

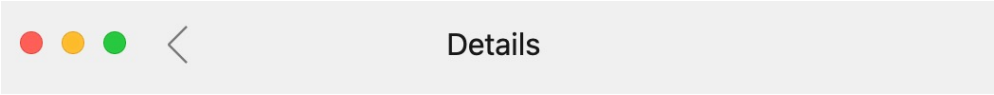
零基础? 从零开始? 速成?

搭建一个提问箱

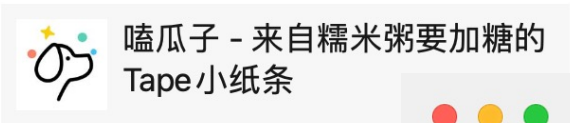
王衍飞, xianfei@bupt.cn

Bilibili: xianfei , Github: xianfei

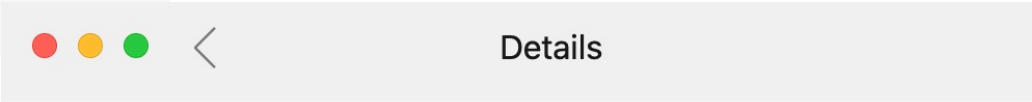
参考的是Tape (Popi)



一位朋友



Yesterday 11:20 PM Tape



令一位朋友

继续蹲一些第一印象🐱



January 14, 2022 4:51 PM Tape



制作提问箱/ 回答问题.....

- 需要注册登录
- 可以创建多个提问箱
- 可以分享某个提问箱的链接
- 可以回答他人提出的问题

进入别人的提 问箱回答问题

- 无需注册登录，匿名提问
- 通过他人分享的提问箱链接进入
- 可以对其进行提问
- 可以获取问题ID用于追踪

关于本分享

- 1. 我也是自己学的 如果有不对的地方请指出
- 2. 致力于用最简单的方式
- 3. 会指出优化方向
- 4. 前提知识：了解一门编程语言（C/Java/JS/...），了解计算机网络基础知识
- 5. 本教程会从一台全新的Windows电脑开始带着大家配环境，同时也会提到如何在macOS进行开发，最后会教大家如何部署到一台Linux云服务器上
- 6. 手把手写代码
- 7. 仅仅分享知识 我也不挣钱 不喜欢可以看看其他UP主的作品

技术栈

- 后端：Node + Express + MongoDB
- 前端：HTML+CSS+JS基本语法 + JQuery示例 + Vue示例
- 需要下载的软件：Node MongoDB VSCode
MongoDB Compass (可选)

需要npm安装的依赖：

```
npm i body-parser cookie-parser express jsonwebtoken mongoose --save
```

将会学到什么

- 像全栈工程师一样自己做一整个系统 (Build-and-fix model)
- 使用HTML CSS 编写一个页面 (CSS选择器语法)
- 使用JavaScript编写逻辑代码，并简单尝试Vue
- 什么是前后端，什么是GET/POST，服务器与用户如何通信
- 如何使用非关系型（文档型）数据库
- 使用SSH、SCP等经典工具操作Linux服务器

安排

- 本次视频:

介绍+安装环境+功能分析+前端线框图绘制+数据库建模

- 之后的视频:

- 编写登录、注册、创建提问箱部分的前后端+数据库操作代码，分享提问箱，未登录用户提问，回答问题，绑定回答的其他人问题等代码实现，部署到Linux服务器，css美化等

制作提问箱/ 回答问题.....

- 需要注册登录
- 可以创建多个提问箱
- 可以分享某个提问箱的链接
- 可以回答他人提出的问题

进入别人的提 问箱回答问题

- 无需注册登录，匿名提问
- 通过他人分享的提问箱链接进入
- 可以对其进行提问
- 可以获取问题ID用于追踪

线框图-登录注册

登录

用户名：

密码：

登录

注册

注册

用户名：

密码：

重复密码

注册并登录

线框图-提问箱列表、消息列表

创建提问箱

绑定问题

消息列表

退出登录

提问箱标题

已回答：待回答：

提问箱标题

已回答：待回答：

提问被回答

提问内容

回答内容

时间

新提问

提问内容

提问箱标题

时间

线框图-提问箱页面

(未登录用户 / 自己的提问箱)

提问箱标题

创建提问

共2个回答

提问
回答

提问
回答

提问箱标题

分享

修改

删除

1个问题待回答

提问

共2个回答

提问
回答

提问
回答

线框图-提问

提问箱标题

回答：
回答

提问箱标题

等待回答

问题ID：1111111111
复制该ID到App中进行绑定可以获得回答通知

数据库建模

User
username:String
password:String
boxiesIds:[String]
bindingQuesIds:[String]
message:[...]
_id:

Box
userId:String
name:String
answeredQuesIds:[String]
noAnswerQuesIds:[String]
_id:

Question
boxId:String
askUserId:String
question:String
hasAnswer:Boolean
answer:String
_id:

第二次课：

通常，在生产环境中，为保证用户密码不被泄露，密码通常经过不可逆变换后存入数据库，来保证密码安全。常见算法如多次Hash等。在此项目中可引入bcrypt库来解决此问题

```
// generate salt to hash password
const salt = await bcrypt.genSalt(10);
// now we set user password to hashed password
user.password = await bcrypt.hash(user.password, salt);
```

算法	特点	有效破解方式	破解难度	其它
明文保存	实现简单	无需破解	简单	
对称加密	可以解密出明文	获取密钥	中	需要确保密钥不泄露
单向 HASH	不可解密	碰撞、彩虹表	中	
特殊 HASH	不可解密	碰撞、彩虹表	中	需要确保“盐”不泄露
Pbkdf2	不可解密	无	难	需要设定合理的参数

源代码：<https://github.com/xianfei/AskBox>