

Distributed, Concurrent Issue Board

Authors: Pedro Chem, Rafael Almeida de Bem

Table of contents

- [Distributed, Concurrent Issue Board](#)
 - [Table of contents](#)
 - [Prerequisites](#)
 - [Running](#)
 - [UNIX-based Systems](#)
 - [Windows](#)
 - [Example](#)
 - [Using the system](#)
 - [Modifying issues](#)
 - [Creating a new issue](#)
 - [Editing an issue](#)
 - [Deleting an issue](#)
 - [Viewing the board](#)
 - [JSON format](#)
 - [Plain text format](#)
- [Concurrency Problems](#)

Prerequisites

This project uses Logrus as a logging framework. To install Logrus, type this in your shell: `go get "github.com/sirupsen/logrus"`.

Running

UNIX-based Systems

```
go build && ./NoticeBoard [args]
```

Alternatively:

```
go run *.go [args]
```

Windows

```
go build  
.\NoticeBoard.exe [args]
```

Example

```
go run *.go localhost:4007 localhost:4008 localhost:4009
```

Where **args** is a list of **IP:Port** pairs, with the first pair being the host (you).

Using the system

When you first run the program, you are greeted by a prompt asking you for your username. After entering your desired username, the system will sleep for around two seconds while it waits for a network message from other users (this message contains the current board from that user).

After that, the system will be ready to go; either with a brand new, blank, board or with an existing board from another user.

Modifying issues

Creating a new issue

To create a new issue, all you need to do is enter the correct number in the main menu, which will prompt you for the issue's contents. Then the issue will be appended to the current board and the whole board will be broadcast to the other connected users.

Editing an issue

To edit an existing issue, you will need to enter the issue's ID, which is comprised of the issue's creator's username and the issue's number. When you pick a valid issue, you'll be able to edit the contents of that issue and it will be edited in the current board, which will be broadcast to the other users.

Deleting an issue

Much like editing an issue, to select an issue you need to input the issue's creator's username and its respective number. Then, the issue will be deleted (removed from the board) and the updated board will be broadcast to all connected users.

Viewing the board

The system allows the user to view the board in two distinct ways: either in JSON format or an arbitrary format chosen by the developer (me).

JSON format

Since JSON is both human-readable and machine-readable, and the Go language has native support for this format, the board structure is printed in the console as a JSON. It's also sent via network messages as an encoded JSON structure and decoded when it's received by the end user. A mock JSON structure is as follows:

```
{  
  "Issues": [  
    {
```

```

    "ID": {
      "CreatorID": "Rafael",
      "IssueNumber": 1
    },
    "Content": "Example issue 1"
  },
  {
    "ID": {
      "CreatorID": "Rafael",
      "IssueNumber": 2
    },
    "Content": "Example issue 2"
  }
],
"TimeStamp": "2019-11-18T21:50:48.7468793-03:00"
}

```

Notice the issues array in the structure, contained in the "Issues" key. Each issue's ID is comprised of the issue's creator's ID (or username) and its creation number. The issue itself is a structure that contains an issue ID and a string that contains its contents. The board also contains a timestamp that helps in the synchronization functions of the system.

Plain text format

Issues being displayed as plain text format are encoded in the following way: [{issue creator}-{issue number}] : {issue content}

```

[Rafael-1] : Example issue 1
[Rafael-2] : Example issue 2

```

Concurrency Problems

The system does not, in any way, shape or form, guarantee that all users will have the same board at all times. Most synchronization problems have been eliminated but some edge cases may still occur. For example, if there are two users in different time zones, the board of the user in the "greater" time zone will always be treated as the most current version of all the boards, even though that might not be the case. Furthermore, if two users try to edit the same issue at the same time, they both will send out their boards and one of the users will be left with the wrong board.

There are no problems with shared memory in this system because there is no shared memory. This system works with network messages and JSON strings and nothing else.