



Last indexed: 27 August 2025 (c12f7a)

PadelFlow Overview

Core Application Architecture

Server Setup and Configuration

User Roles and Authentication

Tournament Management
Features

Real-time Features

Database Layer

SQLite Database Management

Database Extensions

Qt Framework Components

Image Format Support

Development Environment

IDE Configuration

Debugging Setup

Project Configuration

Tournament Management Features

Relevant source files

Purpose and Scope

This document covers the tournament management system within PadelFlow, detailing how tournaments are created, configured, and executed across different tournament formats. It focuses on the core business logic for tournament operations, user workflows, and the underlying code architecture that supports various tournament types including Round Robin, Single Elimination, Liga, and Americana Clásica formats.

For information about real-time score updates and live dashboard features, see [Real-time Features](#). For user authentication and role-based access control, see [User Roles and Authentication](#).

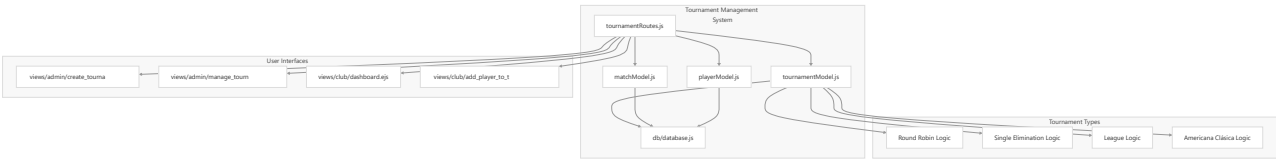
Tournament Types and System Architecture

PadelFlow supports four distinct tournament formats, each with specific business logic and execution patterns:

Supported Tournament Formats

Tournament Type	Spanish Name	Implementation Status	Key Characteristics
Round Robin	Liguilla	Partially Implemented	All teams play each other, table-based ranking
Single Elimination	Eliminatoria Directa	Partially Implemented	Bracket-style knockout format
League	Liga	Planned	Round Robin with home/away matches
Classic American	Americana Clásica	Planned	Rotating partnerships, individual scoring

Tournament Type Architecture



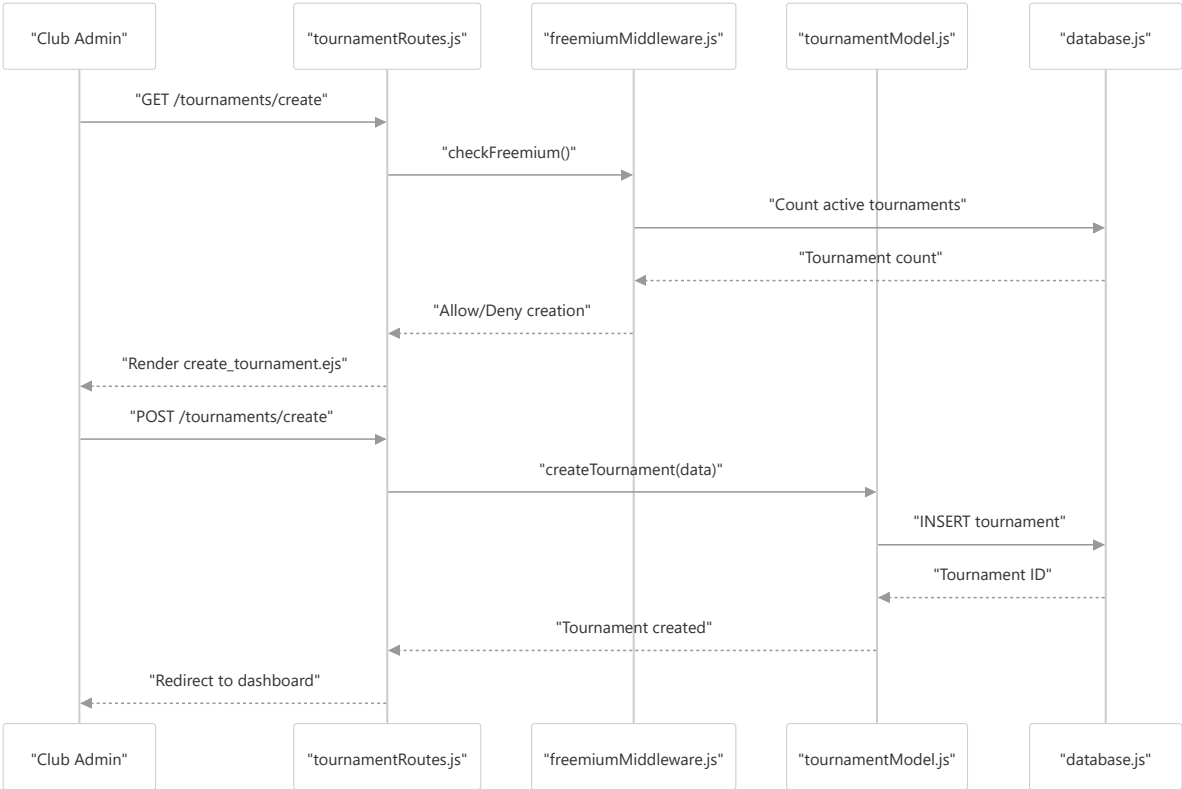
Sources: README.md | 199-216 routes/tournamentRoutes.js models/tournamentModel.js

models/matchModel.js

Tournament Lifecycle and User Workflows

The tournament management system follows a structured lifecycle from creation to completion, with different user roles having specific responsibilities at each stage.

Tournament Creation Workflow

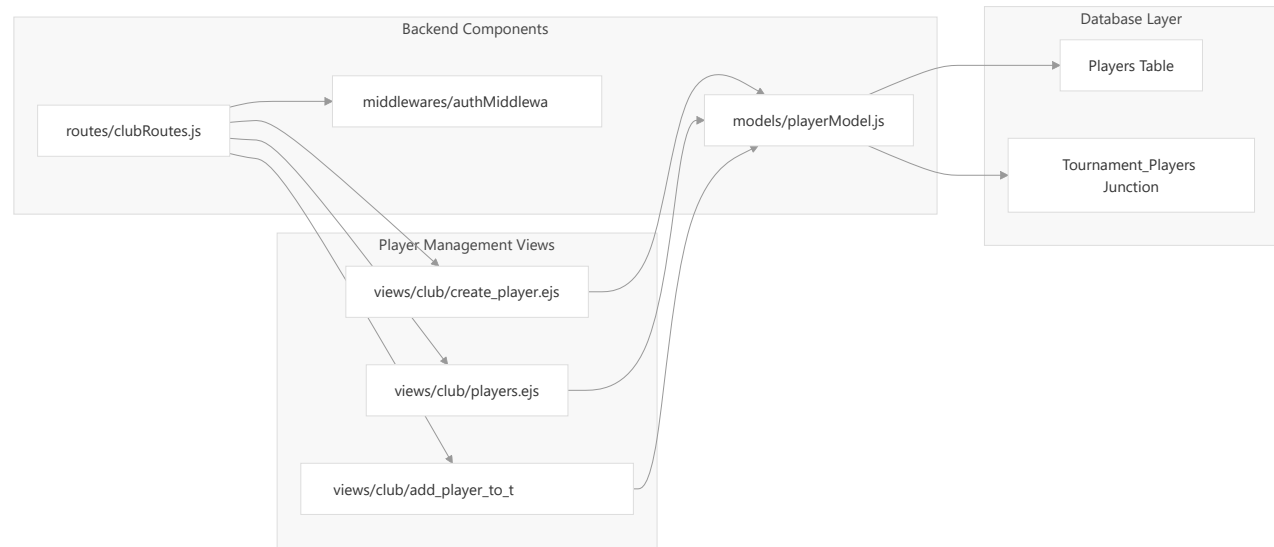






Sources:  routes/tournamentRoutes.js | 131-132  middlewares/freemiumMiddleware.js | 121-124
 models/tournamentModel.js

Player Management in Tournaments

The system provides comprehensive player management capabilities within tournaments, allowing club administrators to manage participant registration and tournament assignments.

Player Management Components

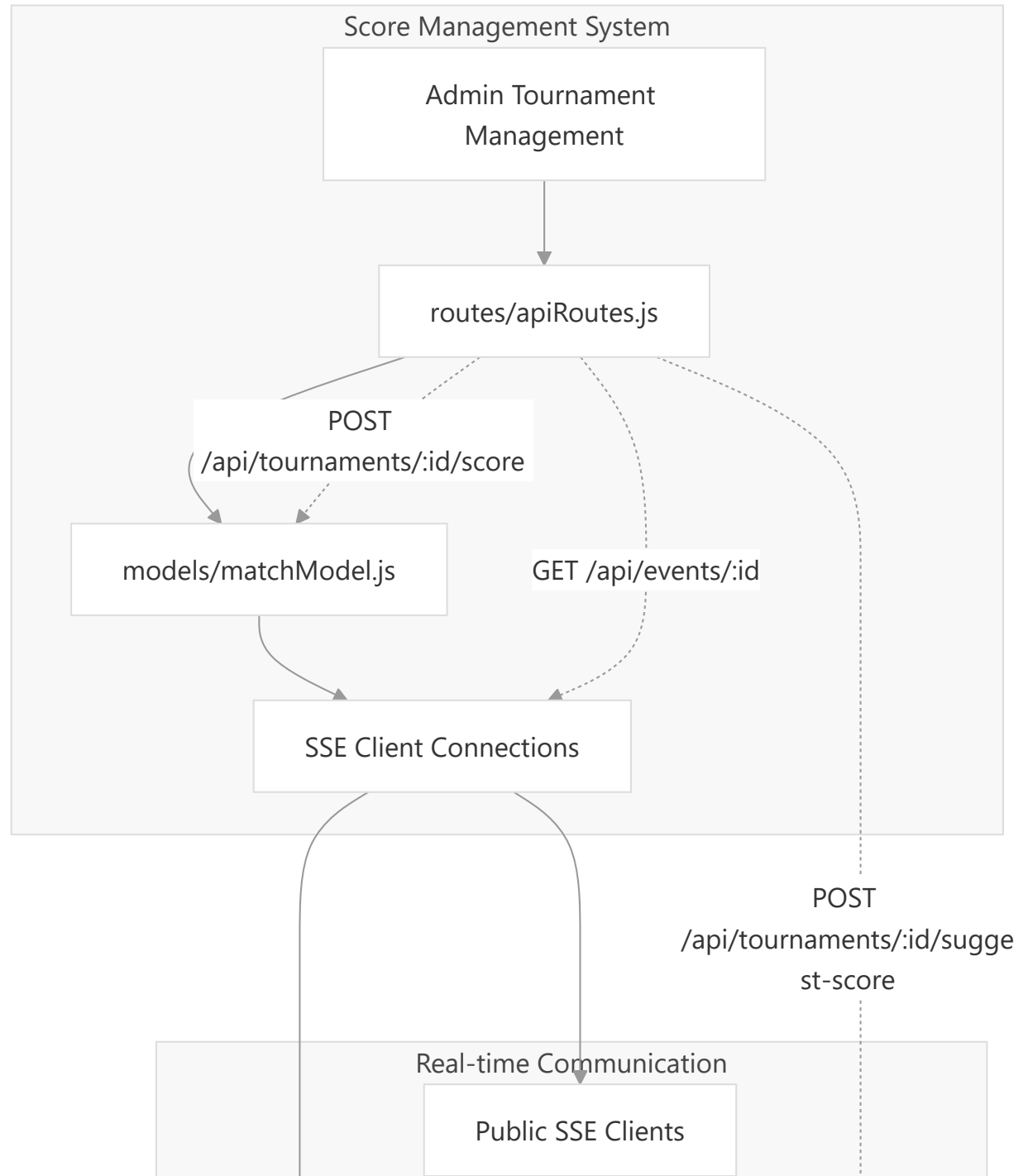


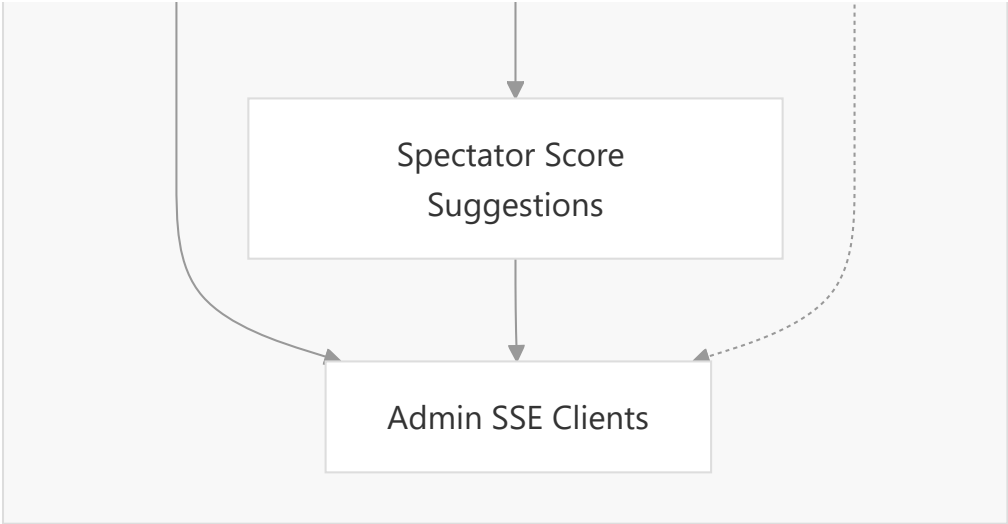
Sources:  `views/club/create_player.ejs`  `views/club/players.ejs`
 `views/club/add_player_to_tournament.ejs`  `models/playerModel.js`

Tournament Execution and Score Management

The tournament execution system handles live score updates, match progression, and real-time notifications to both administrators and spectators.

Score Update Architecture





Sources: routes/apiRoutes.js | 139-140 models/matchModel.js
 public/js/sse-client.js | 142-144

Code Organization and Key Components

The tournament management functionality is distributed across multiple layers following MVC architecture patterns.

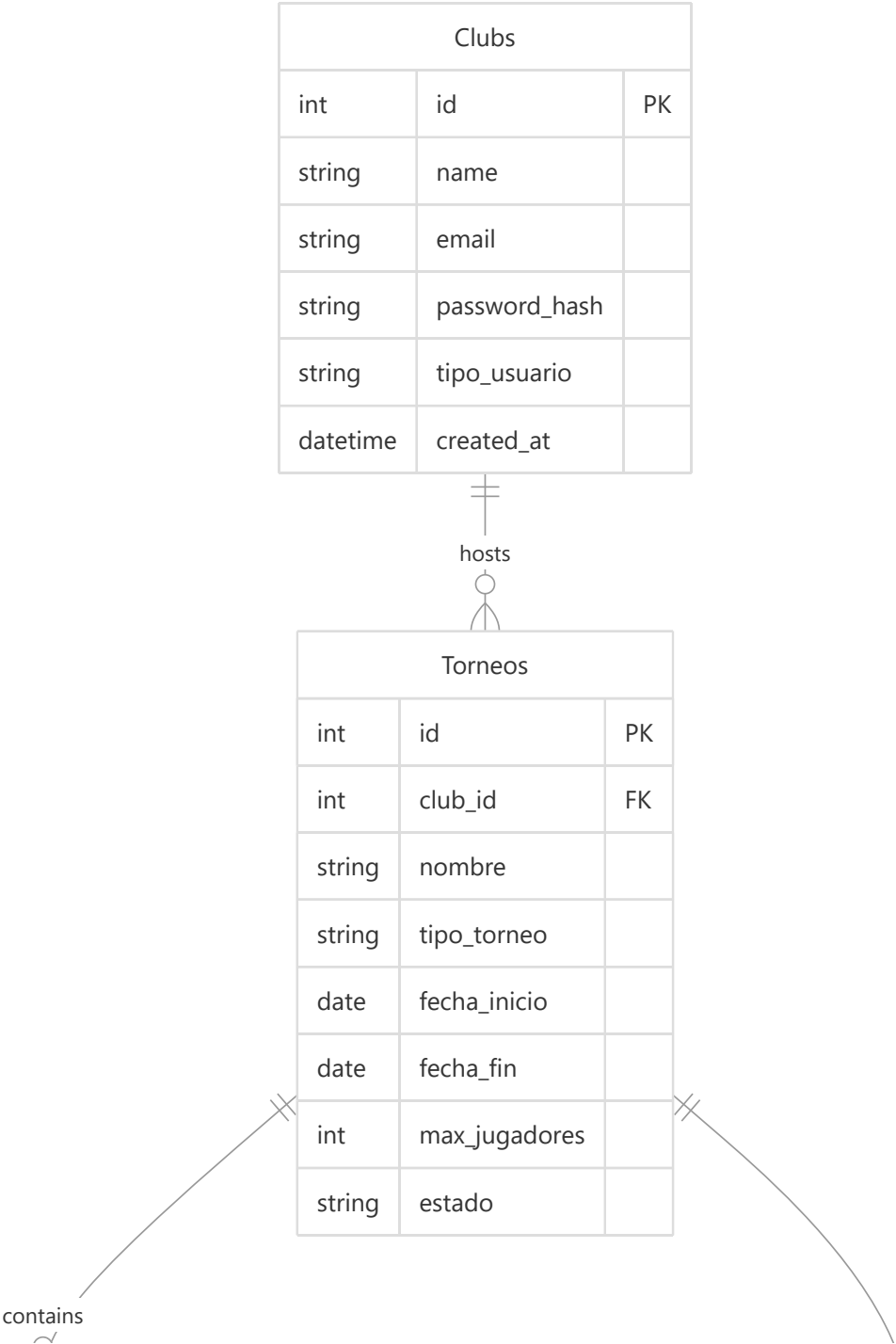
Core Tournament Management Files

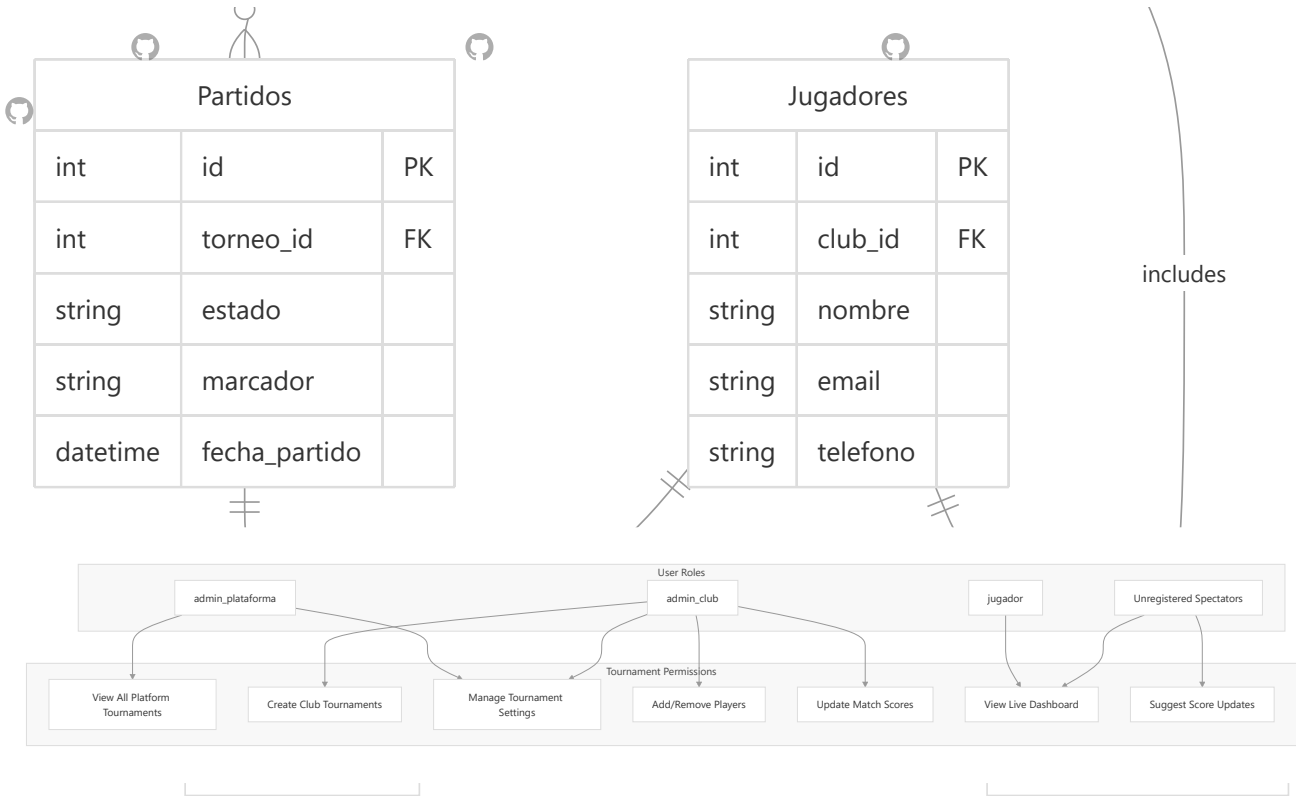
Component	File Path	Primary Responsibility
Routes	routes/tournamentRoutes.js	HTTP request handling, middleware application
Models	models/tournamentModel.js	Tournament CRUD operations
Models	models/matchModel.js	Match scoring and progression logic
Models	models/playerModel.js	Player registration and management

Component	File Path	Primary Responsibility
Views	views/admin/create_tournament.ejs	Tournament creation interface
Views	views/admin/manage_tournament.ejs	Tournament administration interface
Views	views/club/dashboard.ejs	Club-level tournament overview
Middleware	middlewares/freemiumMiddleware.js	Tournament limit enforcement

Database Schema Integration

The tournament system relies on several interconnected database tables managed through SQLite:





Sources: README.md | 6-11 middlewares/authMiddleware.js | 116-119
 middlewares/freemiumMiddleware.js | 121-124

Freemium Model Integration

The tournament management system includes built-in support for freemium limitations, restricting the number of active tournaments for non-premium club accounts.

Freemium Enforcement Mechanism

The `checkFreemium` middleware enforces tournament creation limits:

- **Free Tier:** Maximum 3 active tournaments per club
- **Premium Tier:** Unlimited tournaments (implementation pending)
- **Enforcement Point:** Tournament creation endpoint

The middleware integrates at the route level in `tournamentRoutes.js`, specifically protecting the tournament creation workflow.

Sources:  `middlewares/freemiumMiddleware.js` | 121-124  `routes/tournamentRoutes.js` | 131
 `README.md` | 121-124

Integration with Live Dashboard System

Tournament management is closely integrated with the real-time dashboard system, providing live updates to spectators and administrators during tournament execution.

Live Update Integration Points

1. **Score Updates:** Match score changes trigger SSE broadcasts
2. **Tournament Progression:** Bracket advancement notifications
3. **Spectator Engagement:** Score suggestion system for public viewers
4. **Administrative Alerts:** Real-time notifications for tournament managers

The integration leverages the SSE client system defined in `public/js/sse-client.js` and the API routes in `routes/apiRoutes.js` to maintain real-time synchronization between tournament state and viewer interfaces.

Sources:

 routes/apiRoutes.js	134-144	 public/js/sse-client.js	142-144
 views/public/live_dashboard.ejs	158-160		