

# 作业 1: 寻宝游戏

---

考虑以下游戏场景:

- 每个游戏玩家都有一定数量的金币、宝物。有一个市场供玩家们买卖宝物。玩家可以将宝物放到市场上挂牌, 自己确定价格。其他玩家支付足够的金币, 可购买宝物。
- 宝物分为两类: 一类为工具, 它决定持有玩家的工作能力; 一类为配饰, 它决定持有玩家的运气。
- 每位玩家每天可以通过寻宝获得一件宝物, 宝物的价值由玩家的运气决定。每位玩家每天可以通过劳动赚取金币, 赚得多少由玩家的工作能力决定。(游戏中的一天可以是现实中的1分钟、5分钟、10分钟, 自主设定。)
- 每个宝物都有一个自己的名字(尽量不重复)。每位玩家能够佩戴的宝物是有限的(比如一个玩家只能佩戴一个工具和两个配饰)。多余的宝物被放在存储箱中, 不起作用, 但可以拿到市场出售。
- 在市场上挂牌的宝物必须在存储箱中并仍然在存储箱中, 直到宝物被卖出。挂牌的宝物可以被收回, 并以新的价格重新挂牌。当存储箱装不下时, 运气或工作能力值最低的宝物将被系统自动回收。
- 假设游戏永不停止而玩家的最终目的是获得最好的宝物。

请根据以上场景构建一个假想的 Web 游戏, 可供多人在线上玩耍。界面尽可能简单(简单文字和链接即可, 不需要 style)。后台的数据库使用 MongoDB。对游戏玩家

提供以下几种操作: 寻宝(可以自动每天一次)、赚钱(可以自动每天一次)、佩戴宝物、浏览市场、买宝物、挂牌宝物、收回宝物。

---

提交: 程序 + 文档

要求:

- 文档主要用于解释你的数据库设计, 即需要构建哪些 collection, 每个 collection 的文档结构是什么, 需要构建哪些索引, 应用如何访问数据库(具体的 CRUD 命令)。
  - 为玩家的操作设计 JSON HTTP 协议的接口, 自定义接口格式(request 和 response 的 JSON), 为每个接口编写测试用例和测试代码, 并在文档中包含测试结果。
  - 不限制编程语言及 web 框架。
  - 请将代码和文档(markdown 格式)一起上传至水杉码园。
- 

以下资料可供参考

MongoDB Install (ubuntu): <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/>

MongoDB CRUD: <https://docs.mongodb.com/manual/crud/>

MongoDB Data Model: <https://docs.mongodb.com/manual/core/data-modeling-introduction/>

MongoDB & Language:

Python: <https://www.mongodb.com/languages/python>

Golang: <https://www.mongodb.com/languages/golang>

Others: <https://docs.mongodb.com/drivers/>

Web development Frameworks:

Flask: <https://flask.palletsprojects.com/en/2.0.x/>

FastAPI: <https://fastapi.tiangolo.com>

gin: <https://github.com/gin-gonic/gin#gin-web-framework>

beego: <https://github.com/beego/beego>

API Design: <https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/>

Test API: <https://www.katalon.com/resources-center/blog/api-testing-tips/>