

# Score Service

Here's how the system works:

## 1. API Layer:

- Express-based REST API with Swagger documentation
- Routes for score submission and leaderboard retrieval
- Input validation using Joi
- Error handling middleware

## 2. Service Layer:

- `ScoreService` handles business logic
- Real-time score updates using Redis pub/sub
- Leaderboard management using Redis sorted sets
- Logging through log-api-service

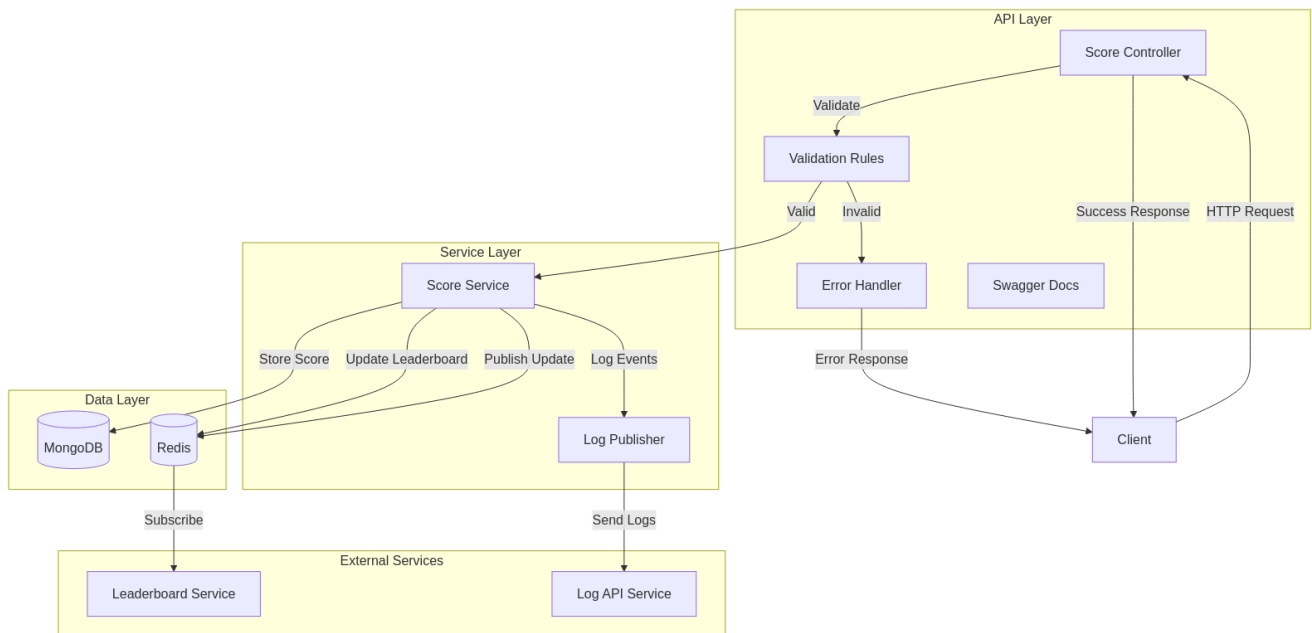
## 3. Data Layer:

- MongoDB for persistent storage of scores
- Redis for real-time leaderboard and pub/sub
- Mongoose for data modeling

## 4. Features:

- Real-time score updates
- Leaderboard functionality
- Input validation
- Error handling
- Swagger documentation
- Logging

Here's the mermaid diagram illustrating the flow:



## Key Features of the Score Service:

### 1. Score Management:

- Score submission with validation
- Persistent storage in MongoDB
- Real-time leaderboard updates
- Score update notifications

### 2. Leaderboard Functionality:

- Redis sorted sets for efficient ranking
- Top scores retrieval
- Real-time updates
- Efficient querying

### 3. Real-time Updates:

- Redis pub/sub for score updates
- Leaderboard service subscription
- Immediate score propagation

### 4. API Endpoints:

- POST /scores - Submit score
- GET /scores/top - Get top scores

### 5. Data Model:

- Score schema with playerId and score
- Timestamps for created/updated
- Required fields validation

### 6. Error Handling:

- Input validation errors
- Database errors
- Redis errors

- 400 for bad requests

#### 7. **Logging:**

- Score submission logs
- Leaderboard retrieval logs
- Error logs
- Different log types (INFO, ERROR)

#### 8. **Documentation:**

- Swagger documentation
- API endpoint descriptions
- Request/response schemas
- Example values

The service follows best practices for:

- Real-time updates
- Data consistency
- Input validation
- Error handling
- Logging
- Documentation

The architecture ensures:

- Real-time score updates
- Efficient leaderboard queries
- Data persistence
- Reliable logging
- Clear API documentation
- Input validation
- Proper error handling

The score service is designed to handle high volumes of score submissions while maintaining real-time leaderboard updates and ensuring data consistency across the system.