**Overview**

This document outlines the specification for the Scoreboard API module, which manages real-time updates for a website displaying the top 10 user scores. The API is responsible for receiving score updates, ensuring data integrity, and broadcasting real-time changes.

**Features**

1. **Scoreboard Display**: Retrieves the top 10 users with the highest scores.
2. **Live Updates**: Uses WebSockets to update the scoreboard in real time.
3. **Score Update API**: Users can trigger an API call to update their scores upon completing an action.
4. **Security Measures**: Prevents unauthorized score manipulations.

**API Endpoints**

**1. Get Top 10 Scores**

**Endpoint:** GET /api/scoreboard

**Description:** Returns the top 10 users with the highest scores.

**Response:**

{

"scoreboard": [

{"id": 1, "username": "Player1", "score": 5000},

{"id": 67, "username": "Player67", "score": 4800},

…..

]

}

**2. Update Score**

**Endpoint:** POST /api/score/update

**Description:** Updates a user's score when they complete an action.

**Request Body:**

{

"id": 1

"action": "play"

"point": : 200

}

**Response:**

{

"message": "Score updated successfully",

"new\_score": 5050

}

**Security Measures:**

* **Action Token Verification**: Each request must include a signed action token that the backend verifies before updating the score.
* **Rate Limiting**: Limits score updates to prevent spamming.
* **Server-side Validation**: Ensures the action triggering the score update is legitimate.

**3. Live Updates (WebSockets / SSE)**

**Endpoint:** GET /api/scoreboard/live

**Description:** Streams real-time updates whenever a score changes.

**WebSocket Message Example:**

{

"id": 1,

"username": "Player1",

"new\_score": 5050

}

**Execution Flow**

graph TD;

User 🡪 Performs Action 🡪 Server

Server 🡪 Verifies Action Token 🡪 Database

Database 🡪 Updates Score 🡪 Server

Server 🡪 Sends Updated Score 🡪 WebSocket Server

WebSocket Server 🡪 Broadcasts Update 🡪 Website

**Comments for Improvement**

1. **Cache Optimization**: Use Redis to store the top 10 scores to reduce database queries.
2. **JWT Authentication**: Require users to authenticate via JWT to send update requests.