



Road to MIXI

Mini SNS 00

Summary: このドキュメントは、*Road to MIXI @ 42 Tokyo*の*Mini SNS 00*モジュール用の課題である。



Contents

I	Introduction	2
II	Instruction	3
III	Prerequisite	4
III.1	Repository Setup	4
III.2	Docker	4
III.3	Data	5
IV	Exercise 00 : get_friend_list	6
V	Exercise 01 : get_friend_of_friend_list	7
VI	Exercise 02 : get_friend_of_friend_list_v2	8
VII	Exercise 03 : get_friend_of_friend_list_paging	9
VIII	Bonus: 新機能の追加	10

Chapter I

Introduction

Road to MIXI Mini SNS 00ではミニマルSNSのバックエンドAPIを実装していただきます。

Chapter II

Instruction

- 課題に関する噂に惑わされないよう気をつけ、信用しないこと。
- この書類は、提出前に変更になる可能性があるため、気をつけること。
- ファイルとディレクトリへの権限があることを、あらかじめ確認すること。
- すべての課題は、提出手順に従い行うこと。提出手順が記載されていない場合、取り組んだ課題をTurn-in Repositoryに提出すること以外は、任意とする。
- 課題の確認と評価は、あなたの周りにいる学生により行われる。
- 課題は、簡単なものから徐々に難しくなるように並べられている。
- 課題で指定されていないものは、どんなファイルもディレクトリ内に置かないこと。
- 助けてくれるのは、Google / 人間 / インターネット / ...と呼ばれているものたちである。
- 出力例には、問題文に明記されていない細部まで表示されている場合があるため、入念に確認すること。
- 各問題は、Go か Rubyを使用すること。使用する言語のバージョンは任意とする。
- 各問題は、機能の実装のみならず、テストもできるだけ作成すること。

Chapter III

Prerequisite

III.1 Repository Setup

こちらのリポジトリからこの課題に取り組むために必要なファイルをダウンロードすること。

<https://github.com/mixigroup/road-to-mixi/tree/master/problem1>

RubyかGo、どちらの言語で課題に取り組むか選択した後は、[docker-compose.yml](#)ファイルのコメントアウトを必要に応じて行うこと。

III.2 Docker

サービスの起動は

```
docker-compose up -d
```

コマンドで実行できる。

III.3 Data

このSNSはユーザー間にフレンド関係だけが設定できるSNSで、ユーザは

- ID
- 名前
- フレンド関係

というデータだけを持つ。 テーブル構成を以下のように定義する。

```
CREATE TABLE 'users' (  
  'id' bigint(20) NOT NULL AUTO_INCREMENT,  
  'user_id' int(11) NOT NULL,  
  'name' varchar(64) DEFAULT '' NOT NULL,  
  PRIMARY KEY ('id')  
);
```

```
-- user1 user2がフレンド関係  
CREATE TABLE 'friend_link' (  
  'id' bigint(20) NOT NULL AUTO_INCREMENT,  
  'user1_id' int(11) NOT NULL,  
  'user2_id' int(11) NOT NULL,  
  PRIMARY KEY ('id')  
);
```

```
-- user1はuser2をblockしている  
CREATE TABLE 'block_list' (  
  'id' bigint(20) NOT NULL AUTO_INCREMENT,  
  'user1_id' int(11) NOT NULL,  
  'user2_id' int(11) NOT NULL,  
  PRIMARY KEY ('id')  
);
```

Chapter IV

Exercise 00 : get_friend_list

ユーザのフレンドのリスト(ID, 名前)を返す機能を実装せよ。

- エンドポイント
 - `‘/get_friend_list‘`
- パラメータ
 - ID: ユーザのID

Chapter V

Exercise 01 : get_friend_of_friend_list

ユーザのフレンドのフレンド(2hop)のリスト(ID, 名前)を返す機能を実装せよ。

- エンドポイント
 - `‘/get_friend_of_friend_list‘`
- パラメータ
 - ID: ユーザのID

Chapter VI

Exercise 02 : get_friend_of_friend_list_v2

以下の機能を前問で作成したエンドポイントに実装せよ。

- 前問の機能を満たしていること。
- フレンドのフレンドのリストに1hopのフレンドが含まれていた場合、それらを削除する機能。
- block関係であった場合は、フレンドのリストには含めない機能。

Chapter VII

Exercise 03 : get_friend_of_friend_list_paging

ページネーションを含めたユーザのフレンドのリストを返す機能を実装せよ。

- エンドポイントとパラメータ以外に関しては、前問の機能を満たしていること。
- エンドポイント
 - `‘/get_friend_of_friend_list_paging‘`
- パラメータ
 - ID: ユーザのID
 - limit: 1ページあたりの表示件数
 - page: ページの位置

Chapter VIII

Bonus: 新機能の追加

このミニマムSNSに何か一つ新機能を実装せよ。



テーブル定義の再設計やコードのリファクタリングなどをして構わない。