**技术文档**

[数据库](#数据库)

[类](#类)

窗体

其他

1. 数据库：

思想：

将所有要存的数据当成一个节点，节点的属性有**编号，类型，内容**。**编号与节点是一一对应关系**，所有节点编号不同，知道编号即可获取一个节点。**类型用来标识这个节点被使用时的情境**。**内容就是存入这个节点点的数据**。例如一个用户的用户名为123，则存入表中建立节点时由数据库分配一个编号，并标记节点的类型为“用户名”，节点数据为“123”。

两个有关联的节点可以连线，表示它们有关联，线的属性有**出发点，指向点，类型**，类型大多时候为无类型。

表：

style表中，id表示类型的数字形式，styleName表示类型的英文名，tableName表示这个类型的数据要存在哪种表中。（显然文章是string，性别是bool，不同类型不分开存会造成空间浪费）

data表中，id列即节点编号，style列即节点类型，根据style找到存储表，tableId表示数据在存储表中第几行。

lines表中，firstId表示出发点，secondId表示指向点，style表示连线类型。

1. 类：
2. essential：

简介：

封装了所有基础操作和节点类型和session cookies的名字（但还是有些repeater绑定时的魔法值没收录）。函数名可以自解释。

2、sqlSever：

简介：

封装了奇怪的建表方式的所有操作。sqlOther和sqlSelect是执行正常sql语句的，对字符串搜索使用了参数化查询。

方法：

NewData表示新建节点，同时会返回数据库分配的编号；deleteData表示按编号删除节点，会同时删除一切连线；deleteLines请与getLines配合使用；带get的均为按条件查询指定内容。其他方法可以自解释。

3、email：网上copy，注释详细。

4、user：

用户的存储方式：

用户名节点，密码节点，邮箱节点，cookies节点。用户名出发向密码、邮箱和cookies各连一条有向边。当用户选择自动登录时cookies节点记录分配给用户的随机字符串。

方法：

send 向指定邮箱发送指定内容，发件人是我的qq邮箱 （emmmmm）；checkLogin 先看session，没有则看cookies，再没有跳转登录界面；foolCheck 判断登录以及url后面的传值，任何页面一个foolCheck就可以防止用户通过地址栏随意跳转页面；getUserLevel 获得主客关系（目前只有好友和陌生人两种关系）。其他方法可自解释。

1. 窗体：