牵手未来-技术部

Hey，很高兴认识你。
我们曾主导过千万**DAU**量级的社交**APP**；
我们是原探探**CEO**和探探核心技术团队组成的创业团队；
我们对新技术保持开放，希望你是对技术有追求且不拘一格的人 ，具有极客精神且热爱创业的人。

为了更好的了解您的技术能力，特邀请您完成以下测试题。请完成后发送代码仓库链接给**HR**，我们会在*8*小时内给到您反馈。

牵手后端实习生编程题

Thank you for showing your interest in Qianshou and applying for the open Backend Engineer Intern position! As part of our hiring procedure we would like you to do a small coding test.

The task is to implement a basic **REST**ful **HTTP** service in **Go** for a simplified Qianshou backend: Adding users and swiping other people in order to find a match.

You should follow the **API** specification given in this document. We want the data to be stored in a **PostgreSQL** database. All input and output should be in **JSON** format, and the API should return the proper **HTTP status codes**.

When finished with the task, commit your source code to a Github repo with a **README** file, and send your Github repo URL to us.

We will primarily look at these criteria

- That the API works and follows the API specification
- That the code is well structured, follows go conventions and is easy to understand
- That the database is properly designed and used
- That the code handle concurrent requests correctly

Good Luck!
The Qianshou Tech Team


牵手未来-技术部

API specification

Users:

GET /users
List all users

Example:
 	
$curl -XGET "http://localhost:80/users"
 	
[
  {
  "id": "21341231231",
  "name": "Bob",
  "type": "user"
  },
  {
  "id": "31231242322",
  "name": "Samantha",
  "type": "user"
  }
]

POST /users

Create a user

allowed fields:
  name = string

Example:

$curl -XPOST -d '{"name":"Alice"}' "http://localhost:80/users"

{
    "id": "11231244213",
    "name": "Alice",
    "type": "user"
}


牵手未来-技术部

Relationships:

GET /users/:user_id/relationships
List a users all relationships

Example:
 	
$curl -XGET "http://localhost:80/users/11231244213/relationships"
 	
[
  {
  "user_id": "222333444",
  "state": "liked",
  "type": "relationship"
  },
  {
  "user_id": "333222444",
  "state": "matched",
  "type": "relationship"
  },
  {
  "user_id": "444333222",
  "state": "disliked",
  "type": "relationship"
  }
]


牵手未来-技术部

PUT /users/:user_id/relationships/:other_user_id

Create/update relationship state to another user.

allowed fields:
  state = "liked"|"disliked"

If two users have "liked" each other, then the state of the relationship is "matched"

Examples:

```
$curl -XPUT -d '{"state":"liked"}' "http://localhost:80/users/11231244213/relationships/21341231231"
{
   "user_id": "21341231231",
   "state": "liked",
   "type": "relationship"
}

$curl -XPUT -d '{"state":"liked"}' "http://localhost:80/users/21341231231/relationships/11231244213"
{
   "user_id": "11231244213",
   "state": "matched",
   "type": "relationship"
}

$curl -XPUT -d '{"state":"disliked"}' "http://localhost:80/users/21341231231/relationships/11231244213"
{
   "user_id": "11231244213",
   "state": "disliked",
   "type": "relationship"
}
```