UFFS GamePlay API Usage

Introduction

The GamePlay API is a RESTful API created to serve GamePlay data for UFFS franchises, players, and sports. Front-end applications (such as the GamePlay website, NFT card game, etc) can use these endpoints to retrieve information. Admin users can use these endpoints to update certain Player/Franchise records as the leagues evolve (franchises are added and dropped, players are traded, etc).

Currently Player and Franchise data is sourced from the UFFS database APIs, but we intend to move the ownership of the data (including the creation of new players) into the GamePlay database eventually. The fantasy data points (player stats) are currently pulled from the sources that the leagues are in currently (e.g. Fantrax for Hockey/Baseball, MyFantasyLeague for American Football)

Each major sport (Hockey, Baseball, American Football, Basketball, etc) will have its own deployment, and thus its own "root" URL. But the endpoint names and return values will be as consistent as possible across all sports, keeping in mind that different sports leagues have different positions, statistics data, league structures, etc, that may affect the return values.

The GamePlay endpoints don't have any authentication, as the data is public and not sensitive.

Currently the sports that are developed and ready are (with their development root URLs):

Hockey: https://svc-dev-uff.scrypted.net/webgameplayapigw/hl/
Baseball: https://svc-dev-uff.scrypted.net/webgameplayapigw/lb/

Other sports that are currently in progress: American Football, Basketball

As we are still in the development process, please reach out to Scott Vig if you encounter any issues or would like to request a new endpoint to access data.

Endpoints

Using **Hockey's** root URL for examples as it is the most "complete" sport currently. Swap in another sport's root URL to get that sport's data instead.

[GET] GetLeagues

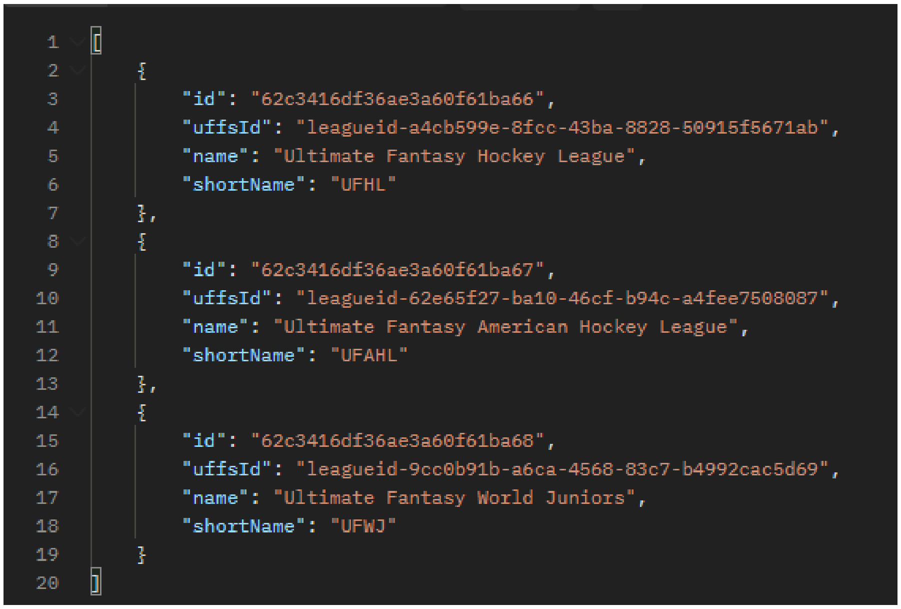
https://svc-dev-uff.scrypted.net/webgameplayapigw/hl/api/v1/GamePlay/GetLeagues

Returns the list of Leagues associated with the UFFS sport ecosystem

Parameters:

id (string): Specify the id of the League you want to return
uffsId (string) Specify the UFFS id of the League you want to return

Example return:



```
[
    {
        "id": "62c3416df36ae3a60f61ba66",
        "uffsId": "leagueid-a4cb599e-8fcc-43ba-8828-50915f5671ab",
        "name": "Ultimate Fantasy Hockey League",
        "shortName": "UFHL"
    },
    {
        "id": "62c3416df36ae3a60f61ba67",
        "uffsId": "leagueid-62e65f27-ba10-46cf-b94c-a4fee7508087",
        "name": "Ultimate Fantasy American Hockey League",
        "shortName": "UFAHL"
    },
    {
        "id": "62c3416df36ae3a60f61ba68",
        "uffsId": "leagueid-9cc0b91b-a6ca-4568-83c7-b4992cac5d69",
        "name": "Ultimate Fantasy World Juniors",
        "shortName": "UFWJ"
    }
]
```

[GET] GetSeasons

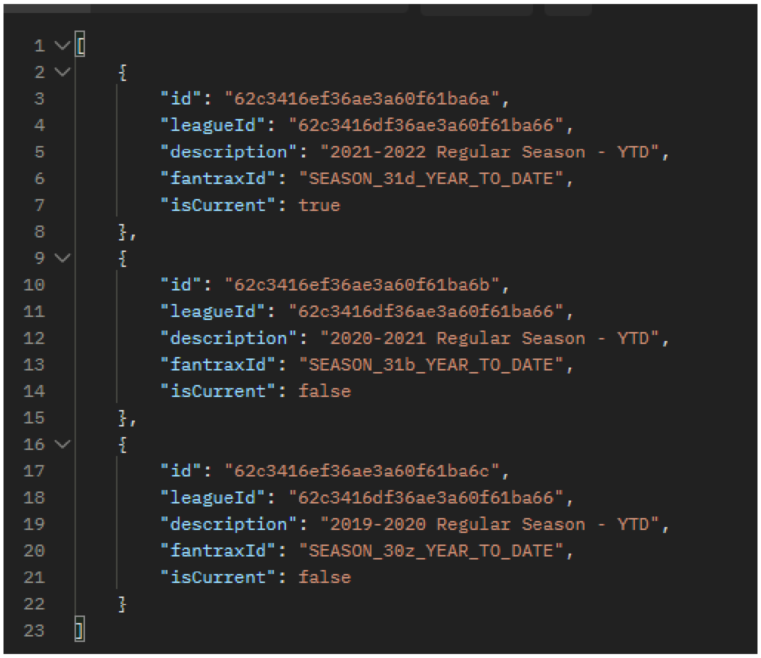
https://svc-dev-uff.scrypted.net/webgameplayapigw/hl/api/v1/GamePlay/GetSeasons

Returns the list of Seasons in the UFFS sport ecosystem

Parameters:

id (string): Specify the id of the Season you want to return
leagueId (string): Specify the id of the Leagues you want to pull Seasons for
getCurrent (bool): Return only Seasons that are “active” or “current”

Example return (no params):



```
[
    {
        "id": "62c3416ef36ae3a60f61ba6a",
        "leagueId": "62c3416df36ae3a60f61ba66",
        "description": "2021-2022 Regular Season - YTD",
        "fantraxId": "SEASON_31d_YEAR_TO_DATE",
        "isCurrent": true
    },
    {
        "id": "62c3416ef36ae3a60f61ba6b",
        "leagueId": "62c3416df36ae3a60f61ba66",
        "description": "2020-2021 Regular Season - YTD",
        "fantraxId": "SEASON_31b_YEAR_TO_DATE",
        "isCurrent": false
    },
    {
        "id": "62c3416ef36ae3a60f61ba6c",
        "leagueId": "62c3416df36ae3a60f61ba66",
        "description": "2019-2020 Regular Season - YTD",
        "fantraxId": "SEASON_30z_YEAR_TO_DATE",
        "isCurrent": false
    }
]
```

[GET] GetDivisions

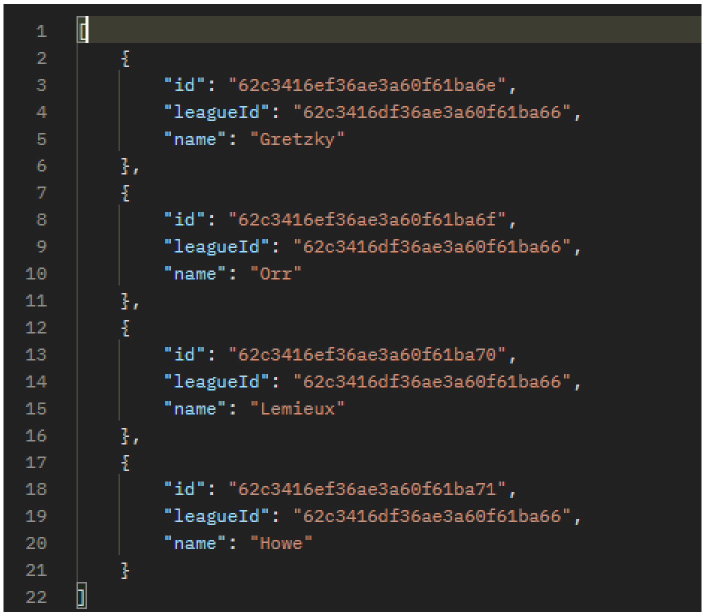
https://svc-dev-uff.scrypted.net/webgameplayapigw/hl/api/v1/GamePlay/GetDivisions

Returns the list of Divisions in the UFFS sport ecosystem

Parameters:

id (string): Specify the id of the Division you want to return
leagueId (string): Specify the id of the League you want to pull Divisions for

Example return (no params):



```json
[
    {
        "id": "62c3416ef36ae3a60f61ba6e",
        "leagueId": "62c3416df36ae3a60f61ba66",
        "name": "Gretzky"
    },
    {
        "id": "62c3416ef36ae3a60f61ba6f",
        "leagueId": "62c3416df36ae3a60f61ba66",
        "name": "Orr"
    },
    {
        "id": "62c3416ef36ae3a60f61ba70",
        "leagueId": "62c3416df36ae3a60f61ba66",
        "name": "Lemieux"
    },
    {
        "id": "62c3416ef36ae3a60f61ba71",
        "leagueId": "62c3416df36ae3a60f61ba66",
        "name": "Howe"
    }
]
```

[GET] GetFranchises

https://svc-dev-uff.scrypted.net/webgameplayapigw/hl/api/v1/GamePlay/GetFranchises

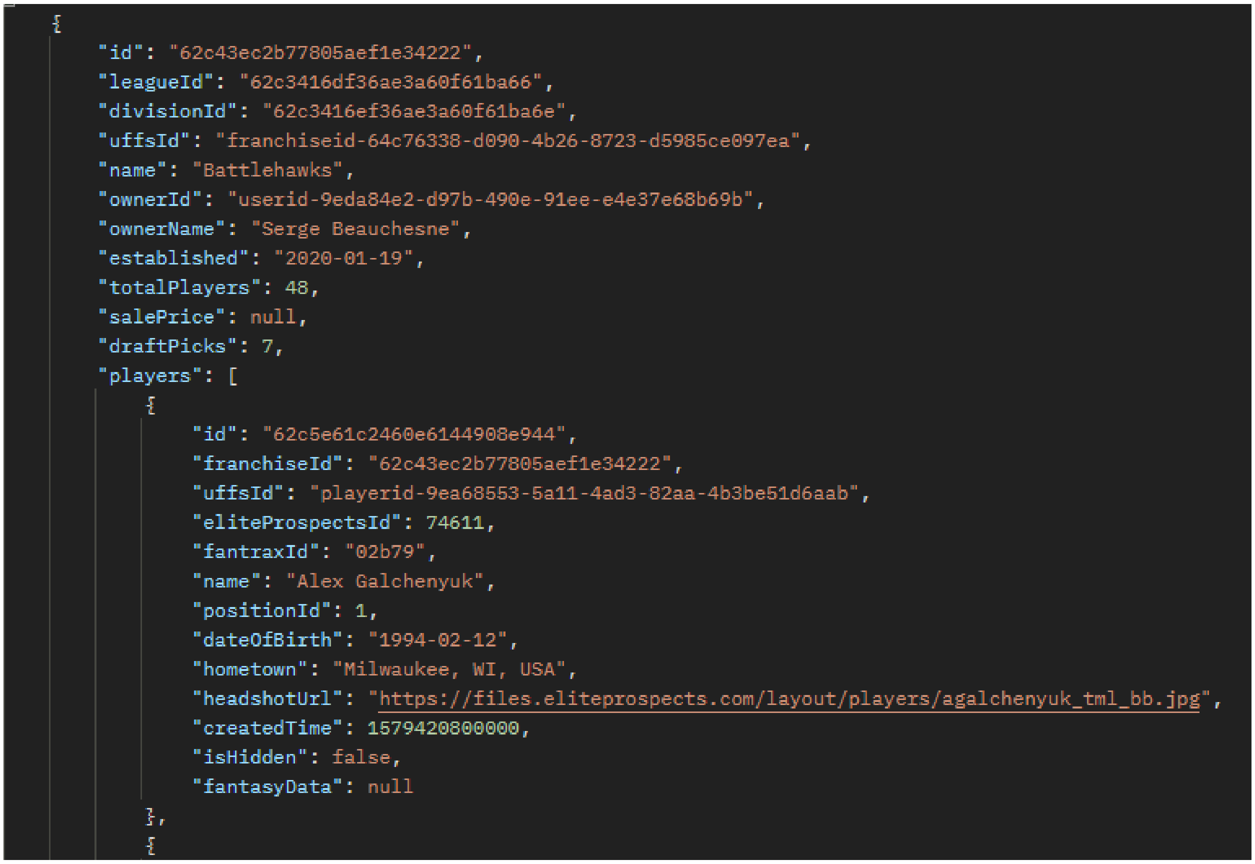
Returns the list of Franchises in the UFFS sport ecosystem

Parameters:

id (string): Specify the id of the Franchise you want to return
leagueId (string): Specify the id of the League you want to pull Franchises for
uffsId (string): Specify the uffs id of the Franchise you want to return
divisionId (string): Specify the id of the Divisions you want to pull Franchises for
includePlayers (bool): If true, return a list of Players on the Franchise roster
includeFantasyData (bool): If true, return the Fantasy Data associated with the Players
fantasyStatsType (int): Return the Fantasy Data for the provided fantasy type id (default of 1 uses the real life league stats)

Remarks: Fantrax has 3 types of fantasy data being pulled, so fantasyStatsType can be 1, 2, or 3. Likely you will be most concerned with the default type of 1.

Example return (uffsId = “franchiseid-64c76338-d090-4b26-8723-d5985ce097ea”, includePlayers = true):



```
{
    "id": "62c43ec2b77805aef1e34222",
    "leagueId": "62c3416df36ae3a60f61ba66",
    "divisionId": "62c3416ef36ae3a60f61ba6e",
    "uffsId": "franchiseid-64c76338-d090-4b26-8723-d5985ce097ea",
    "name": "Battlehawks",
    "ownerId": "userid-9eda84e2-d97b-490e-91ee-e4e37e68b69b",
    "ownerName": "Serge Beauchesne",
    "established": "2020-01-19",
    "totalPlayers": 48,
    "salePrice": null,
    "draftPicks": 7,
    "players": [
        {
            "id": "62c5e61c2460e6144908e944",
            "franchiseId": "62c43ec2b77805aef1e34222",
            "uffsId": "playerid-9ea68553-5a11-4ad3-82aa-4b3be51d6aab",
            "eliteProspectsId": 74611,
            "fantraxId": "02b79",
            "name": "Alex Galchenyuk",
            "positionId": 1,
            "dateOfBirth": "1994-02-12",
            "hometown": "Milwaukee, WI, USA",
            "headshotUrl": "https://files.eliteprospects.com/layout/players/agalchenyuk_tml_bb.jpg",
            "createdTime": 1579420800000,
            "isHidden": false,
            "fantasyData": null
        },
        {
```

[GET] GetPositions

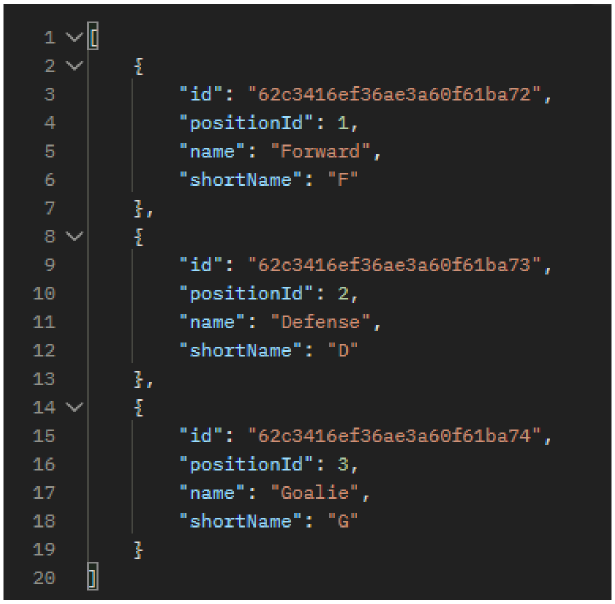
https://svc-dev-uff.scrypted.net/webgameplayapigw/hl/api/v1/GamePlay/GetPositions

Returns the list of Positions for this sport

Parameters:

Remarks: Position IDs are integer values that come from the existing UFFS APIs. Each sport will have a different list of IDs and positions.

Example return (no params):



```
[
    {
        "id": "62c3416ef36ae3a60f61ba72",
        "positionId": 1,
        "name": "Forward",
        "shortName": "F"
    },
    {
        "id": "62c3416ef36ae3a60f61ba73",
        "positionId": 2,
        "name": "Defense",
        "shortName": "D"
    },
    {
        "id": "62c3416ef36ae3a60f61ba74",
        "positionId": 3,
        "name": "Goalie",
        "shortName": "G"
    }
]
```

[GET] GetPlayers

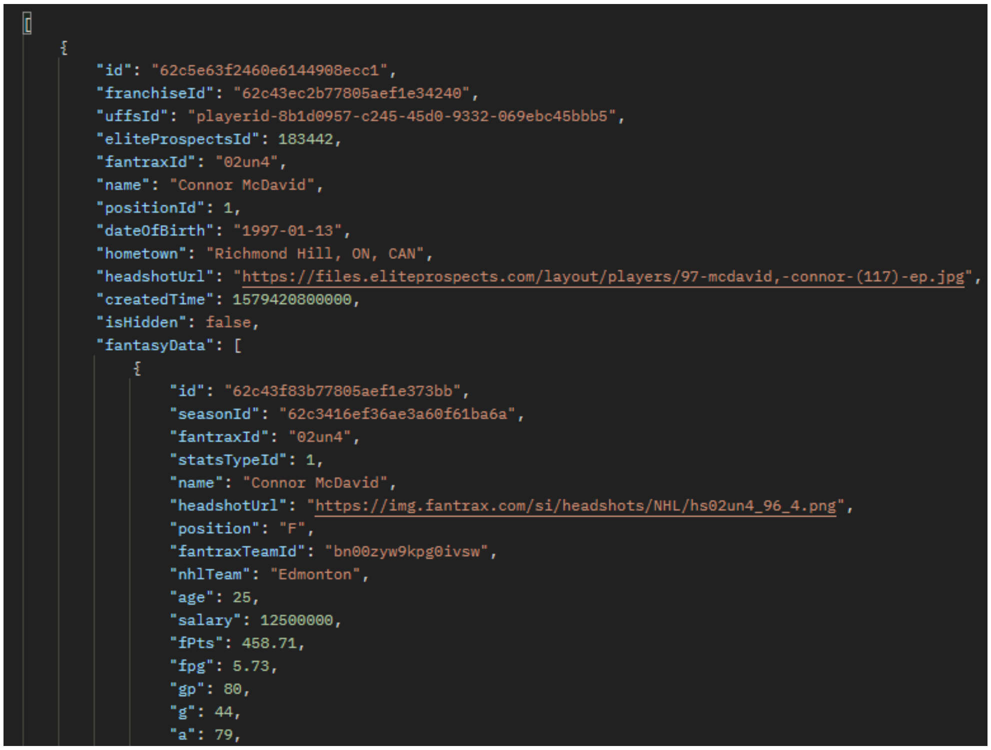
https://svc-dev-uff.scrypted.net/webgameplayapigw/hl/api/v1/GamePlay/GetPlayers

Returns the list of Players in the UFFS sport ecosystem

Parameters:

id (string): Specify the id of the Player you want to return
uffsId (string): Specify the uffs id of the Player you want to return
eliteProspectsId (string): Specify the Elite Prospects id of the player you want to return
fantraxId (string): Specify the Fantrax id of the Player you want to return
name (string): Perform a name search (string “contains”) for the Player(s)
includeFantasyData (bool): If true, return the Fantasy Data associated with the Player(s)
fantasyStatsType (int): Return the Fantasy Data for the provided fantasy type id (default of 1 uses the real life league stats)
includeDraftPicks (bool): If true, include draft picks in the return (default is false)
numRecords (int): Return only the first N number of Player(s) returned by the query

Example return (name = “connor mcdavid”, includeFantasyData = true):



```
[
    {
        "id": "62c5e63f2460e6144908ecc1",
        "franchiseId": "62c43ec2b77805aef1e34240",
        "uffsId": "playerid-8b1d0957-c245-45d0-9332-069ebc45bbb5",
        "eliteProspectsId": 183442,
        "fantraxId": "02un4",
        "name": "Connor McDavid",
        "positionId": 1,
        "dateOfBirth": "1997-01-13",
        "hometown": "Richmond Hill, ON, CAN",
        "headshotUrl": "https://files.eliteprospects.com/layout/players/97-mcdavid,-connor-(117)-ep.jpg",
        "createdTime": 1579420800000,
        "isHidden": false,
        "fantasyData": [
            {
                "id": "62c43f83b77805aef1e373bb",
                "seasonId": "62c3416ef36ae3a60f61ba6a",
                "fantraxId": "02un4",
                "statsTypeId": 1,
                "name": "Connor McDavid",
                "headshotUrl": "https://img.fantrax.com/si/headshots/NHL/hs02un4_96_4.png",
                "position": "F",
                "fantraxTeamId": "bn00zyw9kpg0ivsw",
                "nhlTeam": "Edmonton",
                "age": 25,
                "salary": 12500000,
                "fPts": 458.71,
                "fpg": 5.73,
                "gp": 80,
                "g": 44,
                "a": 79,
```

[GET] GetPlayersForScouting

https://svc-dev-uff.scrypted.net/webgameplayapigw/hl/api/v1/GamePlay/GetPlayersForScouting

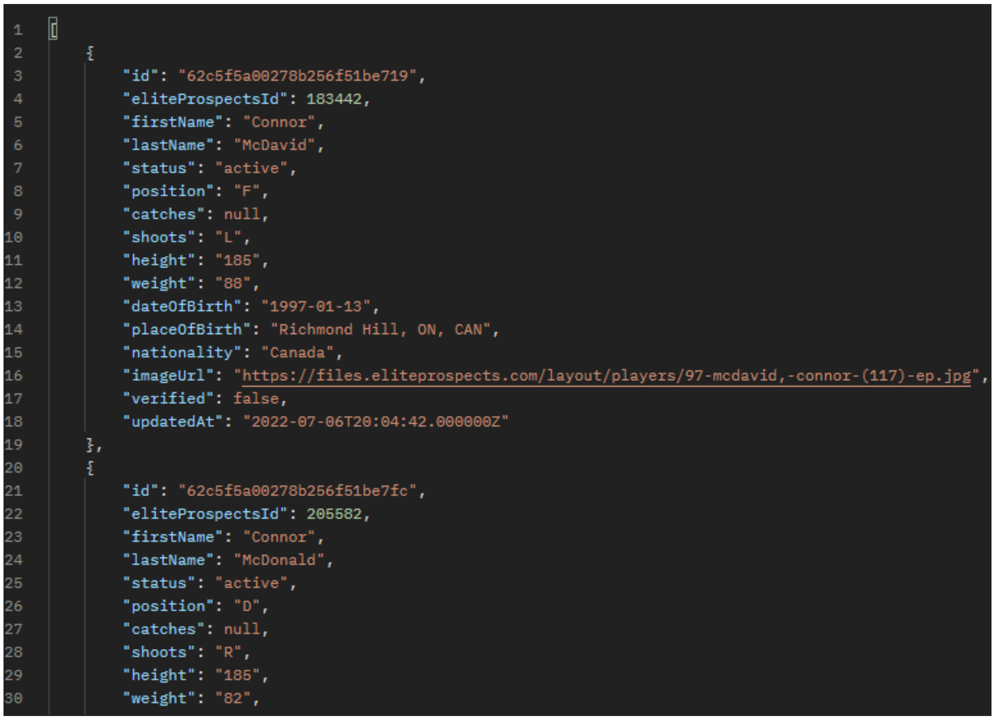
Returns the list of hockey players mined from EliteProspects (for Scouts, not already signed to a fantasy team)

Parameters:

eliteProspectsId (string): Specify the Elite Prospects id of the player you want to return
firstName (string): Perform a first name search (string “contains”) for the player(s)
lastName (string): Perform a last name search (string “contains”) for the player(s)
position (string): One of “F”, “D”, or “G”, or empty string for all
dateOfBirth (string): Format: “yyyy-MM-dd”
numRecords (int): Return only the first N number of player(s) returned by the query

Remarks: The dataset for Elite Prospects players is ~700k players. This endpoint is to scout for players before they become “signed” into the UFFS world.

Example return (firstName = “connor”, lastName = “mcd”)



```
[
    {
        "id": "62c5f5a00278b256f51be719",
        "eliteProspectsId": 183442,
        "firstName": "Connor",
        "lastName": "McDavid",
        "status": "active",
        "position": "F",
        "catches": null,
        "shoots": "L",
        "height": "185",
        "weight": "88",
        "dateOfBirth": "1997-01-13",
        "placeOfBirth": "Richmond Hill, ON, CAN",
        "nationality": "Canada",
        "imageUrl": "https://files.eliteprospects.com/layout/players/97-mcdavid,-connor-(117)-ep.jpg",
        "verified": false,
        "updatedAt": "2022-07-06T20:04:42.000000Z"
    },
    {
        "id": "62c5f5a00278b256f51be7fc",
        "eliteProspectsId": 205582,
        "firstName": "Connor",
        "lastName": "McDonald",
        "status": "active",
        "position": "D",
        "catches": null,
        "shoots": "R",
        "height": "185",
        "weight": "82",
```