

目录

目录	1
入门教程	3
1.如何安装Easy Code ?	3
一、在线安装	3
动图教程 :	3
说明文档 :	3
二、手动安装	3
动图教程	3
说明文档 :	3
2.如何添加数据源 ?	4
如何添加数据源 ?	4
动图教程	4
说明文档 :	4
3.如何生成代码 ?	5
一、单表生成	5
动图教程 :	5
说明文档 :	5
二、多表生成	5
动图教程	5
说明文档 :	5
4.怎么去添加类型映射 ?	6
如何添加类型映射 ?	6
动图教程 :	6
说明文档 :	6
实例 :	6
相关链接	6
5.自定义模板	7
自定义模板	7
动图教程 :	7
说明文档 :	7
相关链接 :	7
6.怎么使用全局变量 ?	8
怎么使用全局变量 ?	8
说明文档 :	8
7.怎么将配置导出到本地 ?	9
怎么将配置导出到本地 ?	9
动图教程 :	9
说明文档 :	9
8.多台电脑怎么同步配置 ?	10
多台电脑怎么同步配置 ?	10
动图教程 :	10
说明文档 :	10
9.重置默认设置	11
重置默认设置	11
说明文档 :	11
10.统一配置	12
统一配置	12
说明文档	12
解释 :	12

11.版本控制如果同步表配置？	13
版本控制如果同步表配置？	13
12.常见问题解答	14
常见问题	14
13.常用的原始对象属性	18
原始对象属性	18
表属性	18
列属性	18
无get方法属性：	18
进阶教程	19
1.如何使用debug	19
如何使用debug？	19
动图教程：	19
说明文档：	19
2.如何在表中添加附加列	20
一、自定义附加列	20
动图教程：	20
说明文档：	20
3.如何在列上添加附加属性	21
一、自定义附加属性	21
动图教程：	21
说明文档：	21
4.如何手动修改Easy Code配置	22
如何手动修改Easy Code配置	22

入门教程

1.如何安装Easy Code ？

一、在线安装

动图教程：



说明文档：

- 1. 安装完毕后重启生效。
- 2. 版本更新时不会覆盖现有模板。

二、手动安装

动图教程



说明文档：

- 1. 安装完毕后重启生效。
- 2. 版本更新时不会覆盖现有模板。
- 3. 不要解压zip包，安装时直接选zip包即可。

2.如何添加数据源？

如何添加数据源？

动图教程



说明文档：

- 1. 首次添加数据源需要先下载驱动。
- 2. 支持哪些数据源取决于IDEA，与EasyCode无关。

3.如何生成代码？

一、单表生成

动图教程：



说明文档：

- 1. 每张表生成一次代码后，配置信息将会被保存，方便下次生成。
- 2. 禁止提示表示所有对话框不再弹出，并自动确认。
- 3. 多module项目，注意需要先选择module。
- 4. 统一配置（鸡肋）后面会介绍。

二、多表生成

动图教程



说明文档：

- 1. 可以通过按住ctrl键去选中多张表，或者通过shift方式。
- 2. 默认使用最先被选中的表的配置信息。

4.怎么去添加类型映射？

如何添加类型映射？

动图教程：



说明文档：

1. 默认只有一部分类型映射，其他的需要用户自行添加。

2. 类型映射统一使用正则表达式实现。

3. 不需要使用Java编码时的\\转义符，直接使用\即可。

实例：

表达式	可匹配项	可匹配项	可匹配项	可匹配项	不可匹配项目
int\\(\\d+\\)	int(1)	int(2)	int(10)	int(99999)	int
int\\(\\d+\\)?	int	int(1)	int(10)	int(99999)	
decimal\\(\\d+,\\d+\\)	decimal(1,1)	decimal(10,10)	decimal(100,100)	decimal(10,1)	decimal(1)
decimal\\(\\d+(,\\d+)?\\)	decimal(1,1)	decimal(10)	decimal(100,100)	decimal(10)	decimal
tinyint\\(1\\)	tinyint(1)				
int\\([1-8]\\)	int(1)	int(2)	int(3)	int(8)	int(9)
`int\\(9	\\d{2,}\\)`	int(9)	int(10)	int(123)	int(999)

相关链接

[Java正则表达式](#)

5.自定义模板

自定义模板

动图教程：



说明文档：

1. 编写模板时下面配有说明文档，注意查看。
2. 编写时可以实时调式。
3. 不推荐编辑默认模板，可以复制一个分组进行编辑。（点击分组右边的复制图标或新增图片）
4. 添加模板时名称尽量带扩展名，如entity.java 这样可以实现代码语法高亮。

相关链接：

1. [官网英文文档](#)
2. [Velocity模板引擎语法](#)

6.怎么使用全局变量？

怎么使用全局变量？

说明文档：

1. 全局变量可以在设置中的Other Settings->Easy Code->Global Config中找到。
2. 例如添加一个全局变量名称为demo值为Hello,那么就可以在任意模板中通过\$!demo去使用这个全局变量。当然通常并不会这么使用，而是使用velocity语法
3. 全局变量主要是用来定义宏（velocity宏定义），或者用来编写一大段重复代码。
4. 全局变量的命名很重要，千万不能与模板中的任意变量名冲突。不然会被替换。

简单的来说全局变量就是一个自定义变量，至于怎么玩那就要看个人了。

7.怎么将配置导出到本地？

怎么将配置导出到本地？

动图教程：



说明文档：

1. 该导出方式为IDEA自带方式导出，将会导出EasyCode的所有配置。
2. 可以通过File菜单下的Import Settings选项导入配置。（如果只导出了EasyCode的配置，则只覆盖EasyCode的所有配置。如果存在其他配置，则其他配置
3. 通常用于保存长期不变的配置到本地文件，防止丢失。

8.多台电脑怎么同步配置？

多台电脑怎么同步配置？

动图教程：



说明文档：

1. 可以通过Easy Code自带的导入导出功能实现模板同步。（导出后将token发给需要同步的电脑导入即可）。
2. 导出可以选择需要导出的分组（这就是与IDEA自带导出不同的地方）。
3. 导入也可以选择是否需要覆盖分组。
4. 注意：token是有有效期的。

9.重置默认设置

重置默认设置

说明文档：

1. 重置默认设置可以在设置中的Other Settings->Easy Code中找到。（右上角位置）
2. 重置默认设置只会重置Easy Code自带的分组内容，不会重置用户新增的分组。（这就是建议不要使用默认分组的原因）
3. 作用：Easy Code版本更新时会对默认分组的信息进更改，但是为了保证不覆盖用户的配置，所以更新版本后需要用户手动重置默认设置才能看到最新的模板

10.统一配置

统一配置

说明文档

1. 统一配置只有在多表生成时才有效。

状态	使用当前配置	使用自身配置	保存当前配置
不勾选统一配置	未配置的表	已配置的表	未配置的表
勾选统一配置	未配置的表+已配置的表	无	未配置的表

解释：

1. 已配置的表：生成过代码的表
2. 未配置的表：从未生成过代码的表

11.版本控制如果同步表配置？

版本控制如果同步表配置？

大部分合作开发环境都会使用Git、SVN等版本控制工具，那么在这种情况下如何同步表的配置信息呢？只需要将.idea/EasyCodeConfig文件夹排除忽略添加至

注意：这里所说的配置并不是模板配置，而是针对每张表的配置，后面会做介绍。

12. 常见问题解答

常见问题

问：怎么去掉每张表中的前缀？

答：参考全局变量中的init变量，代码如下。

```
##去掉表的t_前缀
#if($tableInfo.obj.name.startsWith("t_"))
    $!tableInfo.setName($tool.getClassName($tableInfo.obj.name.substring(2)))
#end
```

问：我的实体类有共用基类，怎么排除一些基类里面已经存在的字段呢？

答：参考全局变量中的init变量，代码如下。

```
##去掉表的t_前缀
##实现列进行排除
#set($temp = $tool.newHashSet("testCreateTime", "otherColumn"))
#foreach($item in $temp)
    #set($newList = $tool.newArrayList())
    #foreach($column in $tableInfo.fullColumn)
        #if($column.name!=$item)
            ##带有返回值的方法调用时使用$tool.call来消除返回值
            $tool.call($newList.add($column))
        #end
    #end
#end
##重新保存
$tableInfo.setFullColumn($newList)
#end

##部分列被排除时，可能importList有的类型已经不需要了，下面将对importList进行重写。

##对importList进行篡改
#set($temp = $tool.newHashSet())
#foreach($column in $tableInfo.fullColumn)
    #if(!$column.type.startsWith("java.lang."))
        ##带有返回值的方法调用时使用$tool.call来消除返回值
        $tool.call($temp.add($column.type))
    #end
#end
##覆盖
#set($importList = $temp)
```

问：怎么获取数据库原始类型，类型长度，默认值，主键，外键等信息呢？

答：参考debug章节。

问：执行方法时有返回值，怎么解决？

答：请使用`$tool.call`来消除返回值。

13.常用的原始对象属性

原始对象属性

这里只展示部分常用属性，更多属性可自行通过\$tool.debug调试查看。

表属性

调用方式如下：

```
//统一使用$!tableInfo.obj打头
//获取表名
$!tableInfo.obj.name
// 获取整个数据库脚本
$!tableInfo.obj.dasParent.text
```

属性名	说明	实例
name	表名	table_name
text	当前表的数据库脚本	
dasParent.name	schema名（数据库名）	db_name
dasParent.text	整个schema的数据库脚本	
documentation	当前表结构的html表格	
system	是否为系统自带的表	false

列属性

调用方式如下：

```
//统一使用$!columnInfo.obj打头
//获取列名
$!columnInfo.obj.name
// 获取数据库类型长度
$!columnInfo.obj.dataType.length
```

属性名	说明	实例
name	列名	table_name
default	列默认值	
notNull	是否为非空列	false
position	列在当前表的位置，从1开始	
dataType	数据库类型	int(11)
dataType.length	数据库类型长度	11

无get方法属性：

获取数据库类型名（不带长度，如int）：

```
$!tool.getField($tableInfo.fullColumn.get(0).obj.dataType, "typeName")
```

进阶教程

1. 如何使用debug

如何使用debug？

动图教程：



说明文档：

1. 通过tool.debug方法可以调试任意对象。
2. 通过调试可以查看到对象的所有属性和所有public方法。
3. 通过tool.getField方法可以获取到对象任意属性，包括private属性。
4. 作用：不清楚内部结构的对象都可以通过debug功能来查询。

2.如何在表中添加附加列

一、自定义附加列

动图教程：



说明文档：

- 1. 自定义附加列与数据库的列无关
- 2. 自定义附加列不存在obj对象(原始对象)
- 3. columnInfo.custom 标记是否为自定义附加列，例如生成XML时不需要附加列，可以通过该属性进行排除。
- 4. 注意：由于JSwing中的table编辑框必须失去焦点才能正常保存值，所以编辑完毕后需要将光标移动到别的位置上。然后再点击确认才能正常保存。
- 5. 作用：通常实体类会有一些数据库里面没有的属性，如果手动添加这些属性每次重新生成代码时这些属性就会被消除掉。又需要重新手动去添加，自定义附加列

3.如何在列上添加附加属性

一、自定义附加属性

动图教程：

说明文档：

1. 自定义附加属性支持3中类型：复选框、文本框、下拉选择框

2. 注意：由于JSwing中的table编辑框必须失去焦点才能正常保存值，所以编辑完后需要将光标移动到别的位置上。然后再点击确认才能正常保存。

3. 作用：

例一：如用于生成DTO代码，有的属性不需要生成，通过自定义属性标记一些。然后在模板里面写一些排除代码即可完成。

例二：如果用于生成HTML代码，有的表单只需要新增属性（标记一下），有的表单只需要更新属性（标记一下），模板里面统一处理这些标记。即可到达想要

类型	selectValue	type
复选框	不支持	BOOLEAN
文本框	不支持	TEXT
下拉选择框	英文逗号分割可选项	SELECT

4.如何手动修改Easy Code配置

如何手动修改Easy Code配置

1. 每张生成过代码的表都会在.idea/EasyCodeConfig文件夹下有一个对应的配置文件。
2. 每个配置文件里面都配置着表的各个属性，可以随意删除（删除后重新生成代码会重新出现，但是配置会丢失）。
3. 配置为null的属性会动态的从数据库中抓取。
4. 相信你都看到这里了，这个配置文件还是可以一目了然的。毕竟Easy Code的可扩展性太强了，使用骚操作难免会出问题。遇到问题时可以看看能不能手动改一改这里来解决。

结束语：工具领进门，修行靠个人。

