



# Road to Mixi

## Mini SNS 01

*Summary:* このドキュメントは、*Road to Mixi @ 42 Tokyo*の*Mini SNS 01*モジュール用の課題である。

**ROAD to** mixi GROUP

**42** | 東京

# Contents

<b>I</b>	<b>Introduction</b>	<b>2</b>
<b>II</b>	<b>Instruction</b>	<b>3</b>
<b>III</b>	<b>Prerequisite</b>	<b>4</b>
III.1	Repository Setup . . . . .	4
III.2	Docker . . . . .	4
III.3	Application . . . . .	5
<b>IV</b>	<b>Exercise 00 : Serve Static Content</b>	<b>6</b>
<b>V</b>	<b>Exercise 01 : Reverse Proxy &amp;&amp; Maintenance mode</b>	<b>7</b>
<b>VI</b>	<b>Bonus 00: Maintenance Mode advanced &amp;&amp; Serve error.html</b>	<b>8</b>
<b>VII</b>	<b>Bonus 01: Browser Caching &amp;&amp; Forwarded header</b>	<b>9</b>

# Chapter I

## Introduction

Road to Mixi Mini SNS 01ではミニマルSNSのnginxの設定を行ってもらいます。

# 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、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。
- 提出物を試せる環境を整えること。

# 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/problem2>

Road-to-Mixi-Mini-SNS-00の課題で取り組んだ言語に基づいて、[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 Application

とあるWebアプリケーションが

- L7ロードバランサー -> nginx -> Webアプリケーション

という構成で作成されている。

このWebアプリケーションにおいてnginxの設定・チューニングを行う。


このロードバランサーとnginx、Webアプリケーションは、172.31.0.0/16 のプライベートネットワーク内にある。

nginxプロセスが稼働しているサーバ（インスタンス）のドキュメントルート配下は以下のようなファイルツリーとなっている。

```
.
├── file
│   ├── 404.html
│   ├── error.html
│   └── maintenance.html
├── img
│   ├── image1.png
│   └── image2.png
└── index.html
```

# Chapter IV

## Exercise 00 : Serve Static Content

	Exercise 00
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : <i>nginx.conf, *</i>	

以下の挙動を行う設定ファイルを提出せよ。


- ‘/’にアクセスすると ‘index.html’を返すこと。
- 存在しないfileにアクセスした場合、 ‘file/404.html’を返すこと。
- ‘img/image1.png’、 ‘img/image2.png’にアクセスした場合、 返すこと。
- ‘/test’にアクセスした場合、 "this is a test"というデータを返すこと。



<http://nginx.org/en/docs/>

# Chapter V

## Exercise 01 : Reverse Proxy && Maintenance mode

	Exercise 01
提出するディレクトリ : <i>ex01/</i>	
提出するファイル : <i>nginx.conf, *</i>	


以下の挙動を行う設定を行うこと。

- 前問の機能を満たしていること。
- リバースプロキシするwebアプリケーションとして、Road-to-Mixi-Mini-SNS-00で作成したアプリケーションを設定すること。
- メンテナンスモードというものを設定することができ、メンテナンスモード中には全リクエストに対して常に‘file/maintenance.html’を返すこと。



# Chapter VI

## Bonus 00: Maintenance Mode advanced && Serve error.html


	Exercise 02
提出するディレクトリ : <code>ex02/</code>	
提出するファイル : <code>nginx.conf</code> , *	

以下の挙動を行う設定を行うこと。

- 前問の機能を満たしていること。
- メンテナンスモードでも特定のIPアドレス(任意)からのアクセスは通常通りにアクセスできること。
- メンテナンスモードでも `/test` は `"this is a test"` を返すこと（無条件）。
- メンテナンスモードの切り替えは無停止でできるようにすること。
- Webアプリケーションからエラーが返った場合に `error.html` を返すこと。

# Chapter VII

## Bonus 01: Browser Caching && Forwarded header

	Exercise 03
提出するディレクトリ : <i>ex03/</i>	
提出するファイル : <i>nginx.conf, *</i>	

以下の挙動を行う設定を行うこと。

- 前問の機能を満たしていること。
- nginx側の裏側のappにおいて、リクエスト元（クライアント）のIPアドレスをWebアプリケーション側で識別できるようにすること。



X-Forwarded-for ヘッダなどを利用する

- ‘img/image1.png’ と ‘img/image2.png’ への有効期限24時間のブラウザキャッシュを行うよう設定すること。