

Team Model

```
// import mongoose module
const mongoose = require("mongoose");

const teamSchema = mongoose.Schema({
  name: String,
  players: [
    {
      type: mongoose.Schema.Types.ObjectId,
      ref: 'Player'
    }
  ]
});

// create Model Name "Team"
const team = mongoose.model("Team", teamSchema);

// make team exportable
module.exports = team;
```

Player Model

```
// import mongoose module

const mongoose = require("mongoose");

const playerSchema = new mongoose.Schema({
  name: String,
  nbr: Number,
  age: Number,
  team: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Team'
  }
});

// create Model Name "Player"
const player = mongoose.model("Player", playerSchema);

// make player exportable
module.exports = player;
```

Business Logic : Add Player

```
// Business Logic : Add Player
app.post("/api/players", (req, res) => {
  try {
    Team.findById(req.body.teamId).then((team) => {
      if (!team) {
        return res.status(404).json({ message: "Team not found" });
      }

      const player = new Player({
        name: req.body.name,
        nbr: req.body.nbr,
        age: req.body.age,
        team: team._id,
      });

      player.save((err, doc) => {
        team.players.push(player);
        team.save();
        res.status(201).json(player);
      });
    });
  } catch (error) {
    res
      .status(500)
      .json({ message: "Error creating player", error: error.message });
  }
});
```

Business Logic : Get All Team Player

```
// Business Logic: get All team Players
app.get("/teams/:teamId/players", (req, res) => {
  try {
    Team.findById(req.params.teamId)
      .populate("players")
      .then((team) => {
        if (!team) {
          return res.status(404).json({ message: "Team not found" });
        }
        console.log("Team Players", team.players);
      });
  }
});
```

```

res.json({ teamPlayers: team.players });
});
} catch (error) {
res
.status(500)
.json({ message: "Error retrieving players", error: error.message });
}
});

```

Component: AddPlayerComponent (TS)

```

export class AddPlayerComponent implements OnInit {

  // Form ID
  playerForm: FormGroup;
  // player Object
  player: any = {};
  teamList: any = [];
  teamId: any;
  constructor(private playerService: PlayerService, private teamService: TeamService)
  { }

  ngOnInit() {
    this.teamService.getAllTeams().subscribe(
      (teams) => {
        this.teamList = teams.teamsTab;
      }
    )
  }

  addPlayer() {
    console.log("Here player object", this.player);
    this.player.teamId = this.teamId;
    this.playerService.addPlayer(this.player).subscribe(
      (response) => {
        console.log("Here response from BE", response);
      }
    );
  }

  getTeamId(evt) {
    this.teamId = evt.target.value;
    console.log("Team ID", this.teamId);
  }
}

```

Component: AddPlayerComponent (HTML)

```
<div class="hero overlay" style="background-image: url('assets/images/bg_3.jpg');">
<div class="container">
<div class="row align-items-center">
<div class="col-lg-9 mx-auto text-center">
<h1 class="text-white">Add Player</h1>
</div>
</div>
</div>
</div>
<div class="site-section">
<div class="container">
<div class="row">
<div class="col-lg-12">
<form #playerForm="ngForm" (ngSubmit)="addPlayer()">
<div class="form-group">
<input type="text" class="form-control" [(ngModel)]="player.name"
name="name" placeholder="Name"
>
</div>
<div class="form-group">
<input type="number" class="form-control" [(ngModel)]="player.nbr"
name="nbr" placeholder="Nbr">
</div>
<div class="form-group">
<input type="number" class="form-control" [(ngModel)]="player.age"
name="age" placeholder="Age">
</div>
<div class="form-group">
<select (change)="getTeamId($event)" name="tID" >
<option [value]="t._id" *ngFor="let t of teamList" >{{t.name}}</option>
</select>
</div>
<div class="form-group">
<input type="submit" class="btn btn-primary py-3 px-5" value="Add Player">
</div>
</form>
</div>
</div>
</div>
</div>
```