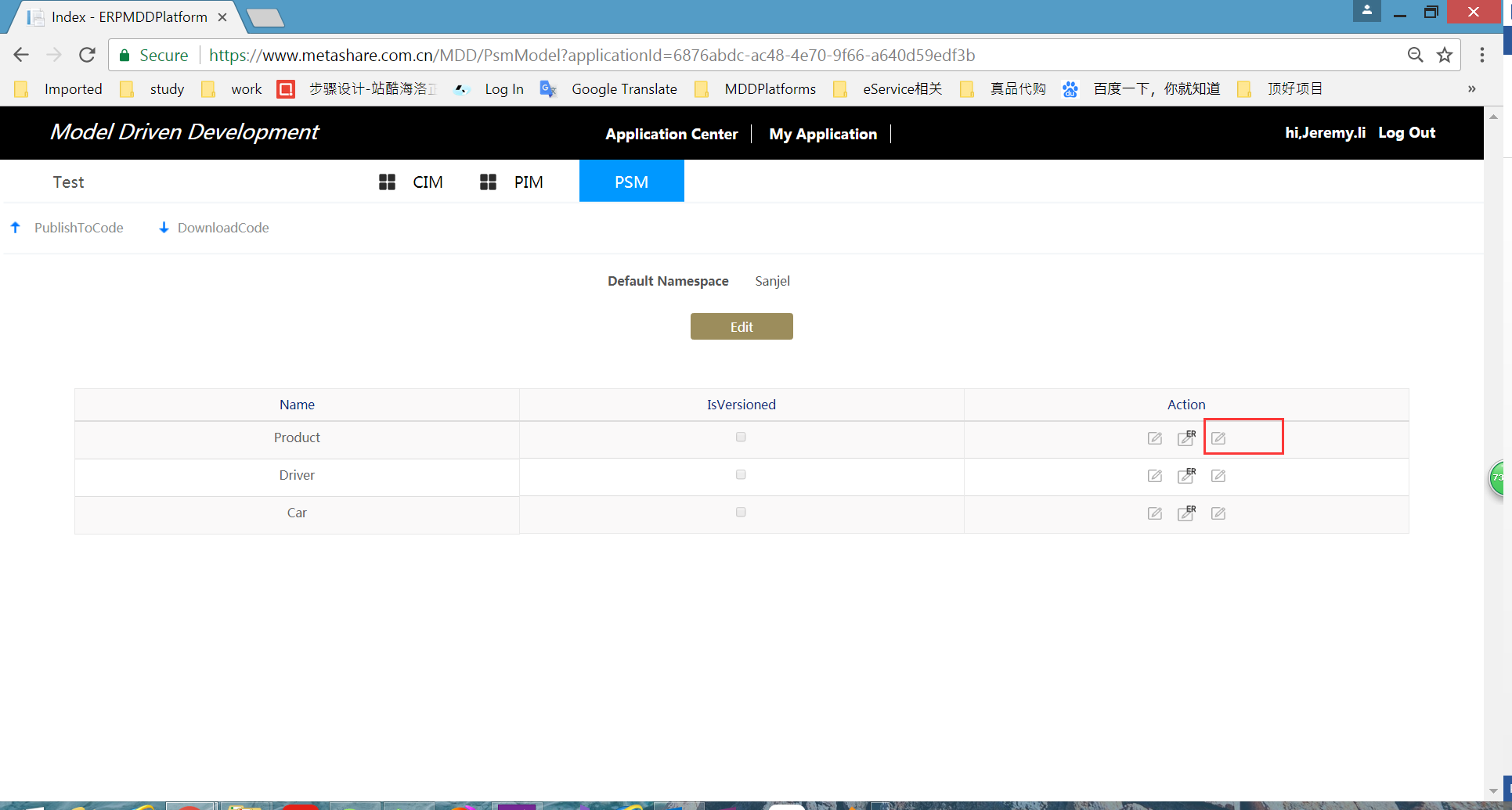
## 自动部署一个自定义域名的应用

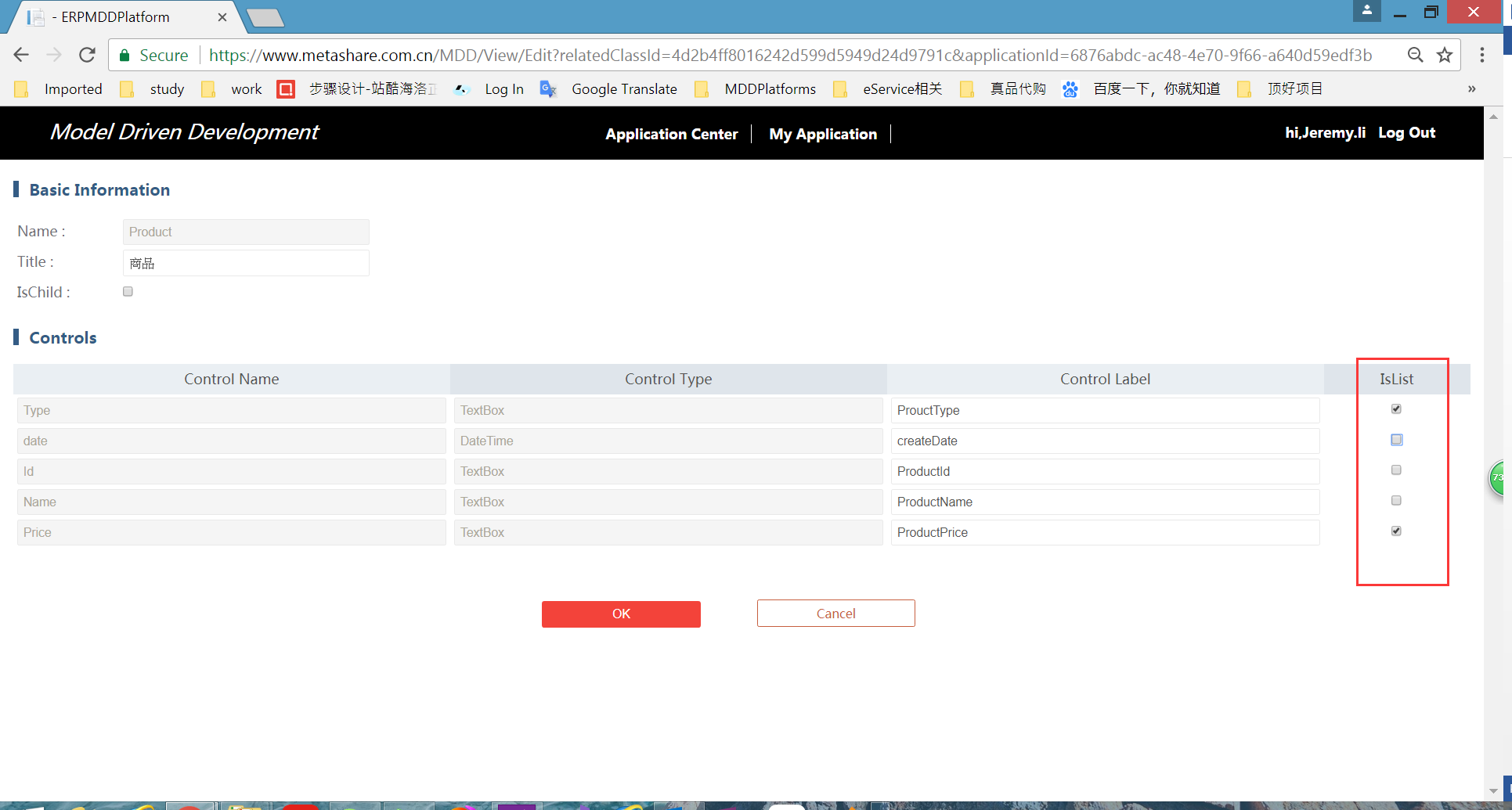
1. 在MDD系统中新建一个应用，用户自定义Namespace。



## 自定义页面显示的字段

在MDD系统中，编辑PSM页面，可以设置页面显示哪些字段



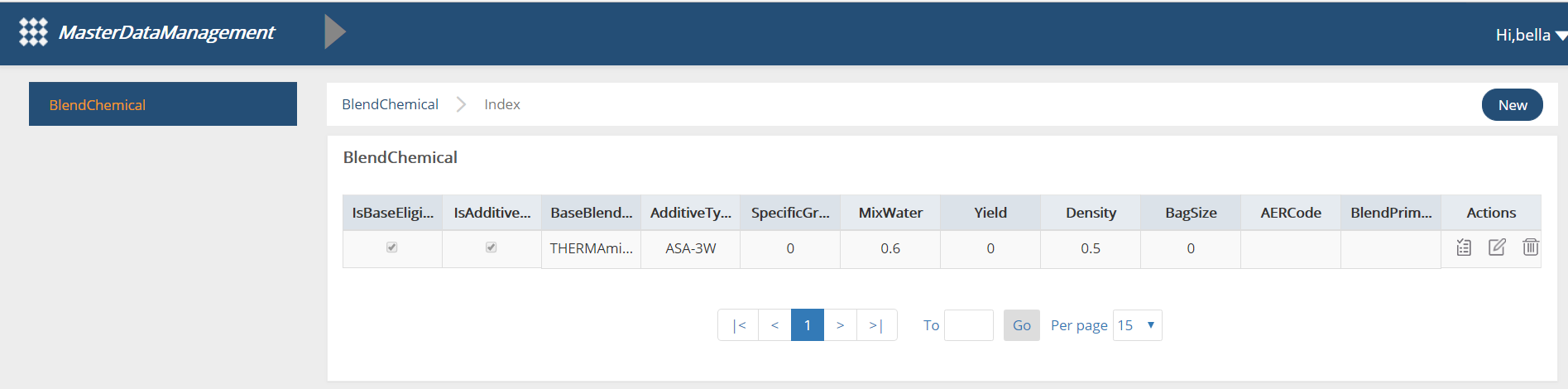


## 生成系统包括的功能

从管理界面上讲CRUD与没有版本的是一样的。版本信息可以显示出来，但它们是只读的。后面会多一个查看历史的按钮。查看历史页面显示一个实体的修改记录列表。可以继续点击一条历史记录显示记录的详细信息，但是只读的。

### 对MasterData的基本操作

以下功能的前台页面与目前MDD生成系统的页面基本一致，Index页面上需要加入版本控制的部分字段。每个对象只对最后一个版本操作，每次操作会产生一个新版本。



1. 增加

每次增加一个新的对象进行操作，操作完成后会新增一个新对象的第一个版本。

1. 删除

每次只对最后一个版本进行操作，操作完成后会新增一个版本，且该版本的失效时期是当前更新时间。思考：这样无法区分失效和删除两个状态，EntityStatus，应当纳入MDD 的考虑。

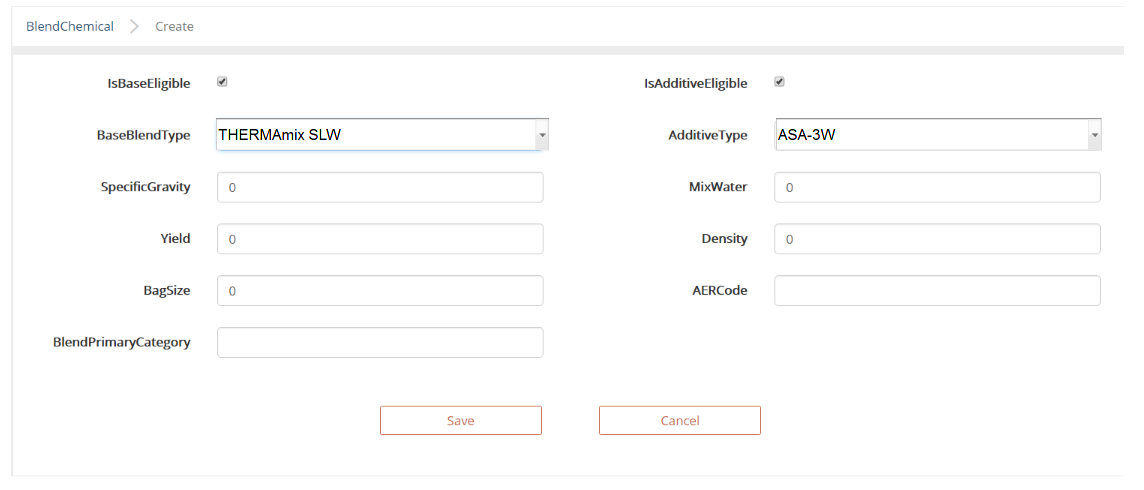
1. 修改

每次只对最后一个版本进行操作，操作完成后会新增一个版本。修改前的最后版本的失效时间是当前更新时间，新增版本生效时间是当前更新时间。

1. 查看
2. 查看历史（具体实现参考3.3.3）

### 关联的实现

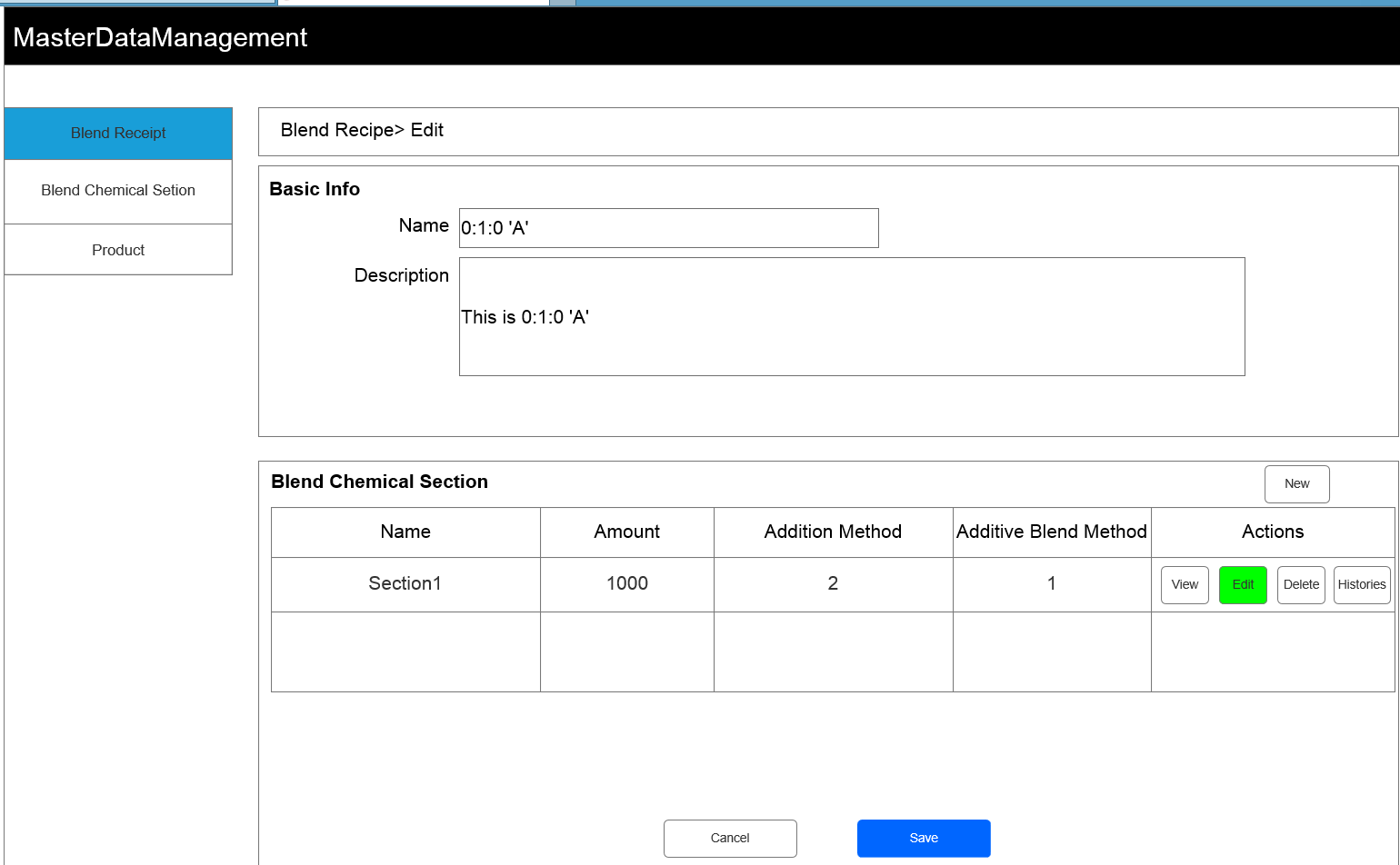
1. 引用数据用下拉框的形式展示



1. 聚合关系使用父子表的展示方式

UI原型见https://184.69.193.254:9880/SanjelDocuments/trunk/Projects/R-Requirements/Sanjel Master Data Management/UI Prototype

对于组合关系：子表的编辑只能在父表编辑页面进行，不能通过左侧菜单进入编辑页面



### 版本控制的实现

实体本身和版本信息是分开的，关联到一起。在使用实体时是没有版本信息的。这一点去参考eService。但是注意这次做的版本控制比eService 更进一步。

#### 问题

1. 组合对象的版本控制如何处理？
2. 系统ID的意义是什么

答案：必须用int的id，systemid是变的，ID或者说是EntityId是不变的，也就是说一个记录，都有相同entityId，他不同的版本会存在于一张表中。

1. MDD目前生成项目的实体ID是string类型，是否满足需求？

答案：一定要是int型。

#### 功能描述

1. 进入BlendChemical的Index页面，显示系统中所有的BlendChemical，在每一条数据后面新增查看历史的操作按钮，点击后进入查看历史版本页面，显示所有的修改记录

#### 实体定义

目前还是用一个类来保存实体和其版本信息。需要添加与版本控制相关的字段：

需要添加与版本控制相关的字段（待完善）：

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| SystemId | int |  |
| Version | int |  |
| ModifiedUserId | string |  |
| ModifiedUserName | string |  |
| ModifiedDateTime | DateTime |  |
| OperationType | DataOperationType |  |
| EffectiveStartDateTime | DateTime |  |
| EffectiveEndDateTime | DateTime |  |
| IsExpired | bool | 是不是做为Status的一个值更好？ |
| IsEffective | bool | 是计算属性 |
|  |  |  |
|  |  |  |

#### 锁机制

多个用户同时对同一条MasterData进行修改时的处理

## WebApi

访问接口也与正常是一样的。如GetById(Id)它返回当前有效的值。GetById(Id,timestamp)返回指时间有效的值。

RequestResult SelectAll()

RequestResult SelectById(int Id)

RequestResult SelectBy(Entity entity)

RequestResult SelectBy(int Id , DateTime timestamp)

RequestResult Insert(Entity entity)

RequestResult Update(Entity entity)

RequestResult Delete(Entity entity)

public class RequestResult

{

public bool IsSucceed { get; set; }

public string Message { get; set; }

public object Data { get; set; }