ModelModel

【】category、model、model\_field关系

创建category时，如果栏目类型为模型栏目或表单栏目，会让选择对应的内容模型和表单模型（关联了有什么用？），例如，新闻栏目对应的内容模型为文章，其下面的国内、国际、军事、娱乐的内容模型也为文章，但是，可以为表单模型，也可以为其他内容模型，如图片等。届时渲染时，会根据栏目设置的首页模板、列表页模板（和内容模型模板关系？？）

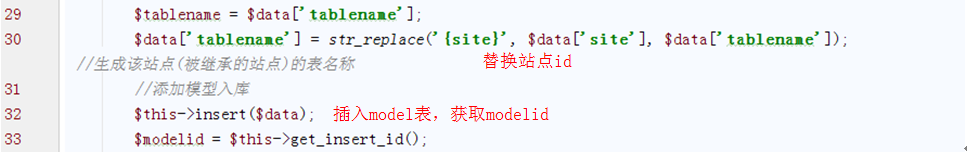
创建1个模型，同时创建1个数据表，表本身的公有信息保存到model表中，表特有的字段信息保存到model\_field表中

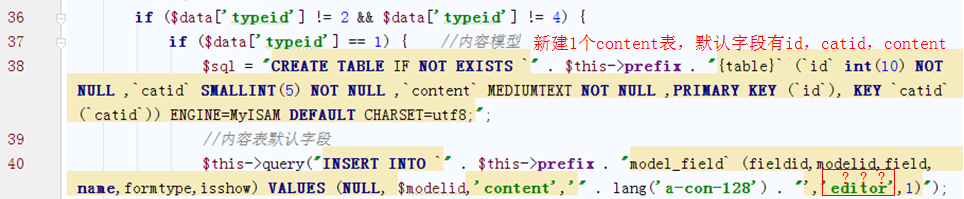
set

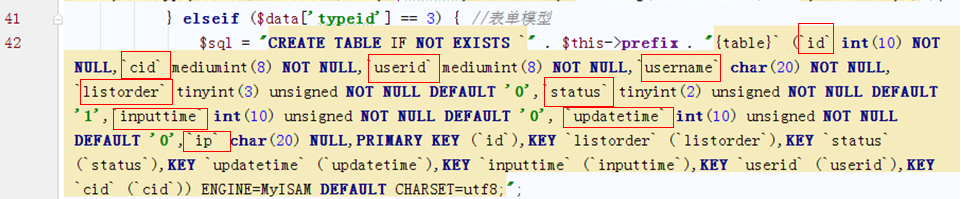
（1）如果有入参modelid，为更新model表信息



（2）新增，包括建模型对应的表，如果是内容模型，还要在model\_field表中插入表字段信息（字段类型为editor是什么意思？）

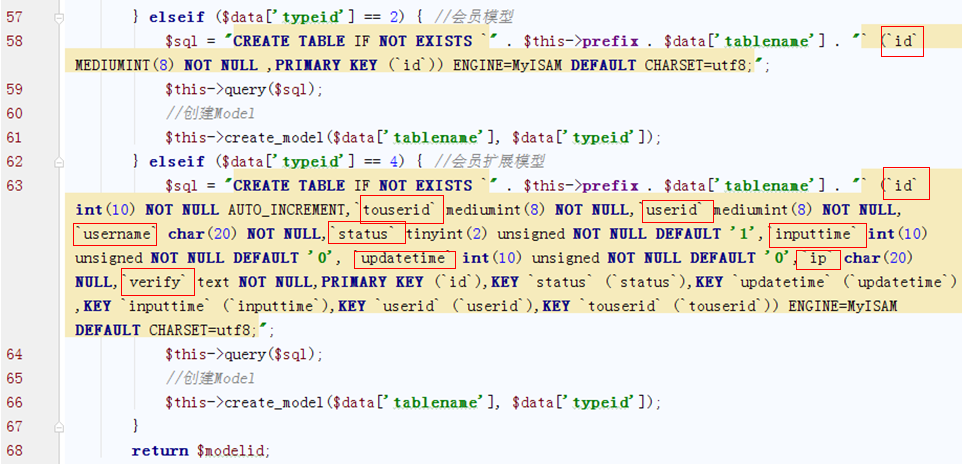








【注意】52行对{site}做了替换，而44行的query其实也做了替换的，在30行，本处的意思是，如果一个站点继承了这次添加模型所在的站点，那么，那边也应该执行新建表操作，另外，因为sql语句有ifnotexist，所以不会重复和出错



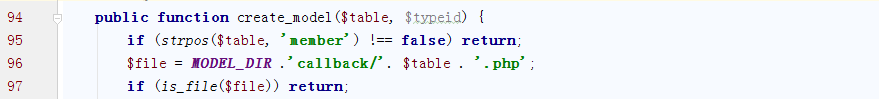
创建表等

del

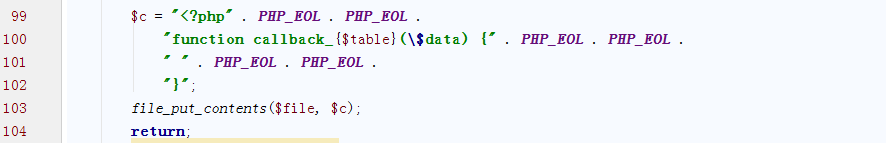
既然模型删了，那么与该模型对应的栏目记录也都应该删除



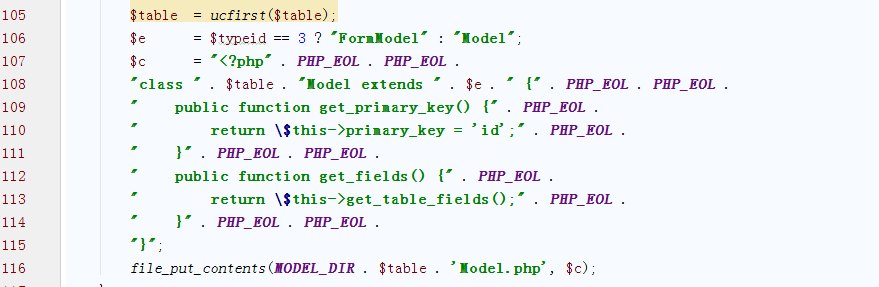
create\_model



检查是否为member，回调函数



写入回调文件内容

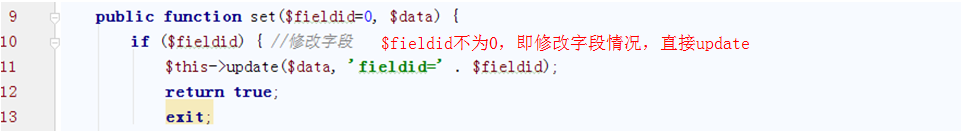


貌似不执行了，创建新的表模型

Model\_fieldModel

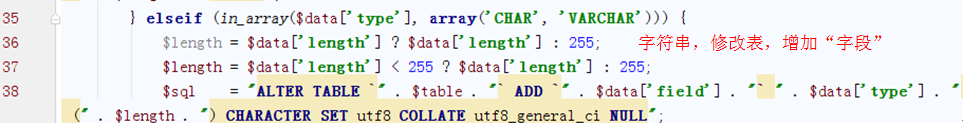
set

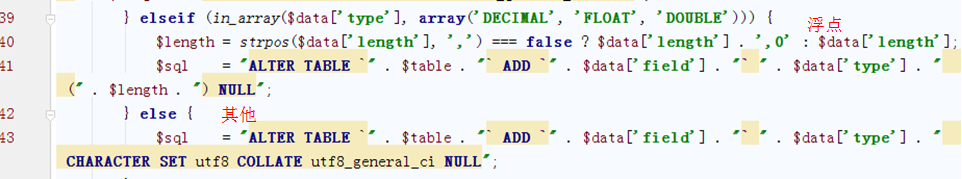
一次只能添加1个字段

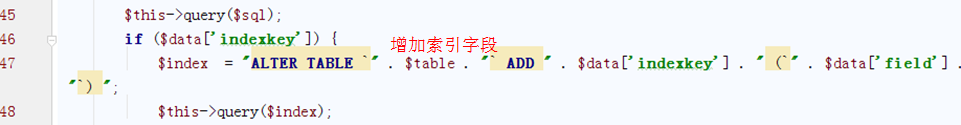


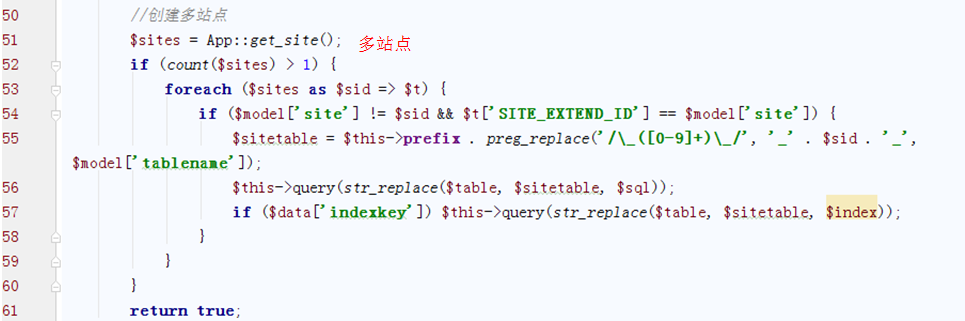










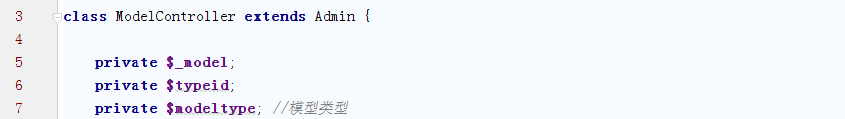


del

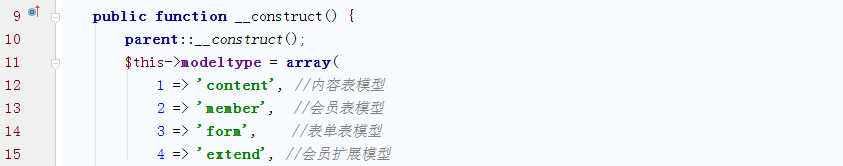


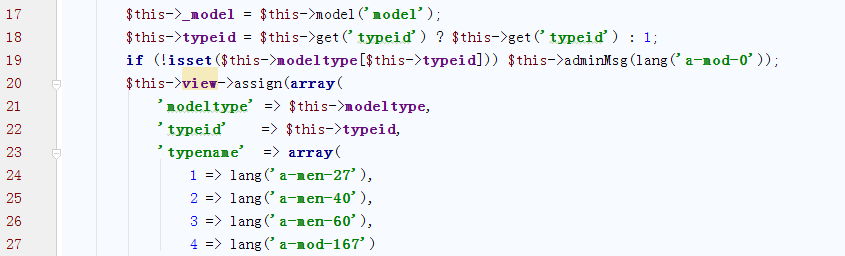


ModelController



\_\_construct





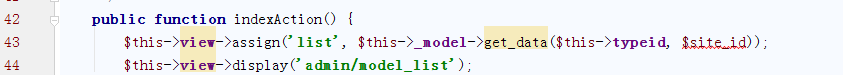
默认模型类型为1即内容模型

get\_model\_site\_id



有继承站点则为继承站点id，否则为自己的id

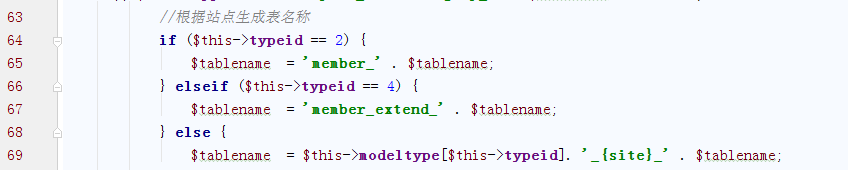
indexAction



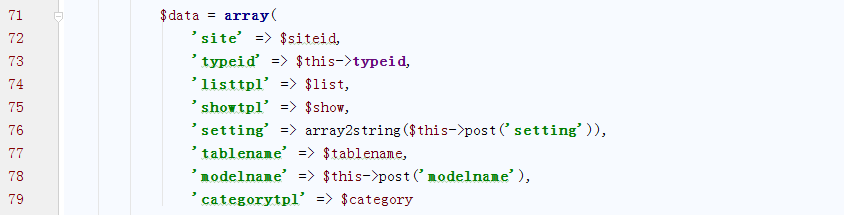
addAction

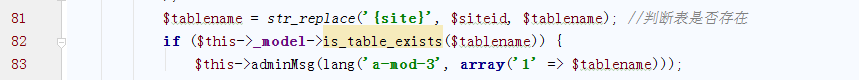


获取设置的表名（表名应该以数字结尾），列表模板、内容模板、栏目模板（如果为表单或会员扩展模型，默认为post\_模板，否则为category\_模板）



生成表名，包含模型类型、站点占位符（用于替换），其中会员和会员扩展模型没有站点名称





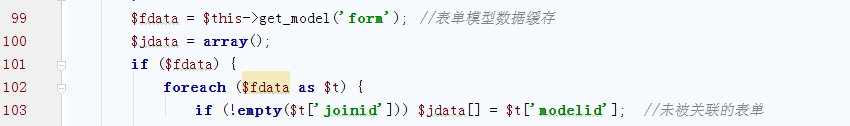
替换表名，检查表是否已存在



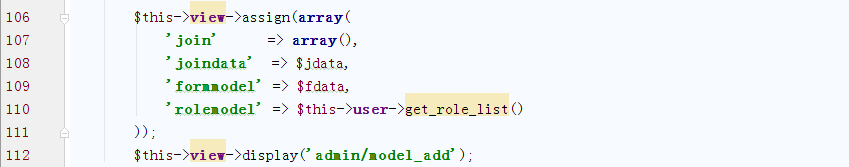
生成表

如果为内容模型，检查是否有入参关联模型，如果有，**更新**对应的关联模型（将model表中id=被关联的$join的那个字段的joinid设为新添加的这个model的id）

【不明白为什么setting输入只有2个值，最后显示完了就有所有的值了】



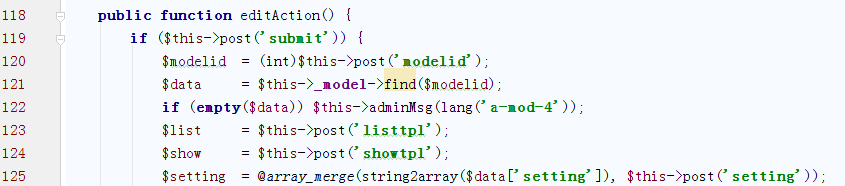
获取表单模型缓存，获取已关联的表单模型id用于显示



【总结】新增模型，就是（1）将填写的字段新增到model表中；（2）创建对应的表；（3）如果有站点继承本站点，创建对应站点的表；（4）如果新建的模型为内容模型，且关联了其他模型，设置对方模型的joinid为新建模型的modelid（显然，被关联的表必须还是表单模型，要不然根本就不会显示）

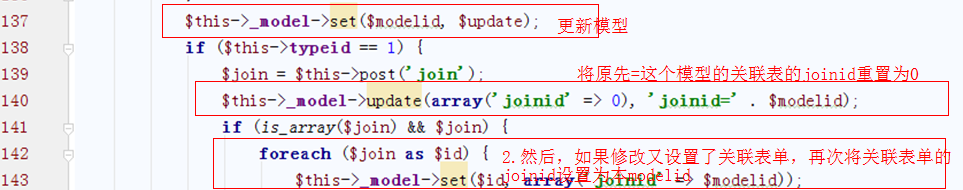
editAction

基本与addAction类似，主要在处理joinid有所不同



略



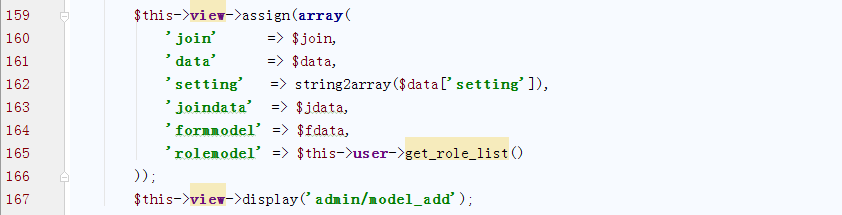


更新数据库

如果为内容模型，获取关联表单，先将joinid=$modelid的记录的joinid全置为0，然后在将获取的关联的表单id对应的记录的joinid全部置为$modelid



（以上是修改post情况）

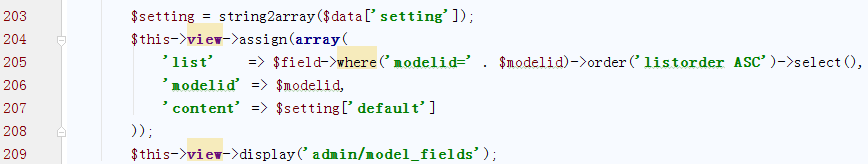


delAction

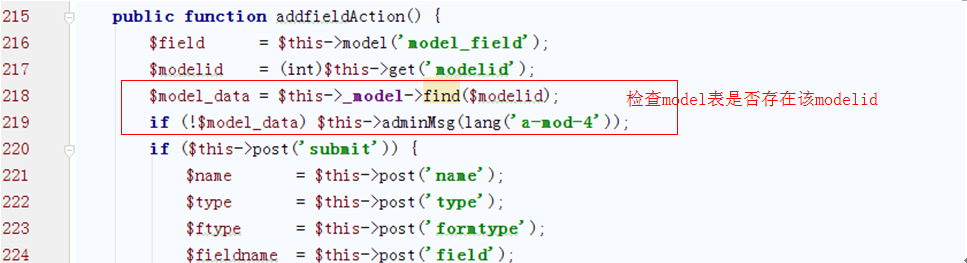


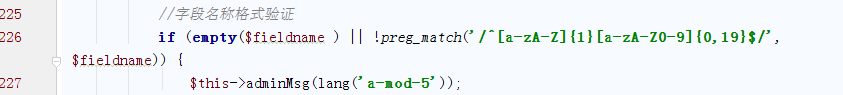
fieldsAction

对表的字段进行排序，略



addfieldAction





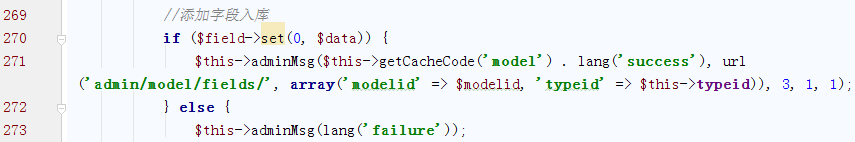
检查字段名







设置数据（均为model\_field表中的字段）



注意：set中也同时修改了附表中的字段

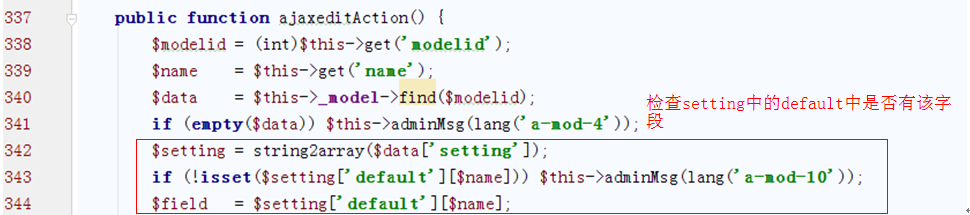


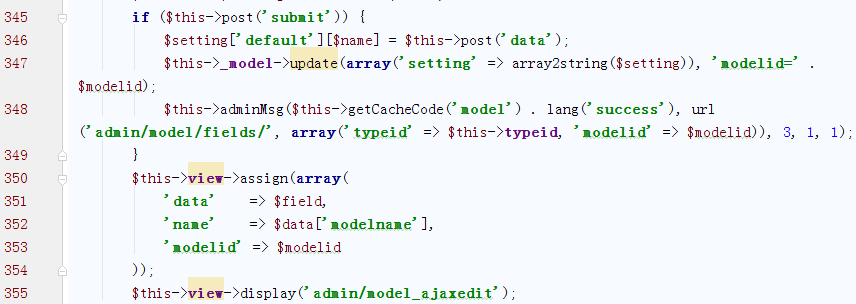
ajaxeditAction





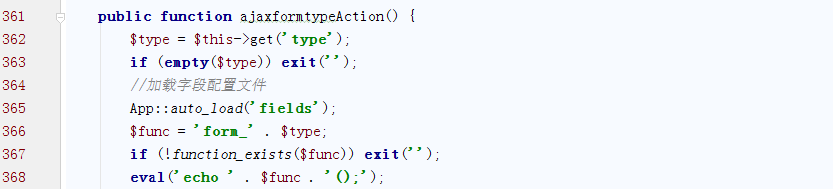
只能修改字段别名和是否显示。



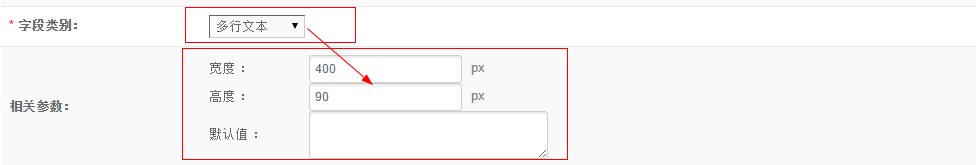


如果为提交，重新设置配置

ajaxformtypeAction



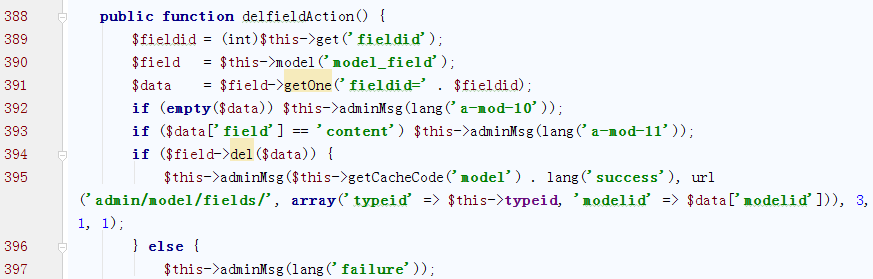
根据字段类型的选择，获取对应的表单类型返回



disableAction

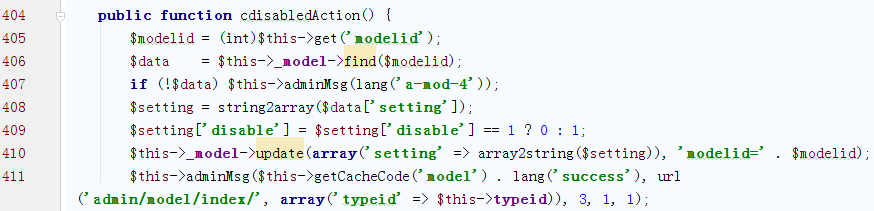


delfieldAction



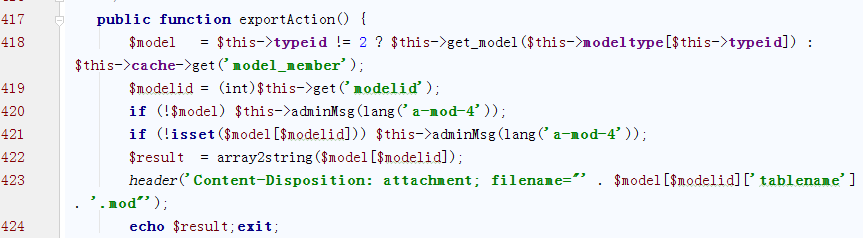
如果字段名为content，返回“字段不存在”

cdisabledAction



禁用/启用模型

exportAction



导出模型，如果为会员模型，直接获取缓存

importAction



检查表名、是否已存在



获取导入文件内容，转为数组



检查模型类型

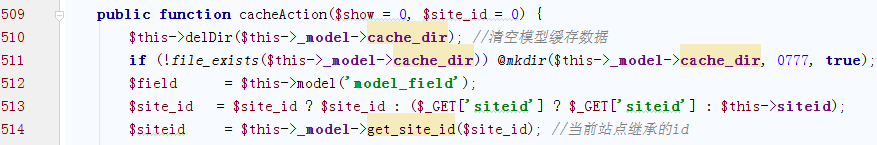


插入数据



更新数据

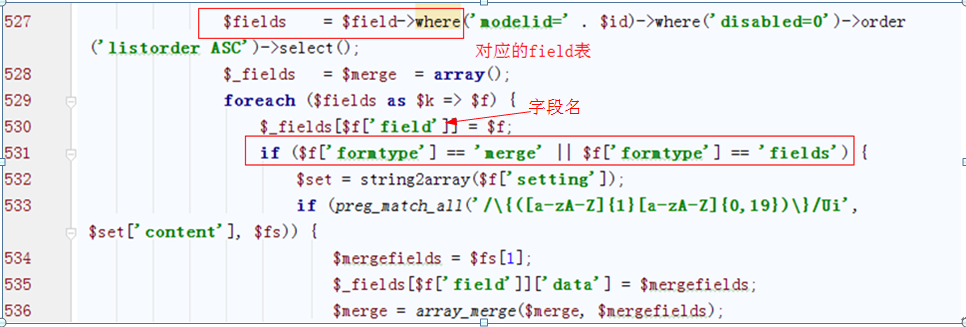
cacheAction



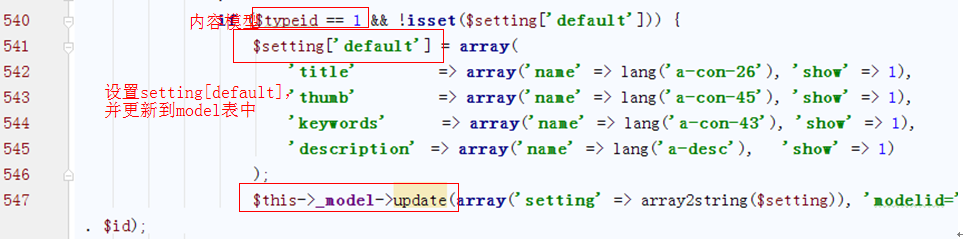
删除模型缓存，创建缓存文件夹，获取模型字段模型

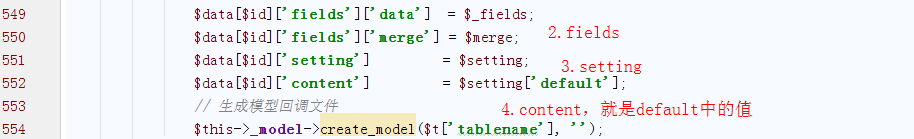


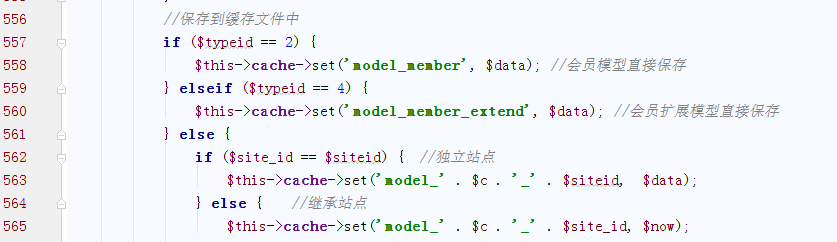
遍历模型类别，获取对应类别的模型数据，遍历，（1）首先获取设置，如果为disabled，continue；如果为继承站点，重新设置站点id和表名，整个缓存数组的id为model类型



（2）设置fields，获取model\_field表中数据，遍历，如果表单类型为merge或fields，获取设置中的content，匹配字母和数字子组，设为[field][data]的值

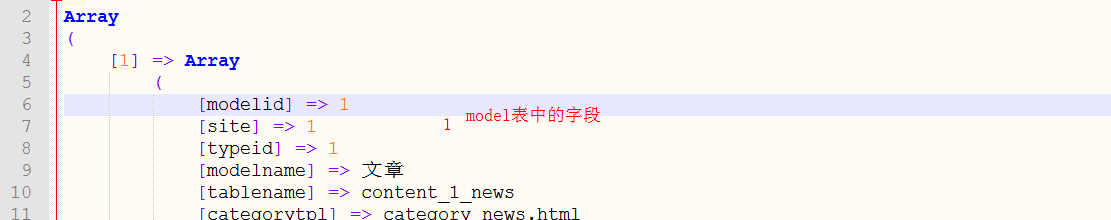


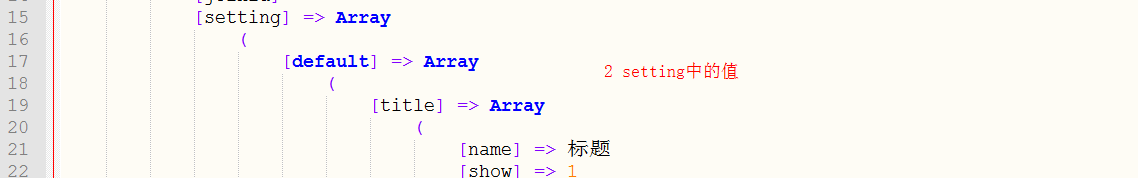


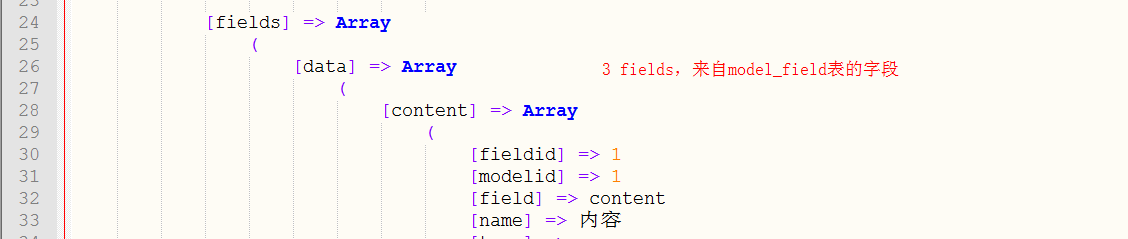


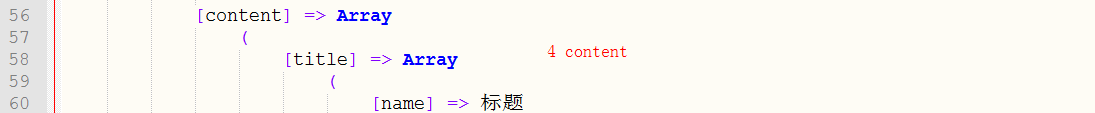
会员模型和会员扩展模型直接更新缓存，其他，加上站点id



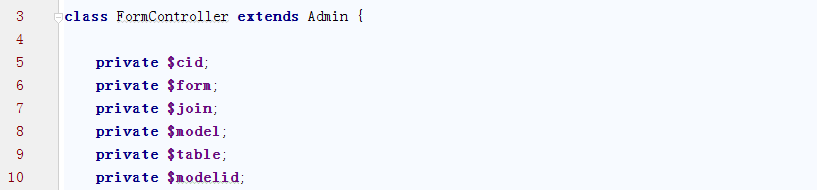




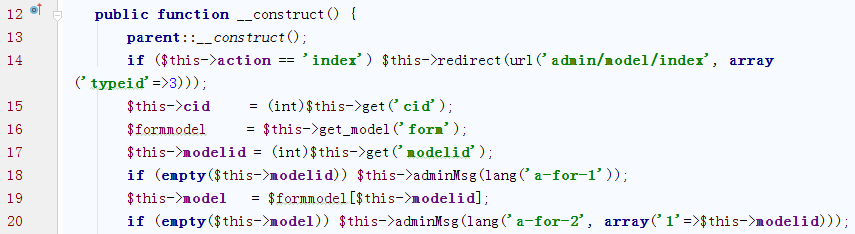




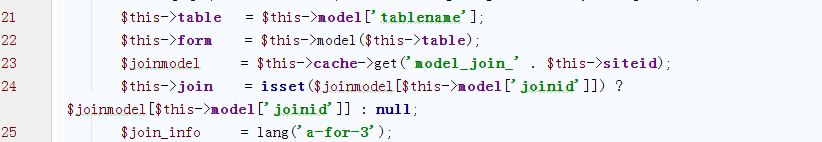
FormController



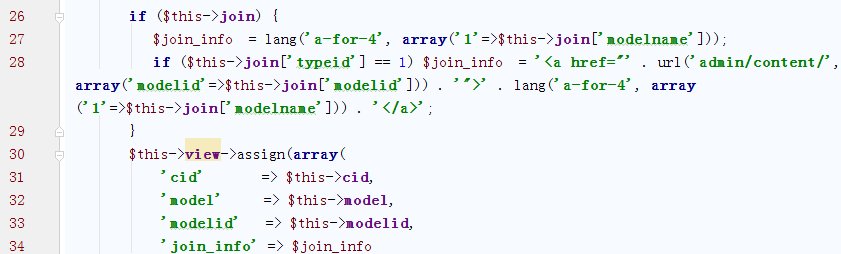
\_\_construct



如果a为index，跳转到model控制器（展示）



获取表名，对应的表model，关联model



如果有关联，如果关联的是内容模型，设置a标签

listAction

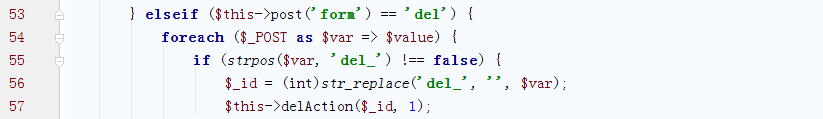


【获取关联文章】

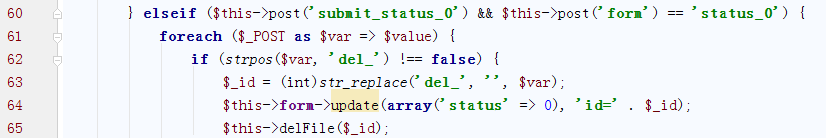


如果为搜索，获取关键字等

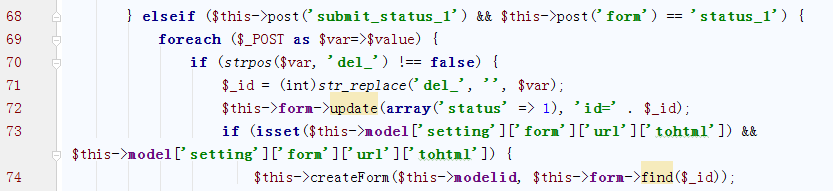
如果为排序，更新排序



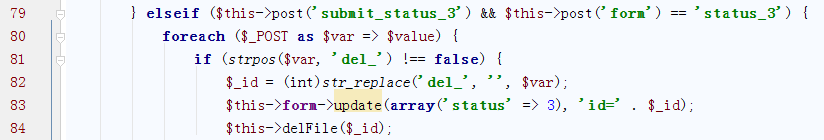
如果为删除



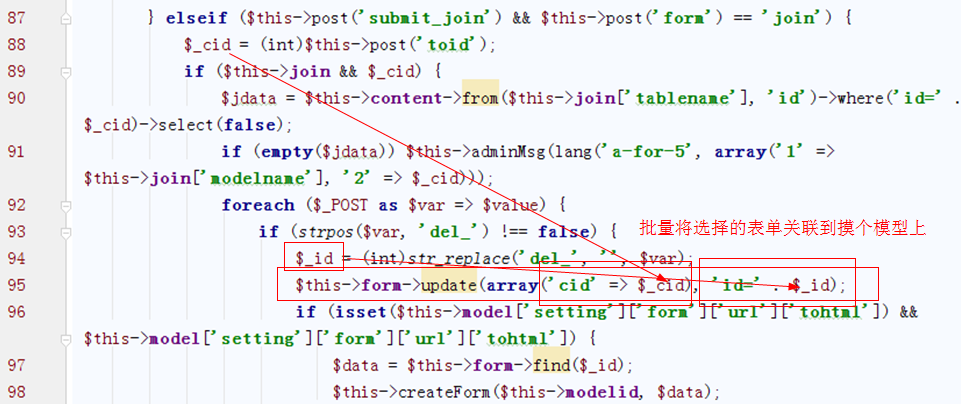
标记为未审



标记为审核



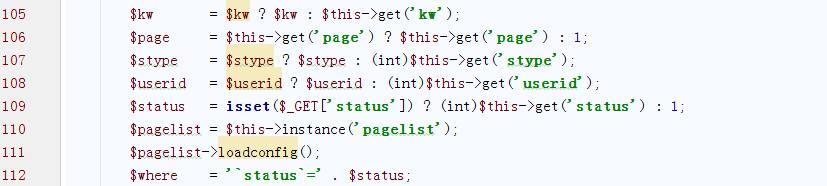
放入回收站



批量关联，批量关联应该是关联到对应的表模型。90行跟content表没关系，就是从join[tablename]中查找对应的关联模型。先更新form表的cid。

如果model模型的设置有生成url的html，生成表单html

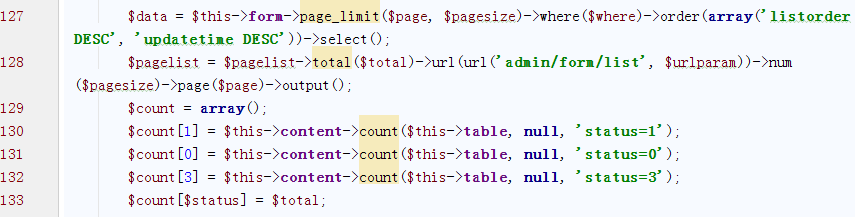




如果为搜索，获取搜索选项



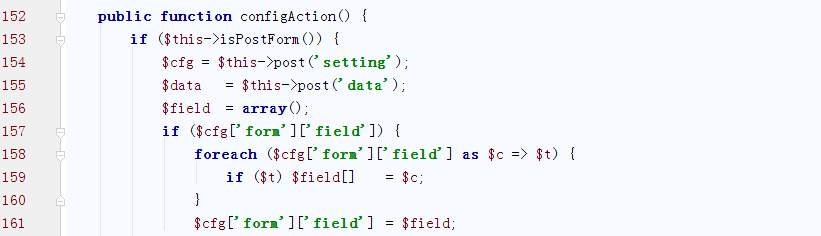
组合sql语句，查询内容表



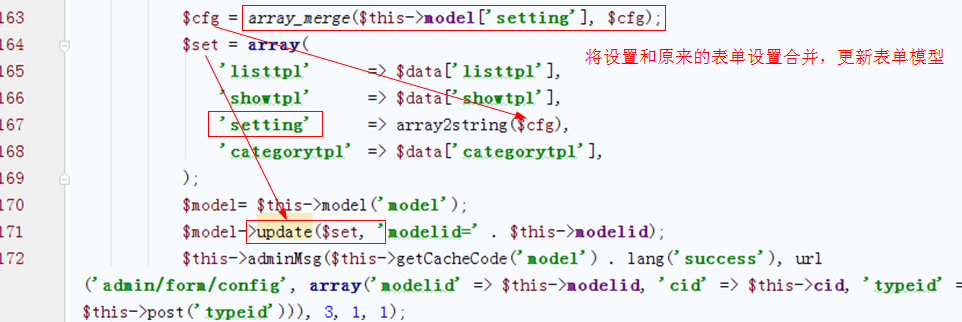
查询form模型，

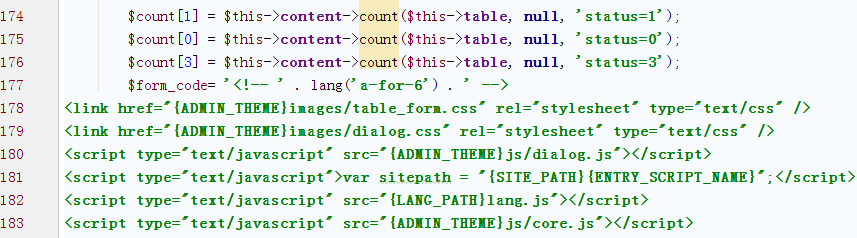


configAction



获取表单配置字段







显示list工具

addAction（发布文档）

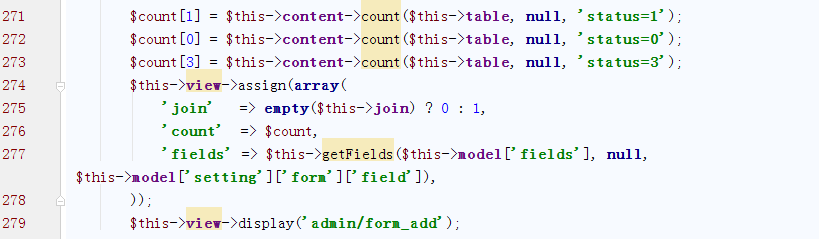


检查权限，获取数据，必须要关联才行

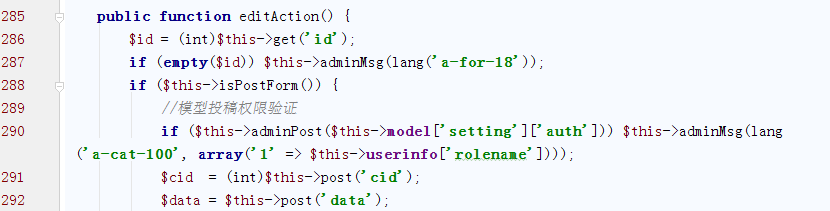
如果关联，获取关联表及记录（由此可见cid不是栏目，而是关联表记录的id）



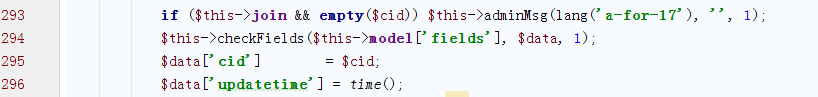
检查字段，设置属性，插入，生成静态文件，否则报错



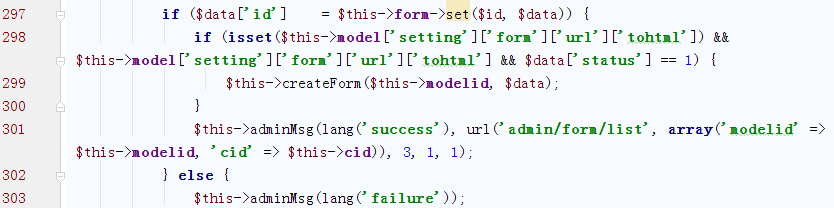
editAction



获取数据



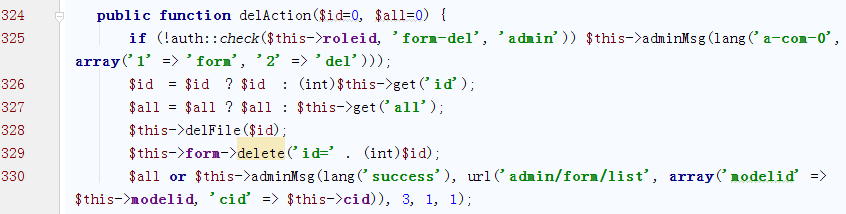
关联检查和检查字段



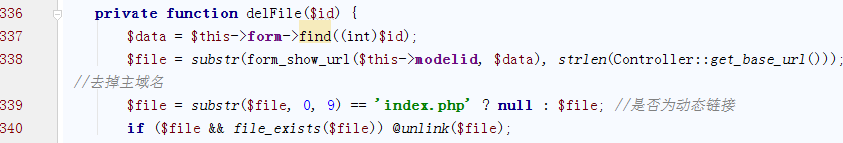
更新表，检查是否需要生成静态文件设置



delAction



delFile



特别说明

**1.表单关联功能**

试将表单模型的joinid设置为对应的被关联的内容模型的modelid，而不是相反。而对应的内容模型的joinid居然为null