

EOVA 快速开发

Version : 1.5

EOVA 官方网站: <http://www.eova.cn> (建设中)

EOVA 官方演示: <http://demo.eova.cn>

EOVA 官方 QQ 群: 424665115

Easy Object View Admin

修订历史记录

日期	版本	描述	作者
2015-01-01	V1.0	体验版	Jieven
2015-05-01	V1.1	重构 EovaUI	Jieven
2015-06-01	V1.2	新增导入导出，完善 View	Jieven
2015-10-01	V1.4	重构版，基础功能完整版	Jieven
2016-01-01	V1.5	相对最稳定版，没有最好只有更好	Jieven

目录

前言.....	5
1. EOVA V1.5 更新摘要.....	6
1.1 四大业务模版.....	6
1.2 定时调度管理.....	6
1.3 快速操作.....	6
1.4 图片服务.....	6
1.5 多选配置.....	7
1.6 级联操作.....	7
1.7 导出业务脚本.....	8
2. 有企业在用 Eova 吗?	9
3. Eova 与其它快速开发框架的区别?	10
4. 快速开发平台.....	10
4.1 简介.....	10
4.2 两种模式.....	11
4.3 使用趋势.....	11
5. Eova 初体验:快速完成 CRUD.....	12
5.1 导入元数据.....	12
5.2 新增菜单.....	12
5.3 完成.....	13
6. Eova 核心思想.....	14
7. Eova 核心功能.....	17
7.1 Eova 控件.....	18
7.2 Eova 组件.....	19
7.3 Query(快速查询).....	19
7.4 ToolBar(工具栏).....	19
7.5 Form(表单).....	19
7.6 Grid(数据表格展示).....	20
7.7 TreeGrid(树形数据表格展示).....	20
7.8 Eova 模版.....	21
7.9 SingleGrid: 单表.....	21
7.10 MasterSlaveGrid: 主子表.....	22
7.11 SingleTree:单树表.....	23
7.12 TreeToGrid:树关联表.....	24
8. EOVA DIY-自定义拓展.....	25

8.1 快速编辑元对象.....	25
8.2 快速编辑元字段.....	26
8.3 DIY AOP:服务端业务扩展.....	27
8.3.1 元对象拦截器: MetaObjectIntercept.....	27
8.3.1.1 拦截器配置.....	27
8.3.1.2 拦截器接口.....	27
8.3.1.3 拦截器案例.....	28
8.3.2 单表模版拦截器: SingleIntercept.....	29
8.3.2.1 拦截器配置.....	29
8.3.2.2 拦截器接口.....	29
8.4 DIY JS:客户端业务扩展.....	30
8.4.1.1 菜单 DIY_JS 配置.....	30
8.4.1.2 对象 DIY_JS 配置.....	30
8.4.1.3 DIY_JS 案例.....	30
8.5 DIY CRUD:可分配功能配置.....	31
8.6 DIY Button:添加一个新的功能按钮.....	32
8.7 DIY View:多表视图.....	34
8.8 DIY 完全自定义.....	35
9. EOVA EXP-表达式.....	36
9.1 元字段表达式.....	36
9.1.1 下拉框表达式.....	36
9.1.2 查找框表达式.....	36
9.2 过滤条件表达式.....	37
10. FAQ.....	38
11. 其它资料.....	39
12. 待续.....	40

前言

- Eova 是一个 J2EE 快速开发平台。
- Eova 旨在用简单的方法，来解决常见的各种需求。
- Eova 适用于对 UI 布局 and 性能没有严苛要求的企业级 WEB 后台管理系统！
- 汝之所向，吾之所往，吾定不负汝！

1. EOVA V1.5 更新摘要

1.1 四大业务模版

单表, 单树表(新增), 主子表, 树级联表(新增)

1.2 定时调度管理

管理中心

定时调度 ×

最大化

刷新

新增

修改

删除

启动任务

暂停任务

立即运行一次

	ID	状态	名称	表达式	实现类	说明
1	1	暂停	每分	0 0/1 * * * ?	com.eova.job.EveryMinJob	每分钟来一发
2	2	暂停	每时	0 0 0/1 * * ?	com.eova.job.EveryHourJob	每小时统计一次
3	12	暂停	每天	59 59 23 * * ?	com.eova.job.EveryDayJob	每天23点59分59秒跑一下

1.3 快速操作

快速编辑元数据

快速查询 新增 修改 删除				快速查询
		登录账户		快速查询
1		test3		编辑元字段 编辑元对象 other
2		test4		
3		test5	000000	
4		test6	000000	

1.4 图片服务

Eova 图片上传到独立图片服务器

#图片服务域名

domain_img = http://127.0.0.1:18080

#静态根目录

static_root = D:/Codeing/static

元字段(表: eova_field)JSON 配置(字段: config)

```
{"filedir": "/image/bucao/product"}
```

So, 经过如上配置, 是为了下面两项

图片上传保存目录:

D:/Codeing/static/image/bucao/product/001.jpg

图片服务访问 URL:

http://127.0.0.1:18080/image/bucao/product/001.jpg

1.5 多选配置

多选配置项：元字段编辑 是否有多选项

是否有多项	状态:	正常, 禁用	
<input checked="" type="checkbox"/>			
<input type="checkbox"/>	状态	登录账户	登录
	正常, 禁用, 删除	test3	0000

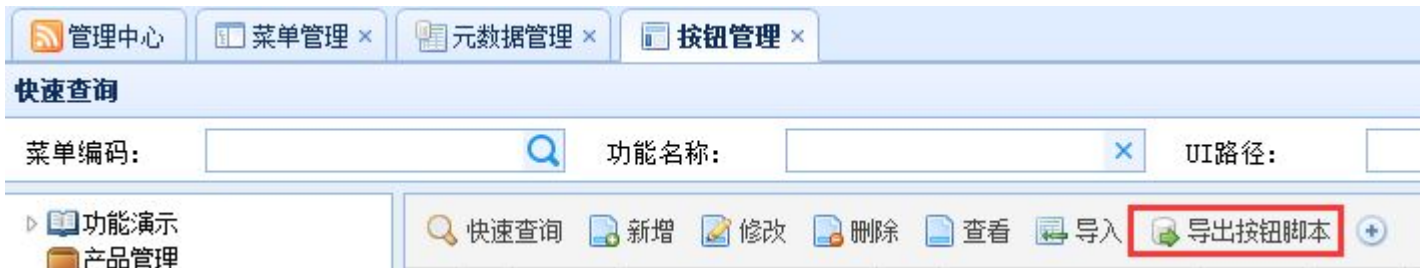
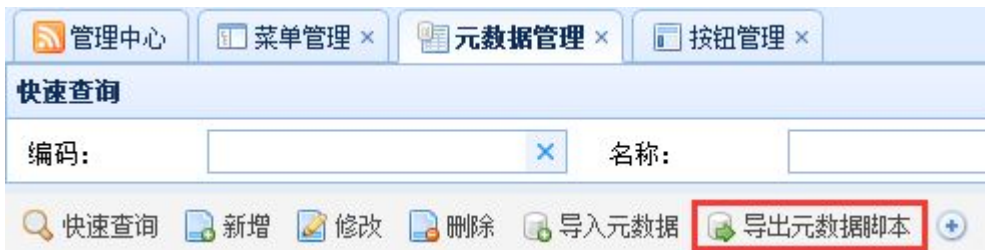
1.6 级联操作

修改酒店

ID:	1	酒店名:	松江锦江之星1	电话:	(021)37621128
商户状态:	签约商户签约商户	积分:	12	创建时间:	2015-09-03 17:17:00
详细地址:	上海市松江区西林北路950号				
省:	上海	市:	上海	区:	闵行区

长宁区
闸北区
闵行区
徐汇区
浦东新区
杨浦区
普陀区
静安区
卢湾区

1.7 导出业务脚本



查看更新的详细内容：

<http://note.youdao.com/share/?id=d3a1606a1b5f6fcf46dff73d4fd43e81&type=note>

2. 有企业在用 Eova 吗？

- 上海电信子公司：电信 OSS 系列产品
- 上海宜信子公司：互联网金融管理系统
- 上海 3GT：网游运营支撑系统
- 山东云游：社区运营系统
- 北京嘉品奇季：酒店布草租赁系统
- Eova 在 GAME，SHOP，OSS，CRM，BBS，CMS 等领域均有广泛应用。

有图有真相：



租赁云平台.png



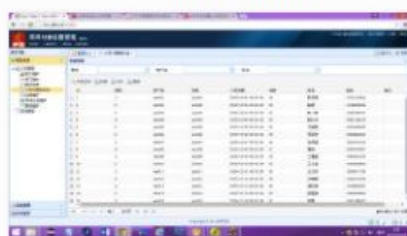
智能营销中心系统.png



微信后台管理系统.png



社交管理系统.jpeg



企业员工管理系统.jpeg



客户关系管理系统.jpeg



进销存管理系统.jpeg



供应链管理平台.png



电商管理系统.png

3. Eova 与其它快速开发框架的区别？

- Eova 使用了极速框架 JFinal ! 没有使用重量级 SSH 等，天生轻量。
- Eova 是引擎模式，不走生成源代码模式。
- Eova 不提供除了权限管理以外的业务功能模块。如：CRM，OA，ERP

但是你用 Eova 能快速个性化定制这些系统，Eova 就是专门用来 DIY 这些系统的神器。

4. 快速开发平台

快速开发平台，就是可以使得开发更为快速的开发平台。当开发平台产生之后，虽然减少了编程人员大量的编程时间，但是很多开发平台的效果并不是很理想，比如说某些开发平台比较复杂、难以掌握；有的开发平台通用性比较差；有的开发平台在时间上并没有得到改善；还有的依然还是需要写很多代码等等。这些问题的存在促使开发者不断的摸索、不断的改进，到最后越做越成熟，以致于现在市面上出现的大部分开发平台效率都非常高，他们改善了以往的产品存在的缺陷，使得开发过程比以往更简洁、编写代码更少、开发效率越来越高。

4.1 简介

当我们最开始接触软件开发的时候，我们大都是采用记事本来编写程序，运用 JDK、MFC 等提供的 API 自己编写代码来完成想要的功能，编写完之后还要编译成可执行的文件，然后再运行。这种方式虽然通俗，但是一点也不方便，慢慢的，编程人员开始寻求比较方便开发的工具，于是诸如 eclipse、jbuilder、VC++ 等等一系列的开发工具便出现在了市面上，这些工具的出现，大大方便了开发人员的编程工作，减少了编程人员很多不必要的麻烦。像包括编译、异常处理、发布、模拟运行等等操作，都可以在这些开发工具上完成。

但是，随着时间的推移，编程人员发现，即使有这么好的开发工具，在开发的过程中，依然要写很多很多的代码，而且仔细的分析来看，很多代码基本上都是重复的编写，功能大同小异。于是，他们便开始琢磨另一种更为方便高效的开发工具，比如说：我们可以将很多重复的代码封装起来，然后需要用到的时候自行调用？或者是我们可以搭出一个基本的开发框架，然后编程人员可以在这个框架的基础上进行二次开发？通过编程人员一次一次的实验，最终形成了一种新的开发工具，那就是开发平台。

开发平台，简单的理解就是：以某种编程语言或者某几种编程语言为基础，开发出来的一个软件，而这软件不是一个最终的软件产品，它是一个二次开发软件框架，用户可以在这个产品上进行各种各样的软件产品的开发，并且在这个产品上进行开发的时候，不需要像以往的编程方式那样编写大量的代码，而是只需要进行一些简单的配置，或者是写极少量的代码便可以完成一个业务系统的开发工作。

4.2 两种模式

市面上的快速开发平台主要分为两种模式。

一种是引擎模式，一种是生成源代码模式。拿报表来举例，所谓引擎模式是指通过报表设计器设计出报表模板，发布到报表引擎中，在运行时，你只需要向报表引擎里传递相关的参数，如报表条件，报表引擎负责查询数据库，加工数据，然后以各种方式展现出来，在这个过程中是不需要开发人员编写代码的，也不产生源代码的。即使是在开发过程中也是如此，利用开发平台开发业务系统时，开发者不需要编码，只需通过 WEB 页面进行参数定制即可，这些参数存放在系统数据库或 XML 文件中。系统运行时，引擎会调用这些参数进行页面展现及业务处理。

另一种便是生成源代码的，这种方式主要通过一个桌面式设计器来定义业务模块，辅助生成源代码框架，然后用户可以在生成的源代码的基础上编写、修改自己的源代码，实现业务逻辑，包括生成、修改 JSP 页面。所以生成源代码模式也可认为是一种代码生成器。这种模式的主要代表是普元平台，这种模式的产品对开发者的要求比较高，但由于面向的对象基本都是软件开发商或者有研发实力的企事业单位，深受政府单位和大中型企业的欢迎。

Eova 就是属于引擎模式的快速开发平台！

4.3 使用趋势

毫无疑问，快速开发平台作为一种新型的开发工具，它的出现减少了代码编写量，简化了开发过程，缩短了开发周期，提高了开发效率，节省了开发成本。越来越多的受到广大开发者的青睐，而且将会有越来越多的开发者使用快速开发平台来完成自己的开发工作。普通用户在接受简单培训后也能够使用快速开发平台 DIY 出自己所需的各种管理软件。

5. Eova 初体验:快速完成 CRUD

PS: 不需要编写任何代码。

5.1 导入元数据

平台维护>对象管理>导入元数据



导入元数据

数据源	main
元类型	Table
元数据表	eova_user
元对象名称	用户信息
元对象编码	user_code

确定

5.2 新增菜单

菜单管理>新增



新功能配置

菜单信息

父级	综合业务	图标	icon-box
名称	我的第一菜单	编码	biz_myuser
序号	1	业务类型	单表[singleGrid]

业务模版配置(单表)

元对象	用户
业务拦截器	

确定

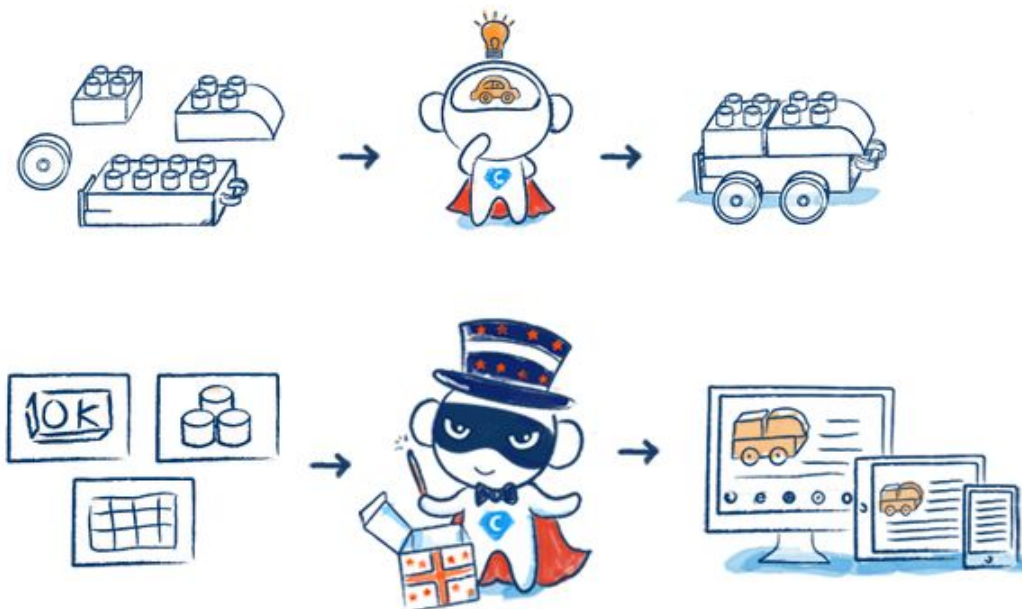
5.3 完成

V1.2 开始会默认将新增菜单分配给超级管理员角色（rid=1），新增(等待 5s 后)刷新即可看到添加的菜单。

权限分配预览：

[超级管理员]可用功能选择		
全选 反选 选择所有查询 选择所有新增 选择所有修改 选择所有删除		
菜单	功能	
1 综合业务		
2 玩家管理		
3 用户信息	<input checked="" type="checkbox"/> 查询	
4 注册信息	<input checked="" type="checkbox"/> 查询	
5 签到	<input checked="" type="checkbox"/> 查询	
6 签到记录	<input checked="" type="checkbox"/> 查询	
7 玩家动态	<input checked="" type="checkbox"/> 查询	
8 玩家信息	<input checked="" type="checkbox"/> 查询	
9 游戏管理		
10 游戏信息	<input checked="" type="checkbox"/> 查询 <input checked="" type="checkbox"/> 新增 <input checked="" type="checkbox"/> 修改 <input checked="" type="checkbox"/> 导出 <input checked="" type="checkbox"/> 导入 <input checked="" type="checkbox"/> 上架	
11 游戏标签	<input checked="" type="checkbox"/> 查询 <input checked="" type="checkbox"/> 新增 <input checked="" type="checkbox"/> 修改 <input checked="" type="checkbox"/> 删除	

6. Eova 核心理念



Eova 是一个基于数据驱动，组件化的开发平台。

Eova 就像一盒乐高，盒子里面提供了各种各样的形状，比如小块积木(各种控件)、大块积木(各种组件)，你可以用这些积木拼出各种你想像中的物体。并且还提供了一些已经拼好的物体(各种模版)。当然盒子里的积木是有限的，当我们要去拼一个奇怪的东西时，会发现现有积木无法拼出，或缺一小块，或多一小块，一步之差非常可惜。这时我们可以对积木进行 DIY 来实现我们的目的(Eova 二次开发：添加新的控件或组件,或者修改业务模版)。

Eova 将常用的控件(文本框，下拉框，时间框...)，组件(表格，树...)，业务(单表 CRUD，多表 CRUD，树表 CRUD...)进行封装抽象，从而实现高度复用。理论上 Eova 中的每种控件，每种组件，每种业务只实现一次，后续直接复用。

Eova 实现原理，通俗的讲和 Navicat 非常相似，可以简单理解为一个 WEB 的数据库维护工具。

但是在 Navicat 的基础上多了一些业务特性，比如：数据校验，级联操作，条件查询什么的，普通用户也能操作。可以友好的进行业务权限分配，有丰富的业务控件。

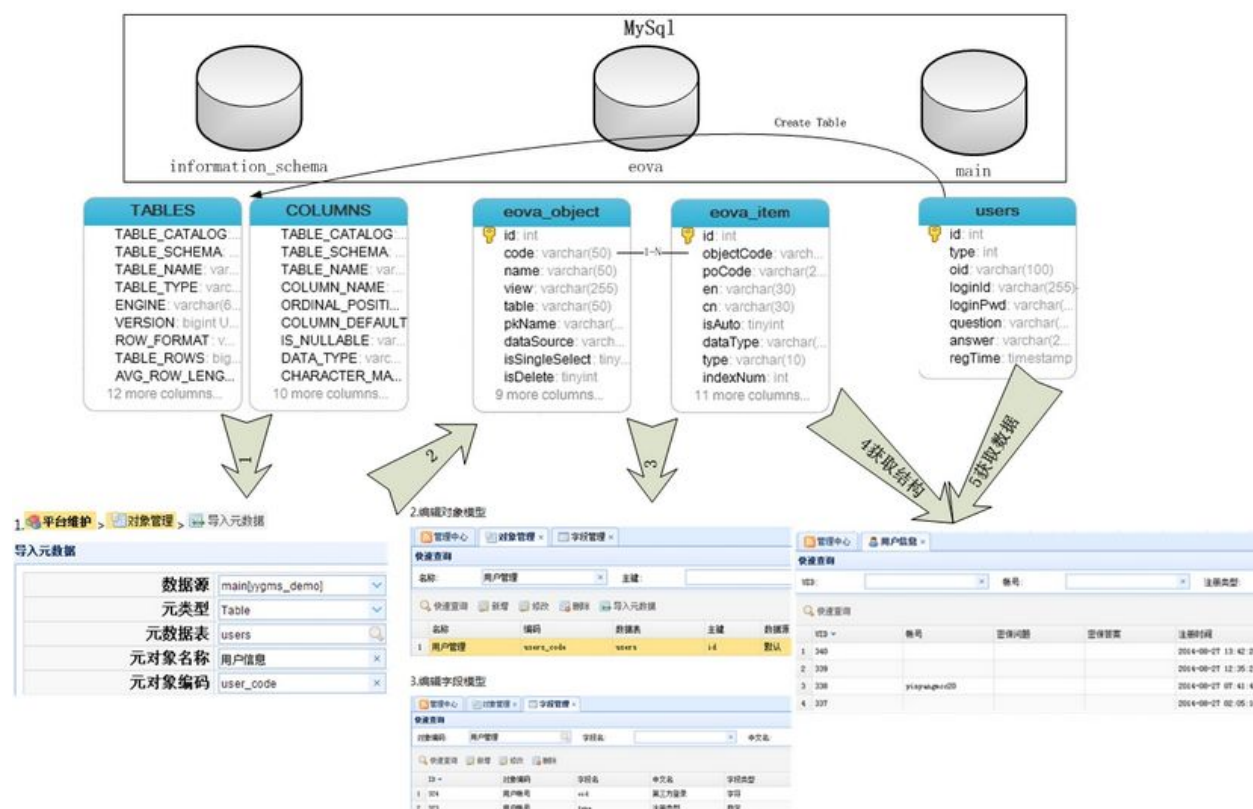
前置知识点：

information_schema.TABLES

information_schema.COLUMNS

如果你不知道 Mysql 的这两张表是干什么的，先去百度搞清楚了再往下看！

Eova核心数据关系



元数据驱动概念：

我们要实现一个功能，首先得在数据库中创建好一张表(数据库会将该表的结构信息存在元数据表中)。

万物皆对象，这张表就是一个对象，所以我们会创建一个对应的实体类与之对应。

然后通过 JDBC 获取到数据，然后使用该对象作为载体存放数据。

最后将对象的数据在页面上显示出来。

那么 Eova 是怎么实现的？

先介绍 Eova 的两张表：

eova_object 对象模型，下文简称 object

eova_field 字段模型，下文简称 field

	UID ▾	玩家ID	昵称	性别
1	340	10839	小白340	女
2	339	10838	狩護佬棍	女
3	338	10837	喜洋洋	女

上图为一个 Grid，我们如果要想实现这样一个 Grid 一般的做法是，将数据查询出来，然后通过循环遍历显示在页面上，并且列名都是在页面上写死的。

Eova 这里没有将列名写死而是将列名保存到了 field 中，然后动态将列名查出来显示。

object 与 field 是一对多关系，object 主要记录了当前对象的信息，通过对象的唯一标识对象编码 (eova_object.code) 可以查到本对象属性的字段名和中文显示名。

综上所述，只需要改变 Grid 的对象编码，就可以形成不同的 Grid 展示。

这就是元数据驱动业务。

建议：如果你看晕了，那么你还是先掌握使用方法，不需深究底层实现，使用熟练使用之后，原理自然知晓。另外你非常想弄懂，可以加入 QQ 群和其它已经理解的同学进行交流！

7. Eova 核心功能

控件 = HTML/CSS+JS

组件 = 多控件组合

模版 = 组件+控件组合

菜单 = 模版<> 绑定对象<>填充数据

自定义 = 尽可能的复用现有功能+手写少量代码<>快速实现需求

开发和使用的中心思想：**尽可能多的使用现成的东西以节省开发量**

优先级	实现方式
1	使用现有模版
2	基于现有模版扩展
3	使用现有组件+控件，自行组装业务
4	完全自定义实现业务

7.1 Eova 控件

文本框	<input type="text"/>
下拉框	客服
查找框	<input type="text"/>
自增框	32
时间框	2015-03-25 00:00:00
日期框	2015-10-13
图标框	<input type="text"/>
图片框	1444791629585.gif
上传框	icon-cartput
复选框	<input checked="" type="checkbox"/>
编辑框	UE 富文本编辑

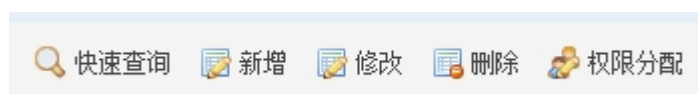
7.2 Eova 组件

7.3 Query(快速查询)

快速查询

菜单编码:	<input type="text"/>		功能名称:	<input type="text"/>	
UI路径:	<input type="text"/>		BS路径:	<input type="text"/>	

7.4 ToolBar(工具栏)



7.5 Form(表单)

修改菜单

ID	110	
名称	访问记录	
编码	biz_uri	
URL	/crud/toList/menuCode=biz_i	
类型	1	
图标	ext-icon-link	
序号	100	
父节点	平台维护	

确定

7.6 Grid(数据表格展示)

	用户 ▲	数值	描述	日志类型	时间
1	雷诺	10		修改签名	2014-08-20 19:54:17
2	枫临天下	10		修改签名	2014-07-25 16:34:27
3	枫临天下	10		修改签名	2014-08-23 09:02:14
4	二蛋	0		修改签名	2014-06-20 22:22:30
5	二蛋	0		领礼包	2014-06-23 17:05:49
6	二蛋	200		领礼包	2014-06-28 18:34:04
7	二蛋	200		领礼包	2014-06-30 19:55:52
8	一笑亦倾城	0		修改签名	2014-06-18 10:00:57
9	一笑亦倾城	0		修改签名	2014-06-20 20:31:07
10	一笑亦倾城	2		领礼包	2014-06-24 10:25:30

15 ▼ | ◀ ▶ | 第 1 共8页 | ▶ ▶ | 显示1到15, 共119记录

7.7 TreeGrid(树形数据表格展示)

	<input type="checkbox"/> 名称	类型	编码	过滤条件
1	<input type="checkbox"/> 功能演示	dir	biz_demo	
2	<input type="checkbox"/> 单表CRUD	single_grid	biz_demo_users	where status =
3	<input type="checkbox"/> 富文本编辑	single_grid	biz_demo_tool	
4	<input type="checkbox"/> 下拉和查找	single_grid	biz_demo_usersitem	
5	<input type="checkbox"/> 多表视图	single_grid	biz_demo_usersview	
6	<input type="checkbox"/> 表格单元格编辑	single_grid	biz_demo_userscell	
7	<input type="checkbox"/> 导入导出	single_grid	biz_demo_import	
8	<input type="checkbox"/> 非自增主键拓展表	single_grid	biz_users_exp	
9	<input type="checkbox"/> 三级菜单	dir	biz_caidan	
10	<input type="checkbox"/> 四级目录	dir	biz_4j	
11	<input type="checkbox"/> 仅显有功能目录	single_grid	biz_4j_test1	
12	<input type="checkbox"/> 我们的爱情	singleGrid	heheh	
13	<input type="checkbox"/> 我们的娃娃	single_grid	22222	
14	<input type="checkbox"/> 23423	single_grid	342	
15				

7.8 Eova 模版

7.9 SingleGrid: 单表

管理中心

评论管理 ×

最大化刷新

快速查询

订单ID:

×

创建时间:

-

快速查询

新增

修改

删除

	ID	订单ID	评分	创建时间	评价
1	1	1	3	2015-09-30 17:49:44	床单没洗干净
2	2	2	3	2015-09-30 17:53:33	太棒了，免费，服务还好！
3	3	4	3	2015-09-30 17:53:50	
4	4	13	2	2015-09-30 17:54:18	还行吧
5	5	11	4	2015-09-30 17:54:28	Nice
6	6	10	3	2015-09-30 17:55:43	还不错，床单洗的很干净

15

第 1 共1页

显示1到6, 共6记录

7.10 MasterSlaveGrid: 主子表

快速查询

酒店名: 商户状态: 创建时间:

快速查询 新增 修改 删除 查看

ID	酒店名	电话	商户状态	积分	创建时间	详细地址
1	松江锦江之星1	(021)37621128	签约商户	12	2015-12-04 17:01:30	上海市松江区西林北路950号
2	松江锦江之星2	(021)37621129	普通商户	0	2015-09-04 17:17:00	上海市松江区西林北路951号
3	松江锦江之星3	(021)37621130	普通商户	0	2015-09-05 17:17:00	上海市松江区西林北路952号
4	10 如家	18824521111	普通商户	0	2015-09-24 00:49:31	上海火车站
5	19 如家酒店	18824521111	普通商户	0		上海某酒店皮草生产有限公司

15 第 1 共1页 显示1到5, 共5记录

酒店床位 酒店存货

新增 修改 删除

床铺尺码	数量
1 90	3
2 120	5
3 150	2

PS:V1.5 开始支持多子，需要手工修改数据库中的 JSON 配置项：eova_menu 表 config 字段

7.11 SingleTree:单树表

快速查询						
名称:		<input type="text"/>	×	编码:	<input type="text"/>	×
快速查询 修改 删除 新增功能 基本功能设置 导出菜单脚本						
	<input type="checkbox"/>	名称	类型	编码	过滤条件	业务拦截器
1	<input type="checkbox"/>	功能演示	dir	biz_demo		
2	<input type="checkbox"/>	单表CRUD	single_grid	biz_demo_users	where status = 1	
3	<input type="checkbox"/>	富文本编辑	single_grid	biz_demo_tool		
4	<input type="checkbox"/>	下拉和查找	single_grid	biz_demo_usersitem		
5	<input type="checkbox"/>	多表视图	single_grid	biz_demo_usersview		
6	<input type="checkbox"/>	表格单元格编辑	single_grid	biz_demo_userscell		
7	<input type="checkbox"/>	导入导出	single_grid	biz_demo_import		
8	<input type="checkbox"/>	非自增主键拓展表	single_grid	biz_users_exp		
9	<input type="checkbox"/>	三级菜单	dir	biz_caidan		
10	<input type="checkbox"/>	四级目录	dir	biz_4j		
11	<input type="checkbox"/>	仅显有功能目录	single_grid	biz_4j_test1		
12	<input type="checkbox"/>	我们的爱情	singleGrid	heheh		
13	<input type="checkbox"/>	我们的娃娃	single_grid	22222		
14	<input type="checkbox"/>	23423	single_grid	342		
15	<input type="checkbox"/>	评论管理	single_grid	biz_comment		
16	<input type="checkbox"/>	产品管理	single_grid	biz_product		
17	<input type="checkbox"/>	酒店管理	master_slave_grid	biz_hotelzz		
18	<input type="checkbox"/>	区域管理	single_tree	biz_demo_area		

PS:V1.6 提供完整版

7.12 TreeToGrid:树关联表

快速查询

功能名称: UI路径: BS路径:

功能演示
产品管理
酒店管理
评论管理

快速查询 新增 修改 删除 查看 导入 导出按钮脚本

	ID	菜单编码	ICON	功能名称	序号	分组号	UI路径	BS路径
1	1071	产品管理		产品发布	7	0	/product/btn/release.html	/product/release

15 第 1 共 1 页 显示 1 到 1, 共 1 记录

PS:V1.6 提供完整版

8. EOVA DIY-自定义拓展

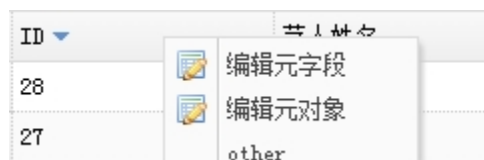
上文介绍了如何快速完成 CRUD，但是在实际的开发过程中会发现很少有纯单表的 CRUD 功能，在实际需求实现的过程中可能会要求实现除了 CRUD 之外的功能，例如：需要导入导出、需要更新状态、需要关联修改其它表...

另外有的表只能新增不能删除，只能修改不能新增...

这些特征我们可以抽象为一件事：添加(减少)一个功能按钮或点击按钮执行一段业务。

特征被抽象之后，那么实现就简单了,下面介绍 Eova 的一些解决方案！

8.1 快速编辑元对象



用超级管理员角色登录，在任意 Grid 表头右键可以快速编辑元对象和元字段数据

修改元对象

ID:

9

名称

玩家信息

视图

数据表

users

主键

id

数据源

默认

单选/多选

☒

行内编辑

☐

显示行号

☒

默认排序

id desc

过滤条件

例如: status = 1 (只显示状态为1的数据)

业务拦截器

继承: com.eova.core.meta.MetaObjectIntercept

依赖JS

8.2 快速编辑元字段

	对象编码	排序 ▲	中文名	字段名	字段类型	控件类型	宽度	高度	自增长	允许查询	允许显示	是否禁用	允许新增
1	产品	1	ID	id	数字	自增框	70	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	产品	1	商品图片	img	字符	文件框	130	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	产品	2	产品类型	type	数字	下拉框	70	20	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	产品	3	分类	category	数字	下拉框	70	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	产品	4	材料	stuff	数字	下拉框	70	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	产品	5	尺码	size	数字	下拉框	70	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

顾名思义，配置不同的属性会出现不同的效果，亲自尝试一次即可了解！

PS：下拉框和查找框表达式编写请参考 [EOVA EXP-表达式功能](#)

8.3 DIY AOP:服务端业务扩展

8.3.1 元对象拦截器: MetaObjectIntercept

默认的业务模版只支持简单的 CRUD，导入，导出...

实际业务需求往往有更多要求，例如：级联删除，更新缓存，备份数据，记录日志，数据转换...

这些业务都非常常见，那么可以使用 EOVA 的【元对象拦截器】来扩展实现这些服务端业务。

8.3.1.1 拦截器配置

平台维护>元数据管理>元对象修改

修改对象

ID:	29	名称:	订单管理	视图:	
数据表:	orders	主键:	id	数据源:	默认
是否单选:	<input checked="" type="checkbox"/>	行内编辑:	<input type="checkbox"/>	显示行号:	<input checked="" type="checkbox"/>
默认排序:	例如: id desc (默认按ID倒序)				
过滤条件:	例如: status = 1 (只显示状态为1的数据)				
业务拦截器:	参考: com.eova.core.meta.MetaObjectIntercept				

确定

8.3.1.2 拦截器接口

```

aop
├── AopContext.java
├── MetaObjectIntercept.java
│   └── MetaObjectIntercept
│       ├── addAfter(AopContext) : String
│       ├── addBefore(AopContext) : String
│       ├── addSucceed(AopContext) : String
│       ├── deleteAfter(AopContext) : String
│       ├── deleteBefore(AopContext) : String
│       ├── deleteSucceed(AopContext) : String
│       ├── queryAfter(AopContext) : void
│       ├── queryBefore(AopContext) : void
│       ├── updateAfter(AopContext) : String
│       ├── updateBefore(AopContext) : String
│       └── updateSucceed(AopContext) : String
    
```

元对象业务拦截器：如上图所示，有常见的各种 AOP 方法，需要实现个性化业务重写之

8.3.1.3 拦截器案例

// 自定义业务拦截器 继承 元对象业务拦截器

```
public class NewsIntercept extends MetaObjectIntercept {
```

// 案例 1：重写新增后置方法 获取当前登录用户的企业 ID 放入当前新增的对象中

```
@Override
```

```
public void addBefore(Controller ctrl, Record record) throws Exception {  
    super.addBefore(ctrl, record);  
    User user = (User)ctrl.getSessionAttr(EovaConst.USER);  
    record.set("company_id", user.get("company_id"));  
}
```

// 案例 2：重写新增后置方法 获取当前时间 作为当前新增对象的注册和更新时间

```
public void addBefore(Controller ctrl, Record record) throws Exception {  
    record.set("reg_time", TimestampUtil.getNow());  
    record.set("update_time", TimestampUtil.getNow());  
}
```

更多案例：Eova 的系统功能很多都有自定义业务拦截器来实现一些个性化功能



DemoIntercept.java

8.3.2 单表模版拦截器: SingleIntercept

说明:

目前 Eova 共提供 4 大模版: 每个模版会提供一个业务拦截器用于扩展模版的业务功能。

目前只有单表模版提供了业务拦截器 SingleIntercept

8.3.2.1 拦截器配置

平台维护>菜单管理>修改菜单

ID:	1001	名称	单表CRUD	类型	single_grid
编码	biz_demo_users	图标	icon-grouplink	序号	1
父节点	1000				
过滤条件	where status = 1 or status = 2				
业务拦截器	继承: 模版业务拦截器				
自定义功能					
是否隐藏	<input type="checkbox"/>				

8.3.2.2 拦截器接口

```
template
├── common
├── masterslave
├── single
│   ├── SingleAtom.java
│   ├── SingleController.java
│   └── SingleIntercept.java
│       ├── SingleIntercept
│       │   ├── importAfter(AopContext) : void
│       │   ├── importBefore(AopContext) : void
│       │   └── importSucceed(AopContext) : void
```

主要用于扩展导入业务, 单表模版的导入是默认向表中新增数据, 实际业务可能需要进行数据校验, 数据过滤, 选择性导入, 部分列导入, 或者覆盖导入...

如有类似需求可以通过模版的业务拦截器来实现。

其它业务模版的业务拦截器后续提供, 大家有好的意见可以积极反馈!

8.4 DIY JS:客户端业务扩展

WEB 应用开发过程中有很多功能需要在客户端实现，例如：数据校验，初始默认值，默认查询条件，自动选中，默认禁用，级联操作，动态变更，异步加载...

上文讲到了使用 AOP 扩展服务端业务，那么 EOVA 作为一个 WEB 开发平台，肯定会支持客户端的业务扩展。在这里 EOVA 使用了最简单的一个方案，支持原生 JS 扩展：直接引入一个 JS 文件，将自定义功能代码写在文件中即可。

8.4.1.1 菜单 DIY_JS 配置

修改菜单					
ID:	1000	名称	功能演示	类型	dir
编码	biz_demo	图标	icon-bookopen	序号	1
父节点	3				
过滤条件	例如: status = 1 (只显示状态为1的数据)				
业务拦截器	继承: 模版业务拦截器				
依赖JS	自定义JS路径				
自定义功能					

PS: JS 作用范围: 模版 List 页面

8.4.1.2 对象 DIY_JS 配置

修改对象					
ID:	1015	名称	地区信息	视图	
数据表	area	主键	id	数据源	默认
单选/多选	<input type="checkbox"/>	行内编辑	<input type="checkbox"/>	显示行号	<input checked="" type="checkbox"/>
默认排序	例如: id desc (默认按ID倒序)				
过滤条件	例如: status = 1 (只显示状态为1的数据)				
业务拦截器	继承: com.eova.core.meta.MetaObjectIntercept				
依赖JS	自定义JS文件路径				

PS: JS 作用范围: 模版 List+Form 页面

8.4.1.3 DIY_JS 案例

Eova Demo 中:

DIY 弹窗: /ui/js/diy/test.js

DIY 级联: /ui/js/diy/area.js

8.5 DIY CRUD:可分配功能配置

菜单管理>功能设置



可用功能配置 (反选禁用功能，禁用后该功能将无法被分配给用户角色)

功能组	新增	修改	删除	查看
功能组0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
功能组1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
功能组2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

功能组位置分布：

快速查询

酒店名: × 商户状态: 创建时间: -

快速查询 ☒ 新增 ☒ 修改 ☒ 删除 ☒ 查看 **功能组0**

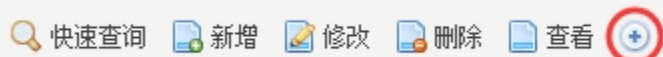
ID	酒店名	电话	商户状态	积分	创建时间	详细地址

确定 取消

典型应用：用户表直接禁止删除和添加数据，由前端系统注册产生数据

此功能控制的是功能是否可被分配给角色，不是控制按钮是否显示

8.6 DIY Button:添加一个新的功能按钮



PS:以超级管理员角色登录才可见快速添加功能入口

快速添加按钮

所属菜单编码	biz_demo_users	
图标	icon-cog	(人靠衣装马靠鞍，选个图标高大上)
名称		(必填)
按钮UI路径	HTML文件路径 例: /order/btn/print.html	(必填)
按钮BS路径	服务端URI 例: /order/print	(必填)
所属按钮组号	0	(默认即可)
分配给角色	超级管理员	(可多选)

确定

取消

PS: BS URI 用于权限控制管理

UI 路径为新增按钮的实现代码(HTML+JS+CSS)

```
<a href="javascript:void(0);" class="easyui-linkbutton" data-options="iconCls:'ext-icon-cog_edit',plain:true"
onclick="eova_importMetaData()">
```

基本功能设置

```
</a>
```

```
<script type="text/javascript">
```

```
var eova_importMetaData = function() {
    var grid = $('#grid');
    var row = grid.datagrid('getSelected');
    if (isSelected(row)){
        loadDialog(grid, '基本功能设置', '/metadata/toMenuFun/' + row.code );
    }
};
```

```
</script>
```

BS 路径为按钮的服务端方法

```
public void menuFun() {
    String menuCode = getPara(0);

    String isAdd = getPara("isAdd");
    String isUpdate = getPara("isUpdate");
    String isDelete = getPara("isDelete");
    ....
}
```



```
// 添加
if (!xx.isEmpty(isAdd)) {
    Eova_Button btn = new Eova_Button();
    btn.set("menuCode", menuCode);
    btn.set("name", SysConst.FUN_ADD);
    btn.set("ui", SysConst.FUN_ADD_UI);
    btn.set("bs", SysConst.FUN_ADD_BS);
    btn.save();
}

.....

renderJson(new Easy());
}
```

PS：因为 数据库的三大范式约定，表在设计的时候都拆分的很细，所以很多功能的业务都要求去查询或者操作关联表。

8.7 DIY View:多表视图

典型应用：多表级联查询合并成一个表格展示数据

1)先在数据库中创建视图

2)[导入元数据]选择元类型=View

3)配置数据案例：

id	object_code	po_code	en	cn	is_auto	data_type	type
2105	v_users_code	player_code	id	ID	1	number	自增框
2106	v_users_code	player_code	status	状态	0	number	文本框
2107	v_users_code	player_code	login_id	登录账户	0	string	文本框
2108	v_users_code	player_code	login_pwd	登录密码	0	string	文本框
2109	v_users_code	player_code	nickname	昵称	0	string	文本域
2110	v_users_code	player_code	reg_time	注册时间	0	time	时间框
2111	v_users_code	player_code	info	备注	0	string	文本域
2112	v_users_code	users_exp_code	exp	经验值	0	number	文本框
2113	v_users_code	users_exp_code	avg	年龄	0	number	文本框
2114	v_users_code	users_exp_code	qq	QQ	0	string	文本框

Eova 中配置的 View 功能直接支持 View 的 CRUD

View 元对象由多个 Table 元对象组成，所以需要先配置好 Table 元对象

po_code 就是配置字段所属的元对象，用于 View 自动 CRUD。

8.8 DIY 完全自定义

直接在菜单管理中新建 DIY 功能，可完全独立开发任意功能，摆脱 Eova 的束缚，完全自由！

9. EOVA EXP-表达式

9.1 元字段表达式

9.1.1 下拉框表达式

select id ID,name 公会 from gms_guild where 1=1;ds=main

语法：select 值 ID,名 CN from 表名 where 条件;ds=数据源

PS:别名必须叫 ID 和 CN，否则无法识别，无条件写 1=1

举例：

select value ID,name CN from `dict` where `class` = 'game' and field = 'type';ds=main

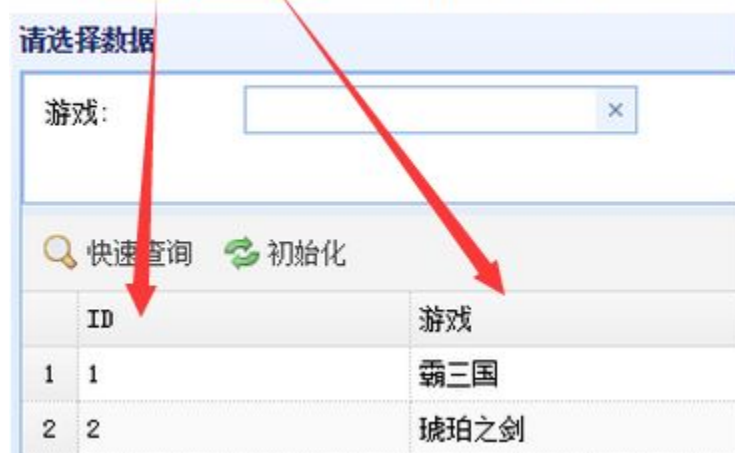


9.1.2 查找框表达式

语法：select 值列 as 显示别名,名称列 as 显示别名 from game;ds=数据源

举例：

select id ID,name 游戏 from game where 1=1;ds=main



	ID	游戏
1	1	霸三国
2	2	琥珀之剑

9.2 过滤条件表达式

```
CREATE TABLE `eova_user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `login_id` varchar(30) NOT NULL COMMENT '帐号',
  `login_pwd` varchar(50) NOT NULL COMMENT '密码',
  `nickname` varchar(255) DEFAULT '' COMMENT '中文名',
  `rid` int(11) DEFAULT '0' COMMENT '角色 ID',
  `hotel_id` int(11) DEFAULT '0' COMMENT '所属酒店',
  PRIMARY KEY (`id`)
```

) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;

用户表中存在用户所属酒店字段，可以用这个字段来控制某些业务的数据权限：酒店员工只能查看其所属酒店的数据

PS：表中的字段可根据业务进行添加，比如常用的：店铺 shop_id；公司 company_id ...

修改元对象			
ID:	28	名称	我的积分记录
数据表	log_score	主键	id
是否单选	<input checked="" type="checkbox"/>	行内编辑	<input type="checkbox"/>
默认排序	0		
过滤条件	hotel_id = \${user.hotel_id}		
业务拦截器	参考: com.eova.core.meta.MetaObjectIntercept		
依赖JS			

如上为酒店积分记录，该酒店管理员只可见其酒店的积分记录。

说明：

目前仅支持当前登录用户对象 语法：**`${user.xxx}`**

还支持 Beetl 逻辑运算：

例：is_del = 0<%(if(user.company_id != 0){%> and company_id =

`${user.company_id}<{%}><%(if(user.shop_id != 0){%> and shop_id = ${user.shop_id}<{%}>`

10. FAQ

Q:Eova 有啥优点呢？

A:省事,通过高度封装业务节省大量代码，甚至不用写代码！

Q:Eova 方便拓展开发吗？

A:Eova 首推的是零代码配置完成功能，当然需求总是奇葩的，所以支持各种灵活的 DIY，比如：CRUD 前后置业务处理(级联变更)，增加新的功能按钮(改变状态按钮),越发丰富的业务模版！


Q:Eova 为何选择 GPL 协议？

A:关于开源，国内很多个人和公司，都是打着开源的幌子，在宣传自己，但是因为天朝的环境，我觉得都能理解。Eova 别的不说，至少可以让大家永久免费使用，也无法强制收费。


如果只是为了学习代码，开源版本已经能学到所有的技能点了，后期的版本主要是修复 BUG，完善功能，做更多的业务模版，而这些对于想学习代码的用户来说没太多的意义，但会方便山寨，方便不劳而获！如果 Eova 真的被大家所认可，后期可能主要会针对企业提供一些技术支持，外包等服务。屌丝何苦为难屌丝，所以请大家放心的用吧。

11. 其它资料


Eova 快速开发视频教程>0 基础操作演示: [点击快速观看](#)


 1.0用Jetty或Tomcat运行Eova By V1.5. gif


 1.1快速创建功能. gif

 1.2快速修改元字段信息. gif


 1.3快速配置自定义按钮. gif

 2.1修改Grid的列名. gif


 2.2设置字段是否显示在Grid中. gif


 2.3设置字段的排列顺序. gif

 2.4设置字段是否允许排序. gif

 2.5设置字段的控件类型. gif

 3.1配置查询条件. gif

 3.2配置Grid单元格编辑. gif

 3.3配置多选下拉框和查找框. gif

12. 待续...

后续会提供更多的干货，项目实战，真实业务需求，高度自定义等...