

- [PlayerService](#)
 - [Project Structure](#)
 - [Features](#)
 - [API Endpoints](#)
 - [1. Create a Player](#)
 - [2. Get All Players](#)
 - [Unit Tests](#)
 - [Overview](#)
 - [Test Cases:](#)
 - [CreatePlayerAsync_WithValidName_ShouldCreatePlayer :](#)
 - [CreatePlayerAsync_WithDuplicateName_ShouldThrowArgumentE
xception :](#)
 - [CreatePlayerAsync_WithNoName_ShouldGenerateUniqueld :](#)
 - [GetPlayersAsync_ShouldReturnListOfPlayers :](#)

PlayerService

Project Structure

```
PlayerService/  
├── Controllers/  
│   └── PlayerController.cs      # Handles HTTP requests for player-related  
operations  
├── Models/  
│   └── Player.cs                # Represents the Player entity  
├── Services/  
│   ├── IPlayerService.cs       # Defines the player service interface  
│   └── PlayerService.cs        # Implements the player service, handles business  
logic  
├── Repositories/  
│   ├── IPlayerRepository.cs    # Defines the repository interface for data  
access  
│   └── PlayerRepository.cs      # Implements the repository, interacts with Redis
```

Features

- **Player Creation:**

- Players can be created via a REST API.
- A player can be assigned a name, or a unique identifier will be generated if no name is provided.
- Validates the uniqueness of player names; if a name already exists, a **400 BadRequest** is returned with an appropriate error message.
- **Player Retrieval:**
 - Fetches the list of all created players from Redis.
 - Player data is persisted in Redis to ensure it remains available across multiple sessions.

API Endpoints

1. Create a Player

- **URL** : **POST** `/api/player`
- **Description** : Creates a new player with an optional name.
- **Request Parameters** :
 - **playerName** (query string, optional): The name of the player. If not provided, a unique name will be generated.
- **Response** :
 - **200 OK**: Player successfully created.
 - **400 BadRequest**: If the **playerName** is already taken.

Example Request (with player name):

```
curl -X POST "https://localhost:7008/api/player?playerName=Mehran"
```

Example Request (without player name):

```
curl -X POST "https://localhost:7008/api/player"
```

Example Response (success):

```
{  
  "id": "1635de3b-1883-40da-a34e-ceffd6fa321b",  
}
```

```
"name": "Mehran"
}
```

Example Response (error - name already taken):

```
{
  "message": "The player name is already taken."
}
```

2. Get All Players

- **URL** : **GET** `/api/player`
- **Description** : Fetches the list of all players.
- **Response** : **200 OK** with a list of players.

Example Request :

```
curl -X GET "https://localhost:7008/api/player"
```

```
[
  {
    "id": "a18a4ce4-00f0-4721-a79f-c0c507e25005",
    "name": "SW0HxegCUd"
  },
  {
    "id": "37014d01-0098-45a7-987e-49250d2774e6",
    "name": "mehran2"
  },
  {
    "id": "d2a56739-8605-4e6b-af93-1f78a747c419",
    "name": "GZR1W5xfkt"
  },
  {
    "id": "3d9c0cad-01d0-4aa2-bd8e-1fc72285f7d7",
    "name": "te+mFFVe+k"
  },
  {
    "id": "1af19130-5f87-424d-aa06-7a2500d8ed11",
    "name": "y3vlzRIItBs"
  },
  {

```

```
    "id": "49fc12c1-0536-4488-a757-eeaca67d1fbd",
    "name": "PsaIRQUP7W"
  },
  {
    "id": "4a8684a4-8d20-4896-96ef-5e751616c77b",
    "name": "mehran"
  }
]
```

Unit Tests

Overview

Unit tests are implemented using **xUnit** with **Moq** for mocking dependencies. The tests ensure that the key features of **PlayerService** behave as expected.

Test Cases:

CreatePlayerAsync_WithValidName_ShouldCreatePlayer :

- Validates that a player is created with the specified name and is added to the repository.

CreatePlayerAsync_WithDuplicateName_ShouldThrowArgumentException :

- Ensures that creating a player with an already taken name results in an exception.

CreatePlayerAsync_WithNoName_ShouldGenerateUniqueId :

- Ensures that if no player name is provided, a unique name prefixed with **"Guest_"** is generated.

GetPlayersAsync_ShouldReturnListOfPlayers :

- Validates that the service returns the correct list of players from the repository.

Test Explorer



Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 173 ms

Test	Duration
PlayerService.Tests (4)	375 ms
player_service.Tests (4)	375 ms
PlayerServiceTests (4)	375 ms
CreatePlayerAsync_WithDuplicateName_ShouldThrowArgumentExce...	2 ms
CreatePlayerAsync_WithNoName_ShouldGenerateUniqueld	53 ms
CreatePlayerAsync_WithValidName_ShouldCreatePlayer	317 ms
GetPlayersAsync_ShouldReturnListOfPlayers	3 ms