Team Management App Tutorial

# Setting Up the Next.js Project

To create a new Next.js project, run:  
**npx create-next-app@latest**

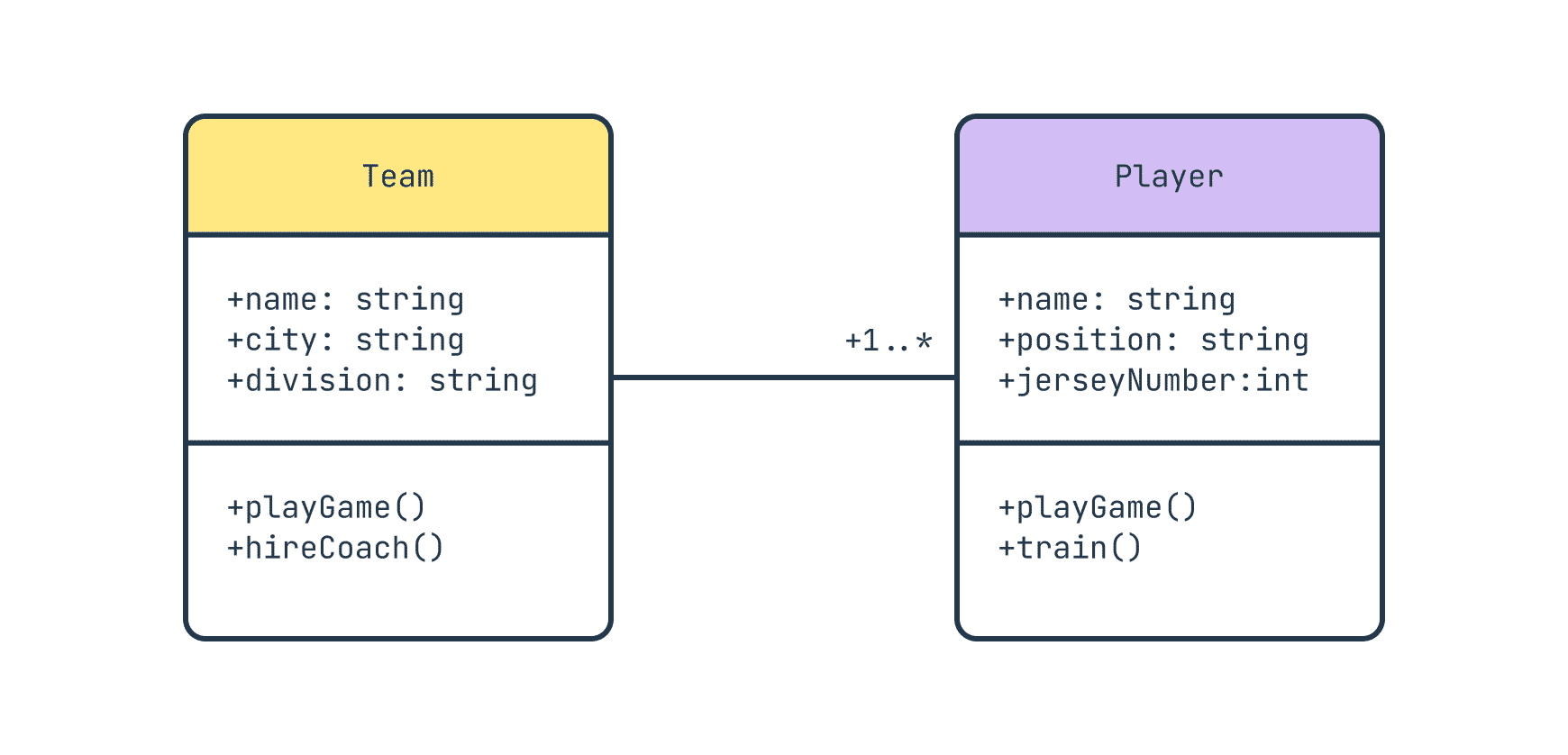
# Installing Prisma and Setting Up the Database

Add Prisma to your project as a dev dependency:  
**npm install prisma --save-dev**  
  
Install the Prisma client for database interaction:  
**npm install @prisma/client**

Install the FS Extra for reading the json files:

**npm install fs-extra**  
  
Initialize a new Prisma project with SQLite as the datasource provider:  
**npx prisma init --datasource-provider sqlite**

**Create the following schema**



# Creating and Migrating the Database

Create the initial migration for your database schema with:  
**npx prisma migrate dev --name init**

# Exploring the Database with Prisma Studio

Use Prisma Studio to visually interact with your database:  
**npx prisma studio**

# Generating Prisma Client

Generate the Prisma client for use in your application with:  
**npx prisma generate**

# Creating the Repository Methods to add update delete and write into team and player tables

# Creating the API routes for Team and Player CRUD Operations

# Team Routes

## GET /api/teams

Returns all teams.

## POST /api/teams

Creates a new team with data provided in the request body.

## GET /api/teams/{teamId}

Returns the team with the specified teamId.

## PUT /api/teams/{teamId}

Updates the team with the specified teamId using data provided in the request body.

## DELETE /api/teams/{teamId}

Deletes the team with the specified teamId.

# Player Routes (Accessed under a Team)

## GET /api/teams/{teamId}/players

Returns all players in the team with the specified teamId.

## POST /api/teams/{teamId}/players

Adds a new player to the team with the specified teamId using data provided in the request body.

## GET /api/teams/{teamId}/players/{playerId}

Returns the player with the specified playerId under the team teamId.

## PUT /api/teams/{teamId}/players/{playerId}

Updates the player with the specified playerId within the team teamId using data provided in the request body.

## DELETE /api/teams/{teamId}/players/{playerId}

Removes the player with the specified playerId from the team teamId.