

# SPORTS TEAM

## TypeScript Interfaces & Classes

**Task:** Create a Player interface and a Team class that work together in TypeScript.

**Setup & Submission:** Create a new TypeScript project on [Replit.com](https://replit.com). Submit in the LMS by pasting a link to your Repl.

### Build Specifications

First, in a **Player.ts** file, create and export an interface called Player that has the following properties:

- name (string)
- jersey (number)
- active (boolean)

Next, in a **Team.ts** file, create and export a class called Team with the following members:  
Properties:

- players (array of Player) - starts as an empty array
- teamName (a string)

Methods:

- constructor: use a parameter to set the teamName
- addPlayer method, which takes a Player object as a parameter and adds it to the list
- removePlayer, which takes an index of a player and removes it from the list.
- setActive, which takes an index of a player, and a boolean for active or inactive, and stores the boolean in that player's active member.
- getPlayerCount: Takes no parameters; returns the number of players.
- logActivePlayers: Log to the console, one per line, a list of players whose active status is set to true
- getActivePlayers: Returns an array of players whose active status is set to true

Manually test your code by exercising each method in **index.ts**. Create one or more Team instances, add players, and call the various methods.

### Extended Challenges

Add additional methods to Team:

- removePlayerByName: removes a player by name rather than index. You can loop through the array to find a match; or you can explore the [findIndex function](#).
- sortByName: Sort the players alphabetically by name. For some tips, see this [stack overflow](#).
- toString: return a string that includes the team name and a list of the active players.