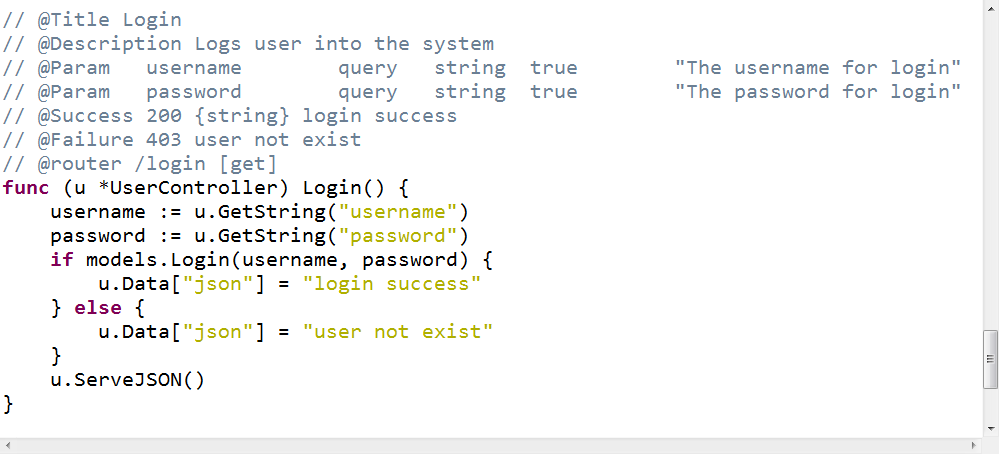
1. Git提交规范
   1. 确保提交的代码可编译运行
   2. 当提交代码时发现存在冲突不能提交时，需要解决冲突，不能直接复制粘贴，覆盖别人的提交
   3. 提交时，需要在message中明确注明这次的修改内容
2. Beego使用规范
   1. 第三方库使用，Import 不要使用相对路径引入包，要使用绝对路径
   2. 一切依赖中间件的参数，都需要写在配置文件，由程序读取配置文件
   3. 路由配置，参照beego中自动生成的api项目，采用注解路由@router自动生成策略，尽量使用明确定义的路由，不要依赖默认的路由配置，https://beego.me/docs/mvc/controller/router.md
   4. 文档API，参照beego中自动生成的api项目，采用自动生成策略，https://beego.me/docs/advantage/docs.md



1. 代码规范
   1. 命名(驼峰)
      1. 参数，变量：小写驼峰
      2. 公开参数：大写驼峰
      3. 参数要尽量有相关性，User user、User u、User u2（User zzz，避免这种写法）
      4. Model：Xxx 、Controller：XxxController
      5. Json字段：小写驼峰
      6. 函数名：动词+名词(驼峰)
   2. Error处理
      1. 有返回error的函数尽量都要处理，不能使用\_忽略error，
      2. 不要直接输出日志，就当解决问题，需要根据函数的定义去实现，函数不能处理error时则返回调用者，由上层解决error，如果函数可以处理error，也需要通过别的方式通知调用者调用有问题，如返回nil，或者返回码等
   3. 禁用panic
   4. 避免长函数，尽量在20行以内
   5. 嵌套层次尽量不要超过3层
   6. 高内聚，低耦合，将一系列功能相似的函数放在一起，在util包中建立YyyUtil工具集，方便查找和复用
   7. 代码中不要出现fmt，使用fmt的地方都改成使用log
   8. 保存前需要将代码格式化
   9. 尽量使用绝对路径，少用相对路径
   10. 复杂代码需要在代码中附上必要的注释
2. 数据库
   1. 表中字段命名采用orm的默认即可
   2. 采用utf8mb4编码
   3. 系统尽量不用使用共享数据库方式交互，尽量采用api方式
   4. 需要注意表索引