**API SPECIFICATION**

Contents

[Request HEADERS: 3](#__RefHeading__7406_2026064013)

[A. User 4](#__RefHeading__2647_384732017)

[1. Register user 4](#__RefHeading__2649_384732017)

[7](#__RefHeading__2651_384732017)

[2. Login user 7](#__RefHeading__2653_384732017)

[3. Change/update user password 8](#__RefHeading__2655_384732017)

[9](#__RefHeading__2657_384732017)

[4. Update/change user record 10](#__RefHeading__2659_384732017)

[5. Log-out user 11](#__RefHeading__2661_384732017)

[B. Team 12](#__RefHeading__2663_384732017)

[1. Create team 12](#__RefHeading__2665_384732017)

[2. Get team record 14](#__RefHeading__2667_384732017)

[3. Update/change team record 15](#__RefHeading__2669_384732017)

[C. Match 17](#__RefHeading__2671_384732017)

[1. Create match 17](#__RefHeading__2673_384732017)

[2. Get login user coming matches records 19](#__RefHeading__2675_384732017)

[3. Get login user match history 20](#__RefHeading__2677_384732017)

[4. Get login user available matches 21](#__RefHeading__2679_384732017)

[5. Cancel the match 22](#__RefHeading__2681_384732017)

[D. JoinMatch 26](#__RefHeading__2683_384732017)

[1. Create Joinmatch 26](#__RefHeading__2685_384732017)

[2. Confirm join\_match records by match creator(home\_user) 28](#__RefHeading__2687_384732017)

[3. Withdraw join\_match records by joiner 29](#__RefHeading__2689_384732017)

[31](#__RefHeading__7408_2026064013)

[E. Joiners list 31](#__RefHeading__2691_384732017)

[1. Get Joiners list 31](#__RefHeading__2693_384732017)

[F. Pitch 33](#__RefHeading__2695_384732017)

[1. Get pitches list 33](#__RefHeading__2697_384732017)

[G. Advertisement 35](#__RefHeading__2699_384732017)

[1. Get advertisements list 35](#__RefHeading__2701_384732017)

[35](#__RefHeading__2703_384732017)

# Request HEADERS:

**#default headers for all the requests**

Content-Type: application/json

**Request Header parameters::**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | access-token | Y |  | Example: WWjsy0Rb7zYl0mC73XrbLw |
| 2 | client | Y |  | Example: 3q8\_KQ\_fIpwtjJS4hgGDRA |
| 3 | expiry | Y |  | Example: 1507792710 |
| 4 | uid | Y |  | Example: testuser@yopmail.com |

**Note:** Request Header parameters would require for all the requests except user registration/signin apis, to get the “access-token”, “client”, “expiry” and “uid” mobile-app have to call the user-signin api

# A. User

## 1. Register user

**1.1 The url to register user will be POST type:-**

<http://localhost:3000/api/auth>

**1.2 Header Parameters**

While registering user required to pass **Content-Type** inside request header. This Content-Type will be used to identify request type. This is a mandatory parameter for calling user registration api.

Content-Type (will be json i.e. application/json)

**1.3 Body Parameters**

All parameters list accepted by User api

**User Registration Request parameters::**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | email | Y | Uniq |  |
| 2 | password | Y | Minimum length 6 |  |
| 3 | password\_confirmation | Y | Minimum length 6 |  |
| 4 | username | Y | Uniq |  |
| 5 | register\_type | Y | N/A | “google”/”facebook”/”mobile” |
| 6 | mobile\_no | Y | N/A |  |
| 7 | status | N | N/A | “a”/”o”  default: “a” |
| 8 | avatar | N | N/A | base64 encoded image |
| 9 | remark | N | N/A |  |

**User Registration Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | Uniq |  |
| 2 | email | Y | Uniq |  |
| 3 | provider | Y | N/A |  |
| 4 | username | Y | Uniq |  |
| 5 | register\_type | Y | N/A |  |
| 6 | mobile\_no | Y | N/A |  |
| 7 | remark | Y | N/A |  |
| 8 | status | Y | N/A |  |
| 9 | avatar | Y | N/A |  |
| 10 | uid | Y | N/A |  |
| 11 | login\_fail | Y | N/A |  |
| 12 | deleted\_at | Y | N/A |  |
| 13 | created\_at | Y | N/A |  |
| 14 | updated\_at | Y | N/A |  |

**1.4 User registration request/response Json sample**

User registration request json sample

{ "user": {"email": "testuser@yopmail.com", "password": "123456789", "password\_confirmation": "123456789", "username": "test\_user", "register\_type": "mobile", "mobile\_no": "9111110000"}}

[\* all fields are compulsory]

After submitting, response on 'failure' will be :-

{"status": "error", "data": {"id": null, "provider": "email", "uid": "", "email": "[testuser@yopmail.com](mailto:testuser2@yopmail.com)", "username": "test\_user", "mobile\_no": null, "register\_type": null, "login\_fail": 0, "remark": null, "deleted\_at": null, "created\_at": null, "updated\_at": null}, "errors": {"password": ["is too short (minimum is 6 characters)"], "mobile\_no": ["can't be blank"], "register\_type": ["can't be blank"], "full\_messages": ["Password is too short (minimum is 6 characters)", "Mobile no can't be blank", "Register type can't be blank"] }}

After submitting, response on 'success' will be :-

{"status": "success","data": {"id": 1,"email": "testuser@yopmail.com","provider": "email","avatar": {"url": null},"username": "test\_user","mobile\_no": "9111110000","register\_type": "mobile","uid": "testuser@yopmail.com","login\_fail": 0,"remark": null,"status": "a","deleted\_at": null,"created\_at": "2017-09-28T07:14:05.000Z","updated\_at": "2017-09-28T07:14:06.000Z"}}

## 

## 2. Login user

**2.1 The url for login user will be POST type:-**

<http://localhost:3000/api/auth/sign_in>

### 

**2.2 Header Parameters**

While user login required to pass **Content-Type** inside request header. This Content-Type will be used to identify request type. This is a mandatory parameter for calling user login api.

Content-Type (will be json i.e. application/json)

### 

**2.3 Body Parameters**

All parameters list accepted by User api

**User Login Request parameters::**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | email | Y | Uniq |  |
| 2 | password | Y | Minimum length 6 |  |

**User Login Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | Uniq |  |
| 2 | email | Y | Uniq |  |
| 3 | provider | Y | N/A |  |
| 4 | username | Y | Uniq |  |
| 5 | register\_type | Y | N/A |  |
| 6 | mobile\_no | Y | N/A |  |
| 7 | remark | Y | N/A |  |
| 8 | status | Y | N/A |  |
| 9 | avatar | Y | N/A |  |
| 10 | uid | Y | N/A |  |
| 11 | login\_fail | Y | N/A |  |
| 12 | deleted\_at | Y | N/A |  |

**2.4 User sign-in request/response Json sample**

User sign-in request json sample

{"user": { "email": "testuser@yopmail.com", "password": "123456789" }}

After submitting, response on 'failure' will be :-

{"errors": ["Invalid login credentials. Please try again."]}

After submitting, response on 'success' will be :-

{"data": {"id": 1,"email": "testuser@yopmail.com","provider": "email","avatar": {"url": null},"username": "test\_user","mobile\_no": "9111110000","register\_type": "mobile","uid": "testuser@yopmail.com","login\_fail": 0,"remark": null,"status": "a","deleted\_at": null}}

## 3. Change/update user password

**3.1 The url for change/update user password will be PUT type:-**

<http://localhost:3000/api/auth/password>

**3.2 Header Parameters**

While updating user password required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling user update password api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**3.3 Body Parameters**

All parameters list accepted by User api

**User change/update password parameters::**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | password | Y | Minimum length 6 |  |
| 2 | password\_confirmation | Y | Minimum length 6 |  |

### 

**User change/update password Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | Uniq |  |
| 2 | email | Y | Uniq |  |
| 3 | provider | Y | N/A |  |
| 4 | username | Y | Uniq |  |
| 5 | register\_type | Y | N/A |  |
| 6 | mobile\_no | Y | N/A |  |
| 7 | remark | Y | N/A |  |
| 8 | status | Y | N/A |  |
| 9 | avatar | Y | N/A |  |
| 10 | uid | Y | N/A |  |
| 11 | login\_fail | Y | N/A |  |
| 12 | deleted\_at | Y | N/A |  |
| 13 | created\_at | Y | N/A |  |
| 14 | updated\_at | Y | N/A |  |

**3.4 Change/Update user password request/response Json sample**

Change/Update user request json sample

{"user":{"password": "12345678", "password\_confirmation": "12345678"}}

After submitting, response on 'failure' will be :-

{"success": false, "errors": {"password\_confirmation": ["doesn't match Password"], "password": ["is too short (minimum is 6 characters)"], "full\_messages": ["Password confirmation doesn't match Password", "Password is too short (minimum is 6 characters)"]}}

After submitting, response on 'success' will be :-

{"success": true,"data": {"provider": "email","id": 1,"email": "testuser@yopmail.com","avatar": {"url": null},"username": "test\_user","mobile\_no": "9111110000","register\_type": "mobile","uid": "testuser@yopmail.com","login\_fail": 0,"remark": null,"status": "a","deleted\_at": null,"created\_at": "2017-09-28T07:14:05.000Z","updated\_at": "2017-09-28T07:43:36.000Z"},"message": "Your password has been successfully updated."}

## 4. Update/change user record

**4.1 The url for change/update user's record will be PUT type:-**

<http://localhost:3000/api/auth>

**4.2 Header Parameters**

While updating user record required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling user update api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**4.3 Body Parameters**

All parameters list accepted by User api

**User change/update user record parameters::**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | email | N | Uniq |  |
| 2 | username | N | Uniq |  |
| 3 | avatar | N | N/A | base64 encoded image |
| 4 | mobile\_no | N | N/A |  |
| 5 | register\_type | N | N/A | “google”/”facebook”/”mobile” |
| 6 | remark | N | N/A |  |
| 7 | status | N | N/A | “a”/”o” |

**User change/update user record Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | Uniq |  |
| 2 | email | Y | Uniq |  |
| 3 | provider | Y | N/A |  |
| 4 | username | Y | Uniq |  |
| 5 | register\_type | Y | N/A |  |
| 6 | mobile\_no | Y | N/A |  |
| 7 | remark | Y | N/A |  |
| 8 | status | Y | N/A |  |
| 9 | avatar | Y | N/A |  |
| 10 | uid | Y | N/A |  |
| 11 | login\_fail | Y | N/A |  |
| 12 | deleted\_at | Y | N/A |  |
| 13 | created\_at | Y | N/A |  |
| 14 | updated\_at | Y | N/A |  |

**4.4 Change/Update user record request/response Json sample**

Change/Update user record request json sample

{"user":{"username": "test\_user12"}}

After submitting, response on 'failure' will be :-

{"status": "error", "errors": {"username": ["has already been taken"], "full\_messages": ["Username has already been taken"]}}

After submitting, response on 'success' will be :-

{"status": "success","data": {"id": 1,"username": "test\_user","email": "testuser@yopmail.com","provider": "email","avatar": {"url": null},"mobile\_no": "9111110000","register\_type": "mobile","uid": "testuser@yopmail.com","login\_fail": 0,"remark": null,"status": "a","deleted\_at": null,"created\_at": "2017-09-28T07:14:05.000Z","updated\_at": "2017-09-28T07:43:36.000Z"}}

## 5. Log-out user

**5.1 The url for sign-out user will be DELETE type:-**

<http://localhost:3000/api/auth/sign_out>

**5.2 Header Parameters**

While destroying user session required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling user sign-out api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**5.3 User log-out request/response Json sample**

**User log-out request json sample**

in this request only header part will be submit

**After submitting, response on 'failure' will be :-**

{ "success": false, "message": "Invalid token." }

**After submitting, response on 'success' will be :-**

{ "success": true }

**Note:**

**1.** In the HEADER of user api response, we get 'access-token', 'client', 'expiry' and 'uid', which will be cumpulsory to pass in request header to access url except(user registration and user sign-in request headers), for which authentication is required.

**2.** Request/Response paramsters formate always in json.

# B. Team

## 1. Create team

**1.1 The url to create team will be POST type:-**

<http://localhost:3000/api/teams>

**1.2 Header Parameters**

While creating user teams required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling create team api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**1.3 Body Parameters**

All parameters list accepted by Team api

**Creating Team Request parameters::**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | team\_name | Y | N/A |  |
| 2 | prefer\_district\_id | N | N/A |  |
| 3 | prefer\_match\_weekday | N | N/A |  |
| 4 | prefer\_match\_starttime | N | N/A |  |
| 5 | prefer\_match\_endtime | N | N/A |  |
| 6 | size | Y | N/A |  |
| 7 | color1 | Y | N/A |  |
| 8 | color2 | Y | N/A |  |
| 9 | remark | N | N/A |  |
| 10 | status | Y | N/A | “a”/”o” |

**Creating Team Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | N/A |  |
| 2 | team\_name | Y | N/A |  |
| 3 | prefer\_match\_weekday | N | N/A |  |
| 4 | prefer\_match\_starttime | N | N/A |  |
| 5 | prefer\_match\_endtime | N | N/A |  |
| 6 | size | Y | N/A |  |
| 7 | color1 | Y | N/A |  |
| 8 | color2 | Y | N/A |  |
| 9 | remark | N | N/A |  |
| 10 | status | Y | N/A | “a”/”o” |
| 11 | district(district) | N | N/A |  |
| 12 | district(region) | N | N/A |  |
| 13 | captain(id) | Y | N/A |  |
| 14 | captain(username) | Y | N/A |  |

**1.4 Team creating request/response Json sample**

User teams creating request json sample

{ "team": {"team\_name": "Detroit Lions", "prefer\_match\_weekday": "2", "prefer\_match\_starttime": "12:00:00", "prefer\_match\_endtime": "01:00:00","size": "5","color1": "ffc300","color2": "ffffff","status": "a"} }

After submitting, response on 'failure' will be :-

{"errors": ["Size can't be blank", "Color1 can't be blank"]}

After submitting, response on 'success' will be :-

{"team": {"id": 1,"team\_name": "Detroit Lions","size": 5,"color1": "ffc300","color2": "ffffff","remark": null,"status": "a","prefer\_match\_weekday": 2, "district": null, "captain": {"id": 1,"username": "test\_user"}, "prefer\_match\_starttime": "12:00:00","prefer\_match\_endtime": "01:00:00"}}

## 2. Get team record

**2.1 The url to get team record will be GET type:-**

[http://localhost:3000/api/teams/](http://localhost:3000/api/teams/9)1

**2.2 Header Parameters**

While creating user teams required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling create team api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**2.3 Get Team Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | N/A |  |
| 2 | team\_name | Y | N/A |  |
| 3 | prefer\_match\_weekday | N | N/A |  |
| 4 | prefer\_match\_starttime | N | N/A |  |
| 5 | prefer\_match\_endtime | N | N/A |  |
| 6 | size | Y | N/A |  |
| 7 | color1 | Y | N/A |  |
| 8 | color2 | Y | N/A |  |
| 9 | remark | N | N/A |  |
| 10 | status | Y | N/A | “a”/”o” |
| 11 | district(district) | N | N/A |  |
| 12 | district(region) | N | N/A |  |
| 13 | captain(id) | Y | N/A |  |
| 14 | captain(username) | Y | N/A |  |

**2.4 Get Team request/response Json sample**

Get Team request json sample

in this request only header part will be submit

After submitting, response on 'failure' will be :-

{"errors": "Record not found"}

After submitting, response on 'success' will be :-

{"team": {"id": 1,"team\_name": "Detroit Lions","size": 5,"color1": "ffc300","color2": "ffffff","remark": null,"status": "a","prefer\_match\_weekday": 2,"district": null, "captain": {"id": 1,"username": "test\_user"}, "prefer\_match\_starttime": "12:00:00","prefer\_match\_endtime": "01:00:00"}}

## 3. Update/change team record

**3.1 The url for change/update user's team record will be PUT type:-**

[http://localhost:3000/api/teams/](http://localhost:3000/api/teams/9)1

**3.2 Header Parameters**

While updating user team record required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling update team api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**3.3 Body Parameters**

All parameters list accepted by team api

**User change/update team record parameters::**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | team\_name | Y | N/A |  |
| 2 | prefer\_district\_id | N | N/A |  |
| 3 | prefer\_match\_weekday | N | N/A |  |
| 4 | prefer\_match\_starttime | N | N/A |  |
| 5 | prefer\_match\_endtime | N | N/A |  |
| 6 | size | Y | N/A |  |
| 7 | color1 | Y | N/A |  |
| 8 | color2 | Y | N/A |  |
| 9 | remark | N | N/A |  |
| 10 | status | Y | N/A | “a”/”o” |

**User change/update team Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | N/A |  |
| 2 | team\_name | Y | N/A |  |
| 3 | prefer\_match\_weekday | N | N/A |  |
| 4 | prefer\_match\_starttime | N | N/A |  |
| 5 | prefer\_match\_endtime | N | N/A |  |
| 6 | Size | Y | N/A |  |
| 7 | color1 | Y | N/A |  |
| 8 | color2 | Y | N/A |  |
| 9 | Remark | N | N/A |  |
| 10 | Status | Y | N/A | “a”/”o” |
| 11 | district(district) | N | N/A |  |
| 12 | district(region) | N | N/A |  |
| 13 | captain(id) | Y | N/A |  |
| 14 | captain(username) | Y | N/A |  |

**3.4 User team record change/update request/response Json sample**

Team record change/update request json sample

{"team": {"team\_name": "Detroit", "prefer\_match\_weekday": "4", "size": "5", "color1": "ffc300"}}

After submitting, response on 'failure' will be :-

{"errors": ["Size can't be blank"]}

After submitting, response on 'success' will be :-

{"team": {"id": 1,"team\_name": "Detroit","prefer\_match\_weekday": 4,"size": 5,"color1": "ffc300","color2": "ffffff","status": "a","remark": null,"district": null, "captain": {"id": 1,"username": "test\_user"}, "prefer\_match\_starttime": "12:00:00","prefer\_match\_endtime": "01:00:00"}}

# C. Match

## 1. Create match

**1.1 The url to create match will be POST type:-**

<http://localhost:3000/api/matches>

**1.2 Header Parameters**

While creating user matches required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling create match api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**1.3 Body Parameters**

All parameters list accepted by Match api

**Creating Match Request parameters::**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | home\_team\_id | Y | N/A |  |
| 2 | home\_team\_color | Y | N/A |  |
| 3 | size | Y | N/A |  |
| 4 | play\_start | Y | N/A |  |
| 5 | play\_end | Y | N/A |  |
| 6 | confirm\_end | Y | N/A |  |
| 7 | pitch\_id | N | N/A | reference to Pitch |
| 8 | other\_pitch | N | N/A | if pitch\_id is NULL, other\_pitch is required to fill |

**Creating Match Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | Id | Y | Uniq |  |
| 2 | home\_team\_id | Y | N/A |  |
| 3 | home\_team\_color | Y | N/A |  |
| 4 | size | Y | N/A |  |
| 5 | play\_start | Y | N/A |  |
| 6 | play\_end | Y | N/A |  |
| 7 | confirm\_end | Y | N/A |  |
| 8 | pitch\_id | N | N/A | reference to Pitch |
| 9 | other\_pitch | N | N/A | if pitch\_id is NULL, other\_pitch is required to fill |
| 10 | deleted\_at | N | N/A |  |
| 11 | created\_at | N | Y |  |
| 12 | updated\_at | N | N/A |  |
| 13 | cancel\_at\_by\_home | N | N/A |  |
| 14 | cancel\_at\_by\_system | N | N/A |  |
| 15 | Status | Y | N/A |  |

**1.4 User match creation request/response Json sample**

User match creation request json sample

{"match": {"home\_team\_id": "9", "home\_team\_color": "ffffff", "size": "5","play\_start": "2017-10-14 12:00:00","play\_end": "2017-10-14 01:00:00","confirm\_end": "2017-10-14 11:30:00", "other\_pitch": "optional"}}

After submitting, response on 'failure' will be :-

{"errors": ["Home team color can't be blank", "Other pitch can't be blank"]}

After submitting, response on 'success' will be :-

{"match": {"id": 1, "size": 5,"play\_start": "2017-10-14T12:00:00.000Z","play\_end": "2017-10-14T01:00:00.000Z","confirm\_end": "2017-10-14T11:30:00.000Z","pitch\_id": null,"other\_pitch": "optional","remark": null,"deleted\_at": null,"home\_team\_id": 1,"created\_at": "2017-09-28T09:50:30.000Z","updated\_at": "2017-09-28T09:50:30.000Z","home\_team\_color": "ffffff","cancel\_at\_by\_home": null,"cancel\_at\_by\_system": null,"status": "w"}}

## 2. Get login user coming matches records

**2.1 The url to get coming\_matches record will be GET type:-**

<http://localhost:3000/api/matches/coming>

**2.2 Header Parameters**

While retrieving user coming matches required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling get coming match api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**2.3 Login user coming matches Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | N/A |  |
| 2 | size | Y | N/A |  |
| 3 | play\_start | Y | N/A |  |
| 4 | play\_end | Y | N/A |  |
| 5 | home\_team(team\_name) | Y | N/A |  |

**2.4 Get user coming matches request/response Json sample**

Get login user coming matches request json sample

in this request only header part will be submit

After submitting, response on 'no records present' will be :-

{"matches": "No records found."}

After submitting, response on 'success' will be :-

{"matches": [{"id": 1,"size": 5,"play\_start": "2017-10-14T12:00:00.000Z","play\_end": "2017-10-14T01:00:00.000Z","home\_team": {"team\_name": "Detroit"}}]}

## 3. Get login user match history

**3.1 The url to get match\_history record will be GET type:-**

<http://localhost:3000/api/matches/history>

### 

**3.2 Header Parameters**

While retrieving user match history required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling get match history api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**3.3 Login user match history Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | N/A |  |
| 2 | size | Y | N/A |  |
| 3 | play\_start | Y | N/A |  |
| 4 | play\_end | Y | N/A |  |
| 5 | home\_team(team\_name) | Y | N/A |  |

**3.4 Get user match history request/response Json sample**

Get login user match history request json sample

in this request only header part will be submit

After submitting, response on 'no records present' will be :-

{"matches": "No records found."}

After submitting, response on 'success' will be :-

{"matches": [{"id": 1,"size": 5,"play\_start": "2017-09-28T10:18:11.000Z","play\_end": "2017-10-14T01:00:00.000Z","home\_team": {"team\_name": "Detroit"}}]}

## 4. Get login user available matches

**4.1 The url to get available matches record will be GET type:-**

<http://localhost:3000/api/matches/available>

**4.2 Header Parameters**

While retrieving user available matches required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling get available matches api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**4.3 Login user match history Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | N/A |  |
| 2 | size | Y | N/A |  |
| 3 | play\_start | Y | N/A |  |
| 4 | play\_end | Y | N/A |  |
| 5 | home\_team(team\_name) | Y | N/A |  |

**4.4 Get user available matches request/response Json sample**

Get login user match history request json sample

in this request only header part will be submit

After submitting, response on 'no records present' will be :-

{"matches": "No records found."}

After submitting, response on 'success' will be :-

{"matches": [{"id": 2,"size": 5,"play\_start": "2017-10-15T12:00:00.000Z","play\_end": "2017-10-15T01:00:00.000Z","home\_team": {"team\_name": "Lions"}}]}

## 5. Cancel the match

**5.1 The url to cancel the match record will be PUT type:-**

<http://localhost:3000/api/matches/5/cancel>

**5.2 Header Parameters**

While canceling user match required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling cancel particuler match api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**5.3 Login user match history Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | N/A |  |
| 3 | cancel\_at\_by\_home | Y | N/A |  |
| 4 | Status | Y | N/A | w: 'w' default: 'w'  m: 'm'  f: 'f'  c: 'c' |
| 5 | home\_team\_id | Y | N/A |  |
| 6 | home\_team\_color | Y | N/A |  |
| 7 | size | Y | N/A |  |
| 8 | play\_start | y | N/A |  |
| 9 | play\_end | Y | N/A |  |
| 10 | confirm\_end | Y | N/A |  |
| 11 | pitch\_id | Y | N/A |  |
| 12 | other\_pitch | Y | N/A |  |
| 13 | remark | Y | N/A |  |
| 14 | deleted\_at | Y | N/A |  |
| 15 | created\_at | Y | N/A |  |
| 16 | updated\_at | Y | N/A |  |
| 17 | cancel\_at\_by\_system | Y | N/A |  |

**5.4 Cancel user match request/response Json sample**

Cancel user match request json sample

in this request only header part will be submit

After submitting, response if 'record not present' will be :-

{"errors": "Record not found"}

After submitting, response if 'record already cancled' will be :-

{"error": "You already have cancelled this match"}

After submitting, response on 'success' will be :-

{"match": {"cancel\_at\_by\_home": "2017-09-28T10:27:44.000Z","id": 1,"status": "c","home\_team\_id": 1,"home\_team\_color": "ffffff","size": 5,"play\_start": "2017-09-28T10:18:11.000Z","play\_end": "2017-10-14T01:00:00.000Z","confirm\_end": "2017-10-14T11:30:00.000Z","pitch\_id": null,"other\_pitch": "optional","remark": null,"deleted\_at": null,"created\_at": "2017-09-28T09:50:30.000Z","updated\_at": "2017-09-28T10:27:44.000Z","cancel\_at\_by\_system": null}}

# D. JoinMatch

## 1. Create Joinmatch

**1.1 The url to create JoinMatch will be POST type:-**

<http://localhost:3000/api/join_matches>

**1.2 Header Parameters**

While creating user join\_matches required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling join match api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**1.3 Body Parameters**

All parameters list accepted by JoinMatch api

**Creating Join Match Request parameters::**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | match\_id | Y | N/A |  |
| 2 | join\_team\_id | Y | N/A |  |
| 3 | join\_team\_color | Y | N/A |  |

**Creating Join Match Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | Id | Y | Uniq |  |
| 2 | match\_id | Y | N/A |  |
| 3 | join\_team\_id | Y | N/A |  |
| 4 | cancel\_at\_by\_join | Y | N/A |  |
| 5 | status | Y | N/A | confirmation\_pending: 'confirmation\_pending' confirmed: 'confirmed' withdraw: 'withdraw' Default: 'confirmation\_pending' |
| 6 | join\_team\_color | Y | N/A |  |
| 7 | deleted\_at | Y | N/A |  |
| 8 | created\_at | Y | N/A |  |
| 9 | updated\_at | Y | N/A |  |

**1.4 User join\_matches creation request/response Json sample**

User join\_matches creation request json sample

{"join\_match": {"match\_id": 1,"join\_team\_id": 2, "join\_team\_color": "ffc300"}}

After submitting, response on 'failure' will be :-

{"errors": ["Join team color can't be blank"]}

After submitting, response on 'success' will be :-

{"join\_match": {"id": 1,"match\_id": 1,"join\_team\_id": 2,"cancel\_at\_by\_join": null,"deleted\_at": null,"created\_at": "2017-09-28T10:43:56.000Z","updated\_at": "2017-09-28T10:43:56.000Z","status": "confirmation\_pending","join\_team\_color": "ffc300"}}

## 2. Confirm join\_match records by match creator(home\_user)

**2.1 The url to confirm join\_match record will be PUT type:-**

[http:// localhost:3000/api/join\_matches/4/confirm](http://localhost:3000/api/join_matches/2/confirm)

**2.2 Header Parameters**

While confirming join\_matches record required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling to confirm join\_matches api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**2.3 Login user confirm join\_match Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | Id | Y | Uniq |  |
| 2 | match\_id | Y | N/A |  |
| 3 | join\_team\_id | Y | N/A |  |
| 4 | cancel\_at\_by\_join | Y | N/A |  |
| 5 | status | Y | N/A | confirmation\_pending: 'confirmation\_pending' confirmed: 'confirmed' withdraw: 'withdraw' Default: 'confirmation\_pending' |
| 6 | join\_team\_color | Y | N/A |  |
| 7 | deleted\_at | Y | N/A |  |
| 8 | created\_at | Y | N/A |  |
| 9 | updated\_at | Y | N/A |  |

**2.4 Confirm join\_match request/response Json sample**

Confirm join\_match request json sample

in this request only header part will be submit

After submitting, response on 'failure' will be :-

1. {"error": "you are not authorized to perform this operation"}

2. {"error": "You already have confirmed this match"}

After submitting, response on 'success' will be :-

{"join\_match": {"status": "confirmed","match\_id": 1,"id": 1,"join\_team\_id": 2,"join\_team\_color": "ffc300","cancel\_at\_by\_join": null,"deleted\_at": null,"created\_at": "2017-09-28T10:43:56.000Z","updated\_at": "2017-09-28T11:05:29.000Z"}}

## 3. Withdraw join\_match records by joiner

**3.1 The url to withdraw join\_match record will be PUT type:-**

<http://localhost:3000/api/join_matches/2/withdraw>

**3.2 Header Parameters**

While withdrawing join\_match record required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling to withdraw join\_match api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**3.3 Login user confirm join\_match Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | Id | Y | Uniq |  |
| 2 | match\_id | Y | N/A |  |
| 3 | join\_team\_id | Y | N/A |  |
| 4 | cancel\_at\_by\_join | Y | N/A |  |
| 5 | status | Y | N/A | confirmation\_pending: 'confirmation\_pending' confirmed: 'confirmed' withdraw: 'withdraw' Default: 'confirmation\_pending' |
| 6 | join\_team\_color | Y | N/A |  |
| 7 | deleted\_at | Y | N/A |  |
| 8 | created\_at | Y | N/A |  |
| 9 | updated\_at | Y | N/A |  |

**3.3 Withdraw join\_match request/response Json sample**

Confirm join\_match request json sample

in this request only header part will be submit

After submitting, response on 'failure' will be :-

1. {"error": "you are not authorized to perform this operation"}

2. {"error": "You already have withdraw this match"}

After submitting, response on 'success' will be :-

{"join\_match": {"status": "withdraw","id": 1,"cancel\_at\_by\_join": "2017-09-28T11:13:01.000Z","match\_id": 1,"join\_team\_id": 2,"join\_team\_color": "ffc300","deleted\_at": null,"created\_at": "2017-09-28T10:43:56.000Z","updated\_at": "2017-09-28T11:13:01.000Z"}}

# 

# E. Joiners list

## 1. Get Joiners list

**1.1 The url to get joiners list will be GET type:-**

[http://localhost:3000/api/matches/1/joiners\_list](http://localhost:3000/api/matches/5/joiners_list)

**1.2 Header Parameters**

While retrieving joiners list required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling get joiners list api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**1.3 Login user get joiners list Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | Uniq |  |
| 2 | status | Y | N/A | confirmation\_pending: 'confirmation\_pending' confirmed: 'confirmed' withdraw: 'withdraw' Default: 'confirmation\_pending' |
| 3 | join\_team\_color | Y | N/A |  |
| 4 | user(username) | Y | N/A |  |
| 5 | user\_match(id) | Y | N/A |  |
| 6 | user\_match(play\_start) | Y | N/A |  |
| 7 | user\_match(play\_end) | Y | N/A |  |
| 8 | user\_match(confirm\_end) | Y | N/A |  |
| 9 | home\_team(team\_name) | Y | N/A |  |
| 10 | team(team\_name) | Y | N/A |  |

**1.3 Get joiners list request/response Json sample**

Get joiners list request json sample

in this request only header part will be submit

After submitting, response on 'failure' will be :-

{"errors": "Record not found"}

After submitting, response on 'success' will be :-

{"join\_matches": [{"id": 1,"status": "confirmation\_pending","join\_team\_color": "ffc300","user\_match": {"id": 1,"size": 5,"play\_start": "2017-10-14T12:00:00.000Z","play\_end": "2017-10-14T01:00:00.000Z","confirm\_end": "2017-10-14T11:30:00.000Z","home\_team\_color": "ffffff","home\_team": {"team\_name": "Detroit"}},"team": {"team\_name": "Lions"}}]}

# F. Pitch

## 1. Get pitches list

**1.1 The url to get pitches list will be GET type:-**

<http://localhost:3000/api/pitches>

**1.2 Header Parameters**

While retrieving pitches list required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling get pitches list api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**1.3 Login user get pitches list Response**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Field** | **Mandatory** | **Validation** | **Remark** |
| 1 | id | Y | Uniq |  |
| 2 | size | Y | N/A |  |
| 3 | image | Y | N/A |  |
| 4 | name | Y | N/A |  |
| 5 | location | Y | N/A |  |
| 6 | district(district) | Y | N/A |  |
| 7 | district(region) | Y | N/A |  |

**1.4 Get pitches list request/response Json sample**

Get pitches list request json sample

in this request only header part will be submit

After submitting, response on 'success' will be :-

{"pitches": [{"id": 1,"size": 5,"image": null,"name": "Hongkong stedium","location": "Hongkong","district": {"district": "Honkong","region": "Hongkong"}}]}

# G. Advertisement

## 1. Get advertisements list

**1.1 The url to get advertisements list will be GET type:-**

<http://localhost:3000/api/advertisements>

**1.2 Header Parameters**

While retrieving advertisements list required to pass **content-Type, 'access-token', 'client', 'expiry' and 'uid'** inside request header. These parameters will be used to identify request type and user. These are mandatory parameters for calling get advertisements list api.

1) Content-Type (will be json i.e. application/json)

2) uid (uid got after log-in in response)

3) access-token (access-token got after log-in in response)

4) client (client got after log-in in response)

5) expiry (expiry got after log-in in response)

**1.3 Get advertisements list request/response Json sample**

Get advertisements list request json sample

in this request only header part will be submit

After submitting, response on 'success' will be :-

1. if no records found in advertisement table

{ "advertisements": [] }