



**BITTIGER**

**Teamcode**

An integrated online collaborate judge system

Xinxin He



Why Teamcode?

Major Use Cases

UI Design / Demo

System Architecture

Process Flows

# Why Teamcode?

- Programming is becoming more and more popular.
- Team work is gaining attention in the coding world.
- Most of the online collaborative programming platform only allows to work with friends, or share by link.
- Teamcode allows developers to solve coding problems with whoever available online, with real-time communicating and collaborative code editing feature.

# Major User Cases

Pick up a problem

Select a room or start a new room

Enter nickname

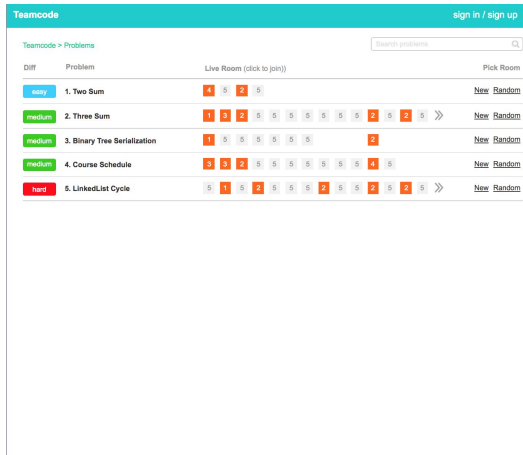
Color is assigned

Timer starts

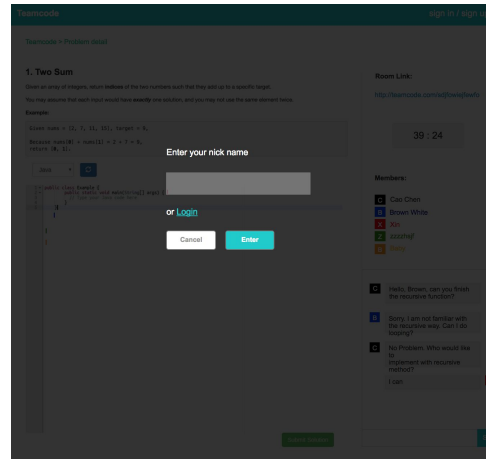
Chat with room members

Code collaboratively

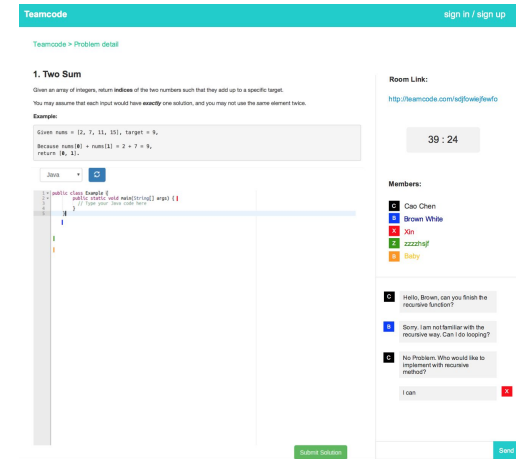
# UI Design – Problem List



Problem list with active rooms



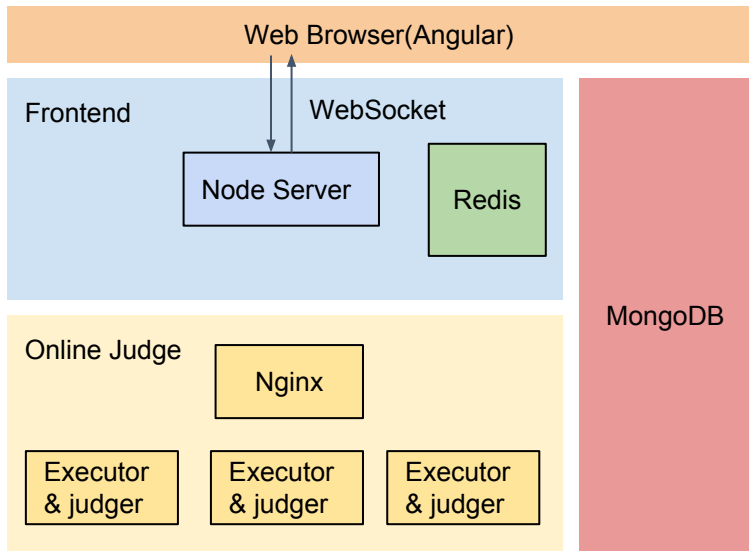
Enter nickname



Problem editor and message box

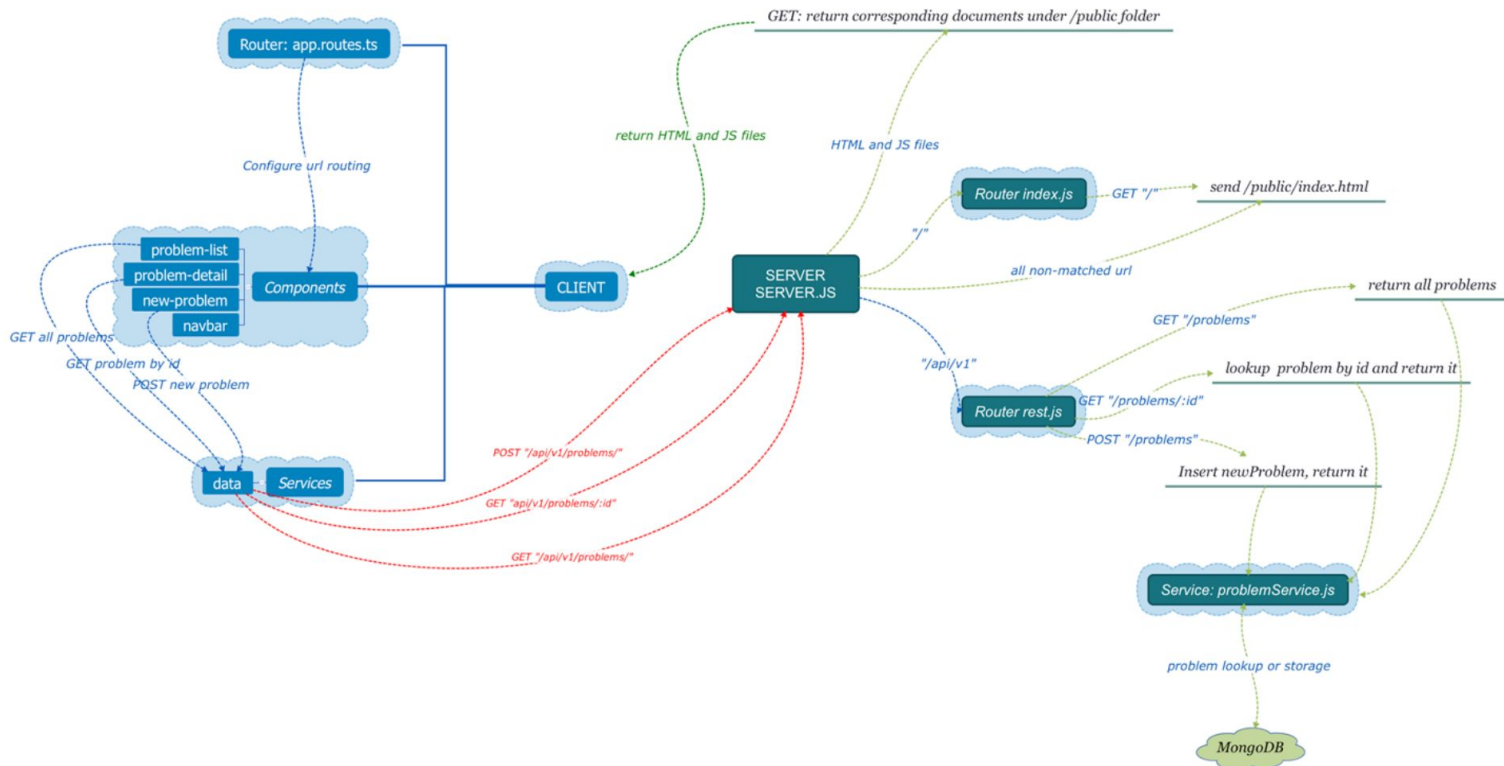
# Demo

# Technology stacks



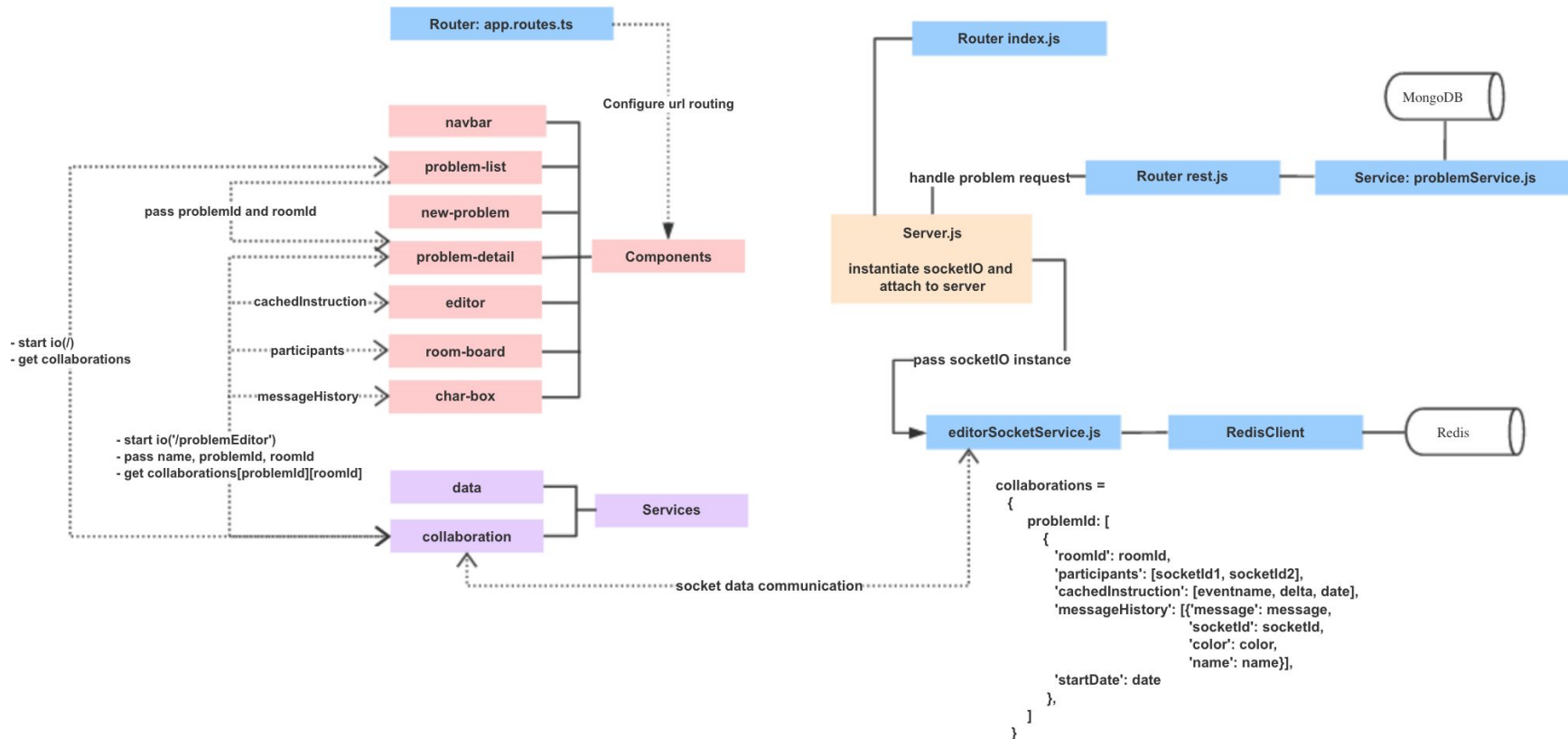
Stack	Technologies
Frontend - client	<b>Angular4</b> : views, components, services <b>Socket.io</b> : maintain and response client-side socket events for coding room, code editor, message box, timers, room members, etc.
Frontend - server	<b>Node.js</b> : javascript engine <b>Express</b> : Node.js web framework <b>Socket.io</b> : maintain collaboration data structure and a list of socket events <b>Redis</b> : cache codes when all of the users leave room <b>MongoDB</b> : store problem list and details
Backend - executor	<b>Nginx</b> : load-balancer <b>Flask</b> : backend framework for code executing <b>Docker</b> : code executing environment

# REST API Process





# Socket Communication



# Socket Communication

## problem-list component

- init IO
- emit “getCollaborations” event
- get collaborations through rxjs
- match problem id and list active rooms

```
collaborations =  
{  
  problemId: [  
    {  
      'roomId': roomId,  
      'participants': [socketId1, socketId2],  
      'cachedInstruction': [eventName, delta, date],  
      'messageHistory': [{  
        'message': message,  
        'socketId': socketId,  
        'color': color,  
        'name': name}],  
      'startDate': date  
    },  
  ],  
}
```

Teamcode

Sign in Sign up

Teamcode > Problem list

Search

Search

Problem Name

Input your problem name

Problem Description

Input your problem description

Difficulty

easy

Submit

super	1. Three Sum	1 2	New
hard	2. Triangle Count		New
easy	3. test		New

# Socket Communication

problem-detail component

- init IO('/problemEditor')
- pass name, problemId, roomId to backend socket service
- emit "getParticipants" event
- assign color in the backend
- emit all participant data back to the client
- emit "sendMessage" event
- get messageHistory
- emit "cursorChange" event
- get color
- get cursor Change and update cursor
- emit "cachedInstruction" event
- get cached code content or code change

Teamcode

Sign In Sign up

Teamcode > Problem list

### 1. Three Sum

Given an array of n integer with duplicate number, and a moving window(size k), move the window at each iteration from the start of the array, find the maximum number inside the window at each moving.

Java Reset

```
1 public class Example {
2     public static void main(String[] args) {
3         // Type your Java code here
4     }
5 }
```

Submit Solution

#### Room Link

http://localhost:3000/problems/1

53 s

#### Room Members

A Alice

J Julia

J Hello This is Julia

A Hello, This is Alice

Input Message...

Send