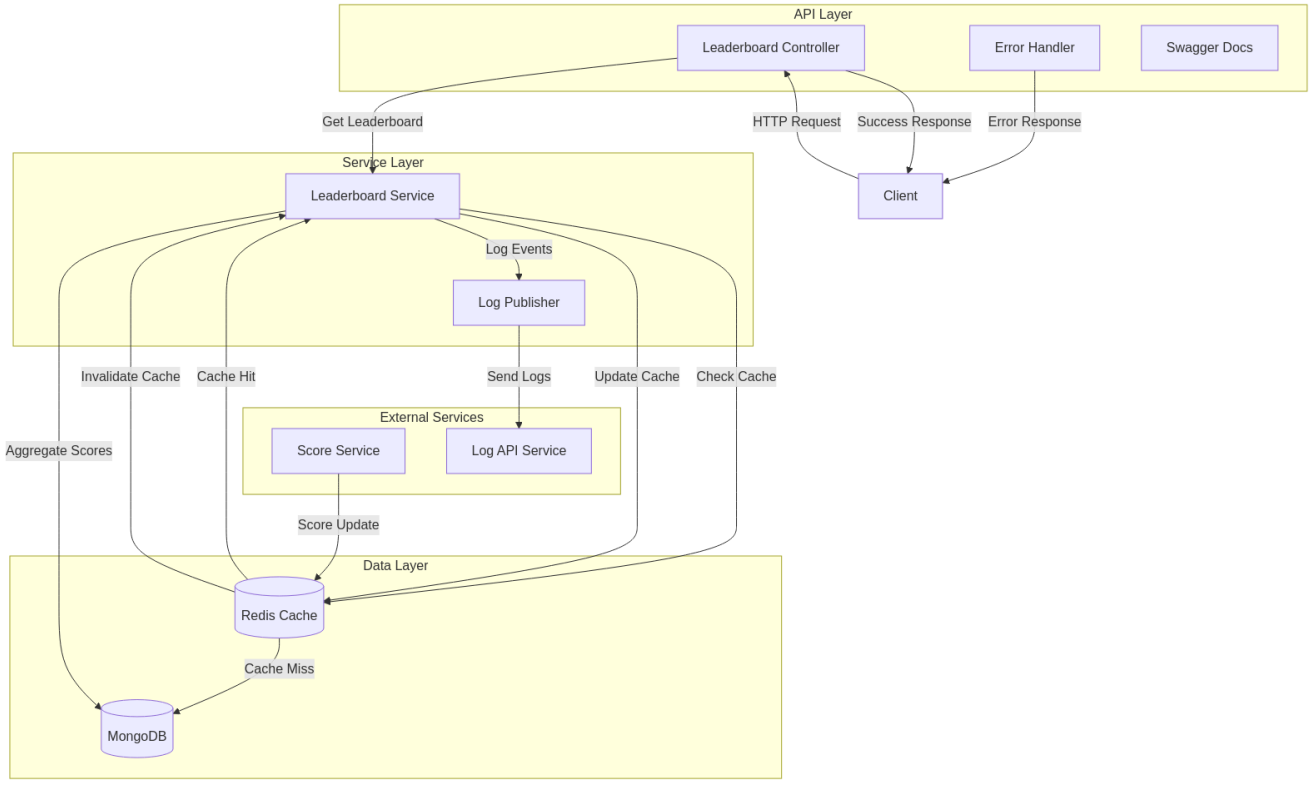
Leader Board Service

Here's how the system works:

1. **API Layer**:
   - Express-based REST API with Swagger documentation
   - Routes for leaderboard retrieval
   - Error handling middleware
2. **Service Layer**:
   - `LeaderboardService` handles business logic
   - Caching strategy with Redis
   - MongoDB aggregation for score calculations
   - Logging through log-api-service
3. **Data Layer**:
   - MongoDB for score aggregation
   - Redis for caching leaderboard pages
   - Mongoose for database connection
4. **Features**:
   - Paginated leaderboard
   - Caching with TTL
   - Score aggregation
   - Real-time cache invalidation
   - Logging

Here's the mermaid diagram illustrating the flow:



Key Features of the Leaderboard Service:

1. **Leaderboard Management**:
   - Paginated leaderboard retrieval
   - Score aggregation by player
   - Sorting by total score
   - Efficient caching strategy
2. **Caching Strategy**:
   - Redis-based caching
   - Page-based caching
   - TTL-based cache invalidation
   - Cache invalidation on score updates
3. **Data Aggregation**:
   - MongoDB aggregation pipeline
   - Grouping by playerId
   - Summing scores
   - Sorting by total score
4. **API Endpoints**:
   - GET /players/leaderboard - Get paginated leaderboard
5. **Cache Configuration**:
   - 1-minute TTL for leaderboard pages
   - Page-based cache keys
   - Automatic cache invalidation
   - Cache miss handling

6. **Error Handling**:
    - Database errors
    - Cache errors
    - Input validation
    - Error logging
7. **Logging**:
    - Cache hit/miss logs
    - MongoDB retrieval logs
    - Error logs
    - Different log types (INFO, ERROR)
8. **Documentation**:
    - Swagger documentation
    - API endpoint descriptions
    - Response schemas
    - Example values

The service follows best practices for:

- Caching strategy
- Data aggregation
- Error handling
- Logging
- Documentation
- Testing

The architecture ensures:

- Fast leaderboard retrieval
- Efficient score aggregation
- Real-time updates
- Data consistency
- Reliable logging
- Clear API documentation
- Proper error handling

The leaderboard service is designed to handle high volumes of score updates while maintaining fast leaderboard retrieval through efficient caching and aggregation strategies.