Lichess.org API reference (2.0.0)

Download OpenAPI specification: Download

Introduction

Welcome to the reference for the Lichess API! Lichess is free/libre, open-source chess server powered by volunteers and donations.

- Get help in the lichess developers discord channel, or in the #lichess IRC channel on freenode.
- Contribute to this documentation on Github.
- Check out lichess widgets to embed in your website.
- Download all lichess rated games

Endpoint

All requests go to `https://lichess.org`.

Clients

- Python general API
- Python general API - async
- Python Lichess Bot

- Python Board API for Certabo

Rate limiting

All requests are rate limited using various strategies, to ensure the API remains responsive for everyone. Only make one request at a time. If you receive an HTTP response with a 429 status, please wait a full minute before resuming API usage.

Authentication

OAuth2

Authorization Code Flow

The authorization code flow allows your users to **login with lichess**.

- NodeJS example
- Flask/Python example
- NodeJS Passport strategy to login with Lichess OAuth2
- Create a Lichess OAuth app
- About authorization code flow

Real life Oauth2 Authorization Code Flow examples

- PyChess (source code)
- Lichess4545 (source code)
- English Chess Federation
- Rotherham Online Chess

Personal API Token

Personal API tokens allow you to quickly interact with Lichess OAuth API.

- `curl https://lichess.org/api/account -H "Authorization: Bearer <token>"`
- NodeJS example
- Generate a personal API token

Security Scheme Type	OAuth2
authorizationCode OAuth Flow	**Authorization URL:** https://oauth.lichess.org/oauth/authorize **Token URL:** https://oauth.lichess.org/oauth **Scopes:** • `preference:read` - Read your preferences • `preference:write` - Write your preferences • `email:read` - Read your email address • `challenge:read` - Read incoming challenges • `challenge:write` - Create, accept, decline challenges • `study:read` - Read private studies and broadcasts • `study:write` - Create, update, delete studies and broadcasts • `tournament:write` - Create tournaments • `puzzle:read` - Read puzzle activity • `team:write` - Join, leave, and manage teams • `msg:write` - Send private messages to other players • `board:play` - Play with the Board API • `bot:play` - Play with the Bot API. Only for Bot accounts

SameOrigin

CORS is supported on almost all endpoints, except those tagged `SameOrigin`. If you need CORS on one of those, please make an issue explaining your use case to request it.

Security Scheme Type	none

None

No authorization required. For requests that don't need an authenticated user.

Security Scheme Type	none

Account

Read and write account informations and preferences.
https://lichess.org/account/preferences/game-display

Get my profile

Public informations about the logged in user.

AUTHORIZATIONS: OAuth2

Responses

> 200

GET /api/account

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "id": "georges",
    "username": "Georges",
    "online": true,
  - "perfs": {
      + "chess960": { … },
      + "atomic": { … },
      + "racingKings": { … },
      + "ultraBullet": { … },
      + "blitz": { … },
      + "kingOfTheHill": { … },
      + "bullet": { … },
      + "correspondence": { … },
      + "horde": { … },
      + "puzzle": { … },
      + "classical": { … },
      + "rapid": { … }
    },
    "createdAt": 1290415680000,
    "disabled": false,
    "engine": false,
  "booster": false,
- "profile": {
      "country": "EC",
      "location": "string",
      "bio": "Free bugs!",
      "firstName": "Thibault",
      "lastName": "Duplessis",
      "fideRating": 1500,
      "uscfRating": 1500,
      "ecfRating": 1500,
      "links": "github.com/ornicar\r\ntwitter.com/ornicar"
  },
  "seenAt": 1522636452014,
  "patron": true,
- "playTime": {
      "total": 3296897,
      "tv": 12134
  },
  "language": "en-GB",
  "title": "NM",
  "url": "https://lichess.org/@/georges",
  "playing": "https://lichess.org/yqfLYJ5E/black",
  "nbFollowing": 299,
  "nbFollowers": 2735,
  "completionRate": 97,
- "count": {
      "all": 9265,
      "rated": 7157,
      "ai": 531,
      "draw": 340,
      "drawH": 331,
      "loss": 4480,
      "lossH": 4207,
      "win": 4440,
      "winH": 4378,
      "bookmark": 71,
      "playing": 6,
      "import": 66,
      "me": 0
  },
        "streaming": false,
        "followable": true,
        "following": false,
        "blocking": false,
        "followsYou": false
    }
```

Get my email address

Read the email address of the logged in user.

AUTHORIZATIONS: OAuth2 (`email:read`)

Responses

> 200

GET /api/account/email

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "email": "abathur@mail.org"
}
```

Get my preferences

Read the preferences of the logged in user.

- https://lichess.org/account/preferences/game-display
- https://github.com/ornicar/lila/blob/master/modules/pref/src/main/Pref.scala

AUTHORIZATIONS: OAuth2 (`preference:read`)

Responses

> 200

GET `/api/account/preferences`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
  - "prefs": {
        "dark": true,
        "transp": false,
        "bgImg": "http://example.com",
        "is3d": false,
        "theme": "blue",
        "pieceSet": "cburnett",
        "theme3d": "Black-White-Aluminium",
        "pieceSet3d": "Basic",
        "soundSet": "silent",
        "blindfold": 0,
        "autoQueen": 2,
        "autoThreefold": 2,
        "takeback": 3,
        "moretime": 3,
        "clockTenths": 1,
        "clockBar": true,
        "clockSound": true,
        "premove": true,
        "animation": 2,
        "captured": true,
        "follow": true,
        "highlight": true,
        "destination": true,
        "coords": 2,
        "replay": 2,
        "challenge": 4,
        "message": 3,
        "coordColor": 2,
        "submitMove": 4,
        "confirmResign": 1,
        "insightShare": 1,
        "keyboardMove": 0,
        "zen": 0,
        "moveEvent": 2,
        "rookCastle": 1
    }
}
```

Get my kid mode status

Read the kid mode status of the logged in user.

- https://lichess.org/account/kid

AUTHORIZATIONS: OAuth2 (`preference:read`)

Responses

> 200

GET `/api/account/kid`

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "kid": false
}
```

Set my kid mode status

Set the kid mode status of the logged in user.

- https://lichess.org/account/kid

AUTHORIZATIONS: OAuth2 (`preference:write`)

QUERY PARAMETERS

`v` `required`	boolean Example: `v=true` Kid mode status

Responses

> 200

POST `/api/account/kid`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Users

Access registered users on Lichess. https://lichess.org/player

Get real-time users status

Read the `online`, `playing` and `streaming` flags of several users.

This API is very fast and cheap on lichess side. So you can call it quite often (like once every 5 seconds).

Use it to track players and know when they're connected on lichess and playing games.

AUTHORIZATIONS: None

QUERY PARAMETERS

`ids` `required`	string Example: `ids=aliquantus,chess-network,lovlas` User IDs separated by commas. Up to 50 IDs.

Responses

› 200

GET `/api/users/status`

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
[
  - {
        "id": "aliquantus",
        "name": "Aliquantus"
    },
  - {
        "id": "chess-network",
        "name": "Chess-Network",
        "title": "NM",
        "online": true,
        "playing": true,
        "streaming": true,
        "patron": true
    }
]
```

Get all top 10

Get the top 10 players for each speed and variant.

See https://lichess.org/player.

AUTHORIZATIONS: SameOrigin

HEADER PARAMETERS

`Accept` required	string Default: `application/vnd.lichess.v3+json`

Responses

> 200

GET /player

Response samples

200

Content type

application/vnd.lichess.v3+json

Copy Expand all Collapse all

```
{
  - "bullet": [
      + { … },
      + { … },
      + { … }
    ],
    "blitz": [ ],
    "rapid": [ ],
    "classical": [ ],
    "ultraBullet": [ ],
    "chess960": [ ],
    "crazyhouse": [ ],
    "antichess": [ ],
    "atomic": [ ],
    "horde": [ ],
    "kingOfTheHill": [ ],
    "racingKings": [ ],
    "threeCheck": [ ]
}
```

Get one leaderboard

Get the leaderboard for a single speed or variant (a.k.a. `perfType`). There is no leaderboard for correspondence or puzzles.

See https://lichess.org/player/top/200/bullet.

AUTHORIZATIONS: SameOrigin

PATH PARAMETERS

`nb` required	integer `[ 1 .. 200 ]` Example: `100` How many users to fetch
`perfType` required	string Enum: `"ultraBullet"` `"bullet"` `"blitz"` `"rapid"` `"classical"` `"chess960"` `"crazyhouse"` `"antichess"` `"atomic"` `"horde"` `"kingOfTheHill"` `"racingKings"` `"threeCheck"` Example: `bullet` The speed or variant

HEADER PARAMETERS

`Accept` required	string Default: `application/vnd.lichess.v3+json`

Responses

› 200

GET `/player/top/{nb}/{perfType}`

Response samples

200

Content type

application/vnd.lichess.v3+json

Copy Expand all Collapse all

```
{
  - "users": [
      + { … },
      + { … },
      + { … }
    ]
}
```

Get user public data

Read public data of a user.

AUTHORIZATIONS: None

PATH PARAMETERS

`username` required	string

Responses

> 200

GET /api/user/{username}

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "id": "georges",
    "username": "Georges",
    "online": true,
  - "perfs": {
      + "chess960": { … },
      + "atomic": { … },
      + "racingKings": { … },
      + "ultraBullet": { … },
      + "blitz": { … },
      + "kingOfTheHill": { … },
      + "bullet": { … },
      + "correspondence": { … },
      + "horde": { … },
      + "puzzle": { … },
      + "classical": { … },
      + "rapid": { … }
    },
    "createdAt": 1290415680000,
    "disabled": false,
    "engine": false,
    "booster": false,
 - "profile": {
       "country": "EC",
       "location": "string",
       "bio": "Free bugs!",
       "firstName": "Thibault",
       "lastName": "Duplessis",
       "fideRating": 1500,
       "uscfRating": 1500,
       "ecfRating": 1500,
       "links": "github.com/ornicar\r\ntwitter.com/ornicar"
   },
   "seenAt": 1522636452014,
   "patron": true,
 - "playTime": {
       "total": 3296897,
       "tv": 12134
   },
   "language": "en-GB",
   "title": "NM",
   "url": "https://lichess.org/@/georges",
   "playing": "https://lichess.org/yqfLYJ5E/black",
   "nbFollowing": 299,
   "nbFollowers": 2735,
   "completionRate": 97,
 - "count": {
       "all": 9265,
       "rated": 7157,
       "ai": 531,
       "draw": 340,
       "drawH": 331,
       "loss": 4480,
       "lossH": 4207,
       "win": 4440,
       "winH": 4378,
       "bookmark": 71,
       "playing": 6,
       "import": 66,
       "me": 0
   },
   "streaming": false,
        "followable": true,
        "following": false,
        "blocking": false,
        "followsYou": false
    }
```

Get rating history of a user

Read rating history of a user, for all perf types. There is at most one entry per day. Format of an entry is `[year, month, day, rating]`. `month` starts at zero (January).

AUTHORIZATIONS: None

PATH PARAMETERS

`username` required	string

Responses

> 200

GET `/api/user/{username}/rating-history`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
  - {
        "name": "Bullet",
      + "points": [ … ]
    },
  - {
        "name": "Blitz",
      + "points": [ … ]
    }
]
```

Get user activity

Read data to generate the activity feed of a user.

AUTHORIZATIONS: None

PATH PARAMETERS

username required	string

Responses

— 200

GET /api/user/{username}/activity

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
"https://gist.github.com/ornicar/0ee2d2427cb74ed1a35e86f5ba09fabc"
```

Get your puzzle activity

Download your puzzle activity in ndjson format.

Puzzle activity is sorted by reverse chronological order (most recent first)

We recommend streaming the response, for it can be very long.

AUTHORIZATIONS: OAuth2 (`puzzle:read`)

QUERY PARAMETERS

`max`	integer `>= 1` Default: `null` How many entries to download. Leave empty to download all activity.

Responses

› 200

GET `/api/user/puzzle-activity`

Response samples

200

Content type

application/x-ndjson

Copy Expand all Collapse all

```
{
    "id": 70232,
    "date": 1514505150384,
    "rating": 1982,
    "ratingDiff": 5,
    "puzzleRating": 1877
    }
```

Get users by ID

Get up to 300 users by their IDs. Users are returned in the order same order as the IDs.

The method is `POST` so a longer list of IDs can be sent in the request body.

AUTHORIZATIONS: None

REQUEST BODY SCHEMA: text/plain

User IDs separated by commas.
string

Responses

> 200

POST /api/users

Request samples

Payload

Content type
text/plain

Copy

```
aliquantus,chess-network,lovlas
```

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
 - {
        "id": "georges",
        "username": "Georges",
        "online": true,
      + "perfs": { … },
        "createdAt": 1290415680000,
        "disabled": false,
        "engine": false,
        "booster": false,
      + "profile": { … },
        "seenAt": 1522636452014,
        "patron": true,
      + "playTime": { … },
        "language": "en-GB",
        "title": "NM"
    }
]
```

Get members of a team

Members are sorted by reverse chronological order of joining the team (most recent first).

Members are streamed as ndjson, i.e. one JSON object per line.

AUTHORIZATIONS: None

PATH PARAMETERS

`teamId` required	string Example: `coders`

Responses

> 200

GET `/api/team/{teamId}/users`

Response samples

200

Content type

application/x-ndjson

Copy Expand all Collapse all

```
{
    "id": "georges",
    "username": "Georges",
    "online": true,
- "perfs": {
    + "chess960": { … },
    + "atomic": { … },
    + "racingKings": { … },
    + "ultraBullet": { … },
    + "blitz": { … },
    + "kingOfTheHill": { … },
    + "bullet": { … },
    + "correspondence": { … },
    + "horde": { … },
    + "puzzle": { … },
    + "classical": { … },
    + "rapid": { … }
  },
  "createdAt": 1290415680000,
  "disabled": false,
  "engine": false,
  "booster": false,
- "profile": {
      "country": "EC",
      "location": "string",
      "bio": "Free bugs!",
      "firstName": "Thibault",
      "lastName": "Duplessis",
      "fideRating": 1500,
      "uscfRating": 1500,
      "ecfRating": 1500,
      "links": "github.com/ornicar\r\ntwitter.com/ornicar"
  },
  "seenAt": 1522636452014,
  "patron": true,
- "playTime": {
      "total": 3296897,
      "tv": 12134
  },
  "language": "en-GB",
  "title": "NM",
  "url": "https://lichess.org/@/georges",
  "playing": "https://lichess.org/yqfLYJ5E/black",
  "nbFollowing": 299,
      "nbFollowers": 2735,
      "completionRate": 97,
    - "count": {
          "all": 9265,
          "rated": 7157,
          "ai": 531,
          "draw": 340,
          "drawH": 331,
          "loss": 4480,
          "lossH": 4207,
          "win": 4440,
          "winH": 4378,
          "bookmark": 71,
          "playing": 6,
          "import": 66,
          "me": 0
      },
      "streaming": false,
      "followable": true,
      "following": false,
      "blocking": false,
      "followsYou": false
  }
```

Get live streamers

Get basic info about currently streaming users.

This API is very fast and cheap on lichess side. So you can call it quite often (like once every 5 seconds).

AUTHORIZATIONS: None **AND** SameOrigin

Responses

> 200

GET /streamer/live

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
  - {
        "id": "aliquantus",
        "name": "Aliquantus"
    },
  - {
        "id": "chess-network",
        "name": "Chess-Network",
        "title": "NM",
        "playing": true,
        "patron": true
    }
]
```

Get crosstable

Get total number of games, and current score, of any two users.

If the `matchup` flag is provided, and the users are currently playing, also gets the current match game number and scores.

AUTHORIZATIONS: None

PATH PARAMETERS

`user1` required	string
`user2` required	string

QUERY PARAMETERS

`matchup`	boolean Whether to get the current match data, if any

Responses

› 200

GET `/api/crosstable/{user1}/{user2}`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
  - "users": {
        "neio": 201.5,
        "thibault": 144.5
    },
    "nbGames": 346,
  - "matchup": {
      + "users": { … },
        "nbGames": 87
    }
}
```

Relations

Access relations between users.

Get users followed by a user

Users are streamed as [ndjson](http://ndjson.org/), i.e. one JSON object per line.

AUTHORIZATIONS: None

PATH PARAMETERS

`username` required	string Example: `thibault`

Responses

> 200

GET `/api/user/{username}/following`

Response samples

200

Content type

application/x-ndjson

Copy Expand all Collapse all

```
{
    "id": "georges",
    "username": "Georges",
    "online": true,
  - "perfs": {
      + "chess960": { … },
      + "atomic": { … },
      + "racingKings": { … },
      + "ultraBullet": { … },
      + "blitz": { … },
      + "kingOfTheHill": { … },
      + "bullet": { … },
      + "correspondence": { … },
      + "horde": { … },
      + "puzzle": { … },
      + "classical": { … },
      + "rapid": { … }
    },
    "createdAt": 1290415680000,
    "disabled": false,
    "engine": false,
    "booster": false,
  - "profile": {
        "country": "EC",
        "location": "string",
        "bio": "Free bugs!",
        "firstName": "Thibault",
        "lastName": "Duplessis",
        "fideRating": 1500,
        "uscfRating": 1500,
        "ecfRating": 1500,
        "links": "github.com/ornicar\r\ntwitter.com/ornicar"
    },
    "seenAt": 1522636452014,
    "patron": true,
  - "playTime": {
        "total": 3296897,
        "tv": 12134
    },
    "language": "en-GB",
    "title": "NM",
    "url": "https://lichess.org/@/georges",
    "playing": "https://lichess.org/yqfLYJ5E/black",
    "nbFollowing": 299,
    "nbFollowers": 2735,
    "completionRate": 97,
  - "count": {
        "all": 9265,
        "rated": 7157,
        "ai": 531,
        "draw": 340,
        "drawH": 331,
        "loss": 4480,
        "lossH": 4207,
        "win": 4440,
        "winH": 4378,
        "bookmark": 71,
        "playing": 6,
        "import": 66,
        "me": 0
    },
    "streaming": false,
        "followable": true,
        "following": false,
        "blocking": false,
        "followsYou": false
    }
```

Get users who follow a user

Users are streamed as ndjson, i.e. one JSON object per line.

AUTHORIZATIONS: None

PATH PARAMETERS

`username` required	string Example: `thibault`

Responses

200

GET /api/user/{username}/followers

Response samples

200

Content type

application/x-ndjson

Copy Expand all Collapse all

```
{
    "id": "georges",
    "username": "Georges",
    "online": true,
  - "perfs": {
      + "chess960": { … },
      + "atomic": { … },
      + "racingKings": { … },
      + "ultraBullet": { … },
      + "blitz": { … },
      + "kingOfTheHill": { … },
      + "bullet": { … },
      + "correspondence": { … },
      + "horde": { … },
      + "puzzle": { … },
      + "classical": { … },
      + "rapid": { … }
    },
    "createdAt": 1290415680000,
    "disabled": false,
    "engine": false,
    "booster": false,
  - "profile": {
        "country": "EC",
        "location": "string",
        "bio": "Free bugs!",
        "firstName": "Thibault",
        "lastName": "Duplessis",
        "fideRating": 1500,
        "uscfRating": 1500,
        "ecfRating": 1500,
        "links": "github.com/ornicar\r\ntwitter.com/ornicar"
    },
    "seenAt": 1522636452014,
    "patron": true,
  - "playTime": {
        "total": 3296897,
        "tv": 12134
    },
        "language": "en-GB",
        "title": "NM",
        "url": "https://lichess.org/@/georges",
        "playing": "https://lichess.org/yqfLYJ5E/black",
        "nbFollowing": 299,
        "nbFollowers": 2735,
        "completionRate": 97,
      - "count": {
            "all": 9265,
            "rated": 7157,
            "ai": 531,
            "draw": 340,
            "drawH": 331,
            "loss": 4480,
            "lossH": 4207,
            "win": 4440,
            "winH": 4378,
            "bookmark": 71,
            "playing": 6,
            "import": 66,
            "me": 0
        },
        "streaming": false,
        "followable": true,
        "following": false,
        "blocking": false,
        "followsYou": false
    }
```

Games

Access games played on Lichess. https://lichess.org/games

Export one game

Download one game in either PGN or JSON format. If the game is ongoing, the 3 last moves are omitted.

AUTHORIZATIONS: None

PATH PARAMETERS

`gameId` required
string
The game ID (8 characters).

QUERY PARAMETERS

`moves`
boolean
Default: `true`
Include the PGN moves.

`pgnInJson`
boolean
Default: `false`
Include the full PGN within the JSON response, in a `pgn` field.

`tags`
boolean
Default: `true`
Include the PGN tags.

`clocks`
boolean
Default: `true`
Include clock comments in the PGN moves, when available.

Example: `2. exd5 { [%clk 1:01:27] } e5 { [%clk 1:01:28] }`

`evals`
boolean
Default: `true`
Include analysis evaluation comments in the PGN, when available.

Example: `12. Bxf6 { [%eval 0.23] } a3 { [%eval -1.09] }`

`opening`
boolean
Default: `true`
Include the opening name.

Example: `[Opening "King's Gambit Accepted, King's Knight Gambit"]`

`literate` boolean

Default: `false`

Insert textual annotations in the PGN about the opening, analysis variations, mistakes, and

Example: `5... g4? { (-0.98 → 0.60) Mistake. Best move was h6. } (5... h6 dxc6)`

`players` string

URL of a text file containing real names and ratings, to replace Lichess usernames and rat

https://gist.githubusercontent.com/ornicar/6bfa91eb61a2dcae7bcd14cce1b2a4eb/raw/7

Responses

> 200

GET /game/export/{gameId}

Response samples

200

Content type

application/x-chess-pgn

```
No sample
```

Export ongoing game of a user

Download the ongoing game, or the last game played, of a user. Available in either PGN or JSON format. If the game is ongoing, the 3 last moves are omitted.

AUTHORIZATIONS: None

PATH PARAMETERS

`username`
required
string

QUERY PARAMETERS

`moves`
boolean
Default: `true`
Include the PGN moves.

`pgnInJson`
boolean
Default: `false`
Include the full PGN within the JSON response, in a `pgn` field.

`tags`
boolean
Default: `true`
Include the PGN tags.

`clocks`
boolean
Default: `true`
Include clock comments in the PGN moves, when available.

Example: `2. exd5 { [%clk 1:01:27] } e5 { [%clk 1:01:28] }`

`evals`
boolean
Default: `true`
Include analysis evaluation comments in the PGN, when available.

Example: `12. Bxf6 { [%eval 0.23] } a3 { [%eval -1.09] }`

`opening`
boolean
Default: `true`
Include the opening name.

Example: `[Opening "King's Gambit Accepted, King's Knight Gambit"]`

`literate`
boolean
Default: `false`
Insert textual annotations in the PGN about the opening, analysis variations, mistakes, and

Example: `5... g4? { (-0.98 → 0.60) Mistake. Best move was h6. } (5... h6 dxc6)`

`players` string

URL of a text file containing real names and ratings, to replace Lichess usernames and rat

https://gist.githubusercontent.com/ornicar/6bfa91eb61a2dcae7bcd14cce1b2a4eb/raw/7

Responses

› 200

GET `/api/user/{username}/current-game`

Response samples

200

Content type

application/x-chess-pgn

`No sample`

Export games of a user

Download all games of any user in PGN or ndjson format.

Games are sorted by reverse chronological order (most recent first)

We recommend streaming the response, for it can be very long. https://lichess.org/@/german11 for instance has more than 320,000 games.

The game stream is throttled, depending on who is making the request:

- Anonymous request: 20 games per second
- OAuth2 authenticated request: 30 games per second
- Authenticated, downloading your own games: 60 games per second

AUTHORIZATIONS: (None) **OR** (OAuth2)

PATH PARAMETERS

`username` `required`
string
The user name.

QUERY PARAMETERS

`since`
integer `>= 1356998400070`
Default: `"Account creation date"`
Download games played since this timestamp.

`until`
integer `>= 1356998400070`
Default: `"Now"`
Download games played until this timestamp.

`max`
integer `>= 1`
Default: `null`
How many games to download. Leave empty to download all games.

`vs`
string
[Filter] Only games played against this opponent

`rated`
boolean
Default: `null`
[Filter] Only rated (`true`) or casual (`false`) games

`perfType`
string
Default: `null`
Enum: `"ultraBullet"` `"bullet"` `"blitz"` `"rapid"` `"classical"` `"correspor`
`"atomic"` `"horde"` `"kingOfTheHill"` `"racingKings"` `"threeCheck"`
[Filter] Only games in these speeds or variants.

Multiple perf types can be specified, separated by a comma.

Example: blitz,rapid,classical

`color` string

Default: `null`

Enum: `"white"` `"black"`

[Filter] Only games played as this color.

`analysed` boolean

Default: `null`

[Filter] Only games with or without a computer analysis available

`ongoing` boolean

Default: `false`

[Filter] Also include ongoing games

`moves` boolean

Default: `true`

Include the PGN moves.

`pgnInJson` boolean

Default: `false`

Include the full PGN within the JSON response, in a `pgn` field. The response type must be
header.

`tags` boolean

Default: `true`

Include the PGN tags.

`clocks` boolean

Default: `false`

Include clock comments in the PGN moves, when available.

Example: `2. exd5 { [%clk 1:01:27] } e5 { [%clk 1:01:28] }`

`evals` boolean

Default: `false`

Include analysis evaluation comments in the PGN, when available.

Example: `12. Bxf6 { [%eval 0.23] } a3 { [%eval -1.09] }`

`opening` boolean

Default: `false`

Include the opening name.

Example: `[Opening "King's Gambit Accepted, King's Knight Gambit"]`

players string
URL of a text file containing real names and ratings, to replace Lichess usernames and rat
https://gist.githubusercontent.com/ornicar/6bfa91eb61a2dcae7bcd14cce1b2a4eb/raw/7

Responses

› 200

GET `/api/games/user/{username}`

Response samples

200

Content type
application/x-chess-pgn

```
No sample
```

Export games by IDs

Download games by IDs in PGN or ndjson format.

Games are sorted by reverse chronological order (most recent first)

The method is `POST` so a longer list of IDs can be sent in the request body.

300 IDs can be submitted. If the request uses OAuth2, 500 IDs can be submitted.

AUTHORIZATIONS: SameOrigin

QUERY PARAMETERS

`moves`
boolean
Default: `true`
Include the PGN moves.

`pgnInJson`
boolean
Default: `false`
Include the full PGN within the JSON response, in a `pgn` field.

`tags`
boolean
Default: `true`
Include the PGN tags.

`clocks`
boolean
Default: `false`
Include clock comments in the PGN moves, when available.

Example: `2. exd5 { [%clk 1:01:27] } e5 { [%clk 1:01:28] }`

`evals`
boolean
Default: `false`
Include analysis evaluation comments in the PGN, when available.

Example: `12. Bxf6 { [%eval 0.23] } a3 { [%eval -1.09] }`

`opening`
boolean
Default: `false`
Include the opening name.

Example: `[Opening "King's Gambit Accepted, King's Knight Gambit"]`

`players`
string
URL of a text file containing real names and ratings, to replace Lichess usernames and rat
https://gist.githubusercontent.com/ornicar/6bfa91eb61a2dcae7bcd14cce1b2a4eb/raw/7

REQUEST BODY SCHEMA: text/plain

Game IDs separated by commas.
string

Responses

> 200

POST /games/export/_ids

Request samples

Payload

Content type

text/plain

Copy

```
TJxUmbWK,4OtIh2oh,ILwozzRZ
```

Response samples

200

Content type

application/x-chess-pgn

```
No sample
```

Stream current games

Stream the games played between a list of users, in real time. Only games where **both players** are part of the list are included.

Maximum number of users: 300.

Games are streamed as ndjson, i.e. one JSON object per line.

The method is `POST` so a longer list of IDs can be sent in the request body.

AUTHORIZATIONS: None

REQUEST BODY SCHEMA: text/plain

Up to 300 user IDs separated by commas.
string

Responses

> 200

POST /api/stream/games-by-users

Request samples

Payload

Content type
text/plain

Copy

```
aliquantus,chess-network,lovlas
```

Response samples

200

Content type
application/x-ndjson

Copy Expand all Collapse all

```
{
    "id": "A5fcMO3k",
    "rated": true,
    "variant": "standard",
    "speed": "bullet",
    "perf": "bullet",
    "createdAt": 1525789431889,
    "status": 20,
  - "clock": {
        "initial": 60,
        "increment": 0,
        "totalTime": 60
    },
  - "players": {
      + "white": { … },
      + "black": { … }
    }
}
```

Get ongoing games

Get the ongoing games of the current user. Real-time and correspondence games are included. The most urgent games are listed first.

AUTHORIZATIONS: OAuth2

QUERY PARAMETERS

nb	integer `[ 1 .. 50 ]` Default: `9` Max number of games to fetch

Responses

› 200

GET /api/account/playing

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
  - "nowPlaying": [
      + { … }
    ]
}
```

Get current TV games

Get basic info about the best games being played for each speed and variant, but also computer games and bot games.

See lichess.org/tv.

AUTHORIZATIONS: SameOrigin

Responses

> 200

GET /tv/channels

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
  - "Bot": {
      + "user": { … },
        "rating": 2660,
        "gameId": "Zznv9MIl"
    },
  - "Blitz": {
      + "user": { … },
        "rating": 2603,
        "gameId": "hTJ4v7Mp"
    },
  - "Racing Kings": {
      + "user": { … },
        "rating": 2123,
        "gameId": "lgCDl5Of"
    },
  - "UltraBullet": {
      + "user": { … },
        "rating": 2338,
        "gameId": "NEY6OQ32"
    },
  - "Bullet": {
      + "user": { … },
        "rating": 2499,
        "gameId": "5LgyE516"
    },
- "Classical": {
    + "user": { … },
      "rating": 1806,
      "gameId": "k3oLby6N"
  },
- "Three-check": {
    + "user": { … },
      "rating": 1978,
      "gameId": "Og5RCvmu"
  },
- "Antichess": {
    + "user": { … },
      "rating": 2103,
      "gameId": "toCr41yx"
  },
- "Computer": {
    + "user": { … },
      "rating": 2314,
      "gameId": "TkI4qZxu"
  },
- "Horde": {
    + "user": { … },
      "rating": 1803,
      "gameId": "oMofN63H"
  },
- "Rapid": {
    + "user": { … },
      "rating": 2289,
      "gameId": "IcWOl8ee"
  },
- "Atomic": {
    + "user": { … },
      "rating": 2210,
      "gameId": "tvMxtCMN"
  },
- "Crazyhouse": {
    + "user": { … },
      "rating": 2397,
      "gameId": "i3gTZlUb"
  },
    - "Chess960": {
        + "user": { … },
          "rating": 1782,
          "gameId": "lrXLcedu"
      },
    - "King of the Hill": {
        + "user": { … },
          "rating": 1500,
          "gameId": "DsQn8aEV"
      },
    - "Top Rated": {
        + "user": { … },
          "rating": 2603,
          "gameId": "hTJ4v7Mp"
      }
  }
```

Import one game

Import a game from PGN. See https://lichess.org/paste.

Rate limiting: 200 games per hour for OAuth requests, 100 games per hour for anonymous requests.

To broadcast ongoing games, consider pushing to a broadcast instead.

AUTHORIZATIONS: (None) **OR** (OAuth2)

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

A single game to import

`pgn`	string The PGN. It can contain only one game. Most standard tags are supported.

Responses

> 200

POST /api/import

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "id": "R6iLjwz5",
    "url": "https://lichess.org/R6iLjwz5"
}
```

Teams

Access and manage Lichess teams and their members. https://lichess.org/team

Get team swiss tournaments

Get all swiss tournaments of a team.

Tournaments are sorted by reverse chronological order of start date (last starting first).

Tournaments are streamed as ndjson, i.e. one JSON object per line.

AUTHORIZATIONS: None

PATH PARAMETERS

`teamId` required

string

Example: `coders`

QUERY PARAMETERS

`max`

integer `>= 1`

Default: `100`

How many tournaments to download.

Responses

› 200

GET `/api/team/{teamId}/swiss`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
  - {
        "rated": true,
      + "clock": { … },
        "createdBy": "thibault",
      + "greatPlayer": { … },
        "id": "ZmKWCOye",
        "name": "Wang",
        "nbOngoing": 0,
        "nbPlayers": 0,
        "nbRounds": 2,
      + "nextRound": { … },
      + "quote": { … },
        "round": 0,
        "startsAt": "2020-05-11T12:23:18.233-06:00",
        "status": "created",
        "variant": "standard"
    }
]
```

Get a single team

Infos about a team

AUTHORIZATIONS: None

PATH PARAMETERS

`teamId` `required`	string

Responses

> 200

GET /api/team/{teamId}

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "id": "coders",
    "name": "Coders",
    "description": "There are 10 kinds of people in the world: those who ur
    "open": false,
  - "leader": {
        "name": "Chess-Network",
        "title": "NM",
        "patron": true,
        "id": "chess-network"
    },
  - "leaders": [
      + { … }
    ],
    "nbMembers": 3129
}
```

Get popular teams

Paginator of the most popular teams.

AUTHORIZATIONS: None

QUERY PARAMETERS

`page`	number Default: `1` Example: `page=1`

Responses

› 200

GET `/api/team/all`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
  - {
        "id": "coders",
        "name": "Coders",
        "description": "There are 10 kinds of people in the world: those wh
        "open": false,
      + "leader": { … },
      + "leaders": [ … ],
        "nbMembers": 3129
    }
]
```

Teams of a player

All the teams a player is a member of.

AUTHORIZATIONS: None

PATH PARAMETERS

`username` required	string Example: `thibault`

Responses

> 200

GET `/api/team/of/{username}`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
    - {
          "id": "coders",
          "name": "Coders",
          "description": "There are 10 kinds of people in the world: those wh
          "open": false,
        + "leader": { … },
        + "leaders": [ … ],
          "nbMembers": 3129
      }
  ]
```

Search teams

Paginator of team search results for a keyword.

AUTHORIZATIONS: None

QUERY PARAMETERS

`text`	string Example: `text=coders`
`page`	number Default: `1` Example: `page=1`

Responses

> 200

```
GET /api/team/search
GET /api/team/search
```

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
[
  - {
        "id": "coders",
        "name": "Coders",
        "description": "There are 10 kinds of people in the world: those wh
        "open": false,
      + "leader": { … },
      + "leaders": [ … ],
        "nbMembers": 3129
    }
]
```

Get members of a team

Members are sorted by reverse chronological order of joining the team (most recent first).

Members are streamed as ndjson, i.e. one JSON object per line.

AUTHORIZATIONS: None

PATH PARAMETERS

`teamId` required	string Example: `coders`

Responses

> 200

GET /api/team/{teamId}/users

Response samples

200

Content type

application/x-ndjson

Copy Expand all Collapse all

```
{
    "id": "georges",
    "username": "Georges",
    "online": true,
  - "perfs": {
      + "chess960": { … },
      + "atomic": { … },
      + "racingKings": { … },
      + "ultraBullet": { … },
      + "blitz": { … },
      + "kingOfTheHill": { … },
      + "bullet": { … },
      + "correspondence": { … },
      + "horde": { … },
      + "puzzle": { … },
      + "classical": { … },
      + "rapid": { … }
    },
    "createdAt": 1290415680000,
    "disabled": false,
    "engine": false,
      "booster": false,
    - "profile": {
          "country": "EC",
          "location": "string",
          "bio": "Free bugs!",
          "firstName": "Thibault",
          "lastName": "Duplessis",
          "fideRating": 1500,
          "uscfRating": 1500,
          "ecfRating": 1500,
          "links": "github.com/ornicar\r\ntwitter.com/ornicar"
      },
      "seenAt": 1522636452014,
      "patron": true,
    - "playTime": {
          "total": 3296897,
          "tv": 12134
      },
      "language": "en-GB",
      "title": "NM",
      "url": "https://lichess.org/@/georges",
      "playing": "https://lichess.org/yqfLYJ5E/black",
      "nbFollowing": 299,
      "nbFollowers": 2735,
      "completionRate": 97,
    - "count": {
          "all": 9265,
          "rated": 7157,
          "ai": 531,
          "draw": 340,
          "drawH": 331,
          "loss": 4480,
          "lossH": 4207,
          "win": 4440,
          "winH": 4378,
          "bookmark": 71,
          "playing": 6,
          "import": 66,
          "me": 0
      },
        "streaming": false,
        "followable": true,
        "following": false,
        "blocking": false,
        "followsYou": false
    }
```

Get team arena tournaments

Get all arena tournaments relevant to a team.

Tournaments are sorted by reverse chronological order of start date (last starting first).

Tournaments are streamed as ndjson, i.e. one JSON object per line.

AUTHORIZATIONS: None

QUERY PARAMETERS

`max` integer `>= 1`
Default: `100`
How many tournaments to download.

Responses

> 200

GET `/api/team/{teamId}/arena`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
  - {
        "id": "2VoKhb8g",
        "createdBy": "lichess",
        "system": "arena",
        "minutes": 57,
      + "clock": { … },
        "rated": true,
        "fullName": "Petrov's Defence Blitz Arena",
        "nbPlayers": 156,
      + "variant": { … },
        "startsAt": 1599886846000,
        "finishesAt": 1599890266000,
        "status": 30,
      + "perf": { … },
        "secondsToStart": 0,
        "hasMaxRating": false,
        "private": false,
      + "position": { … },
      + "schedule": { … },
      + "winner": { … }
    }
]
```

Join a team

Join a team. If the team join policy requires a confirmation, and the team owner is not the oAuth app owner, and the `message` field is not set, then the call fails with `403 Forbidden`.

AUTHORIZATIONS: OAuth2 (`team:write`) **AND** SameOrigin

PATH PARAMETERS

`teamId` `required`

string

Example: `coders`

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`message`

string

Optional request message, if the team is not open.

Responses

> 200

POST `/team/{teamId}/join`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Leave a team

Leave a team

Leave a team.

- https://lichess.org/team

AUTHORIZATIONS: OAuth2 (`team:write`) **AND** SameOrigin

PATH PARAMETERS

`teamId` `required`	string Example: `coders`

Responses

> 200

POST `/team/{teamId}/quit`

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Kick a user from your team

Kick a member out of one of your teams.

- https://lichess.org/team

AUTHORIZATIONS: OAuth2 (`team:write`) **AND** SameOrigin

PATH PARAMETERS

`teamId` `required`	string Example: `coders`
`userId` `required`	string Example: `neio`

Responses

> 200

POST `/team/{teamId}/kick/{userId}`

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Message all members

Send a private message to all members of a team. You must own the team.

AUTHORIZATIONS: OAuth2 (`team:write`)

PATH PARAMETERS

`teamId` `required`	string Example: `coders`

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`message`	string The message to send to all your team members.

Responses

› 200

› 400

POST `/team/{teamId}/pm-all`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Board

Play on Lichess with physical boards and third-party clients. Works with normal Lichess accounts. Engine play or assistance is forbidden.

Features

- Stream incoming chess moves
- Play chess moves
- Read and write in the player and spectator chats
- Receive, create and accept (or decline) challenges
- Abort and resign games
- Compatible with normal Lichess accounts

Restrictions

- Engine assistance is strictly forbidden
- Casual games: Rapid, Classical and Correspondence only
- Rated games: Estimated duration >= 1200. That includes Classical, Correspondence, and some Rapid time controls such as 15+10 and 20+0.

Links

- Announcement
- Certabo support
- Lichs (play from command-line)

Stream incoming events

Stream the events reaching a lichess user in real time as ndjson.

Each line is a JSON object containing a `type` field. Possible values are:

- `gameStart` Start of a game
- `gameFinish` Completion of a game
- `challenge` A player sends you a challenge
- `challengeCanceled` A player cancels their challenge to you
- `challengeDeclined` The opponent declines your challenge

  When the stream opens, all current challenges and games are sent.

AUTHORIZATIONS: OAuth2 (`challenge:read,` `bot:play,` `board:play`)

Responses

— 200

GET /api/stream/event

Response samples

200

Content type

text/plain

Copy

```
{"type":"challenge","challenge":{"id":"7pGLxJ4F","status":"created","chall
{"type":"gameStart","game":{"id":"1lsvP62l"}}
```

Create a seek

Create a public seek, to start a game with a random player.

The response is streamed but doesn't contain any information. **Keep the connection open to keep the seek active**.

If the client closes the connection, the seek is canceled.

If the seek is accepted, or expires, the server closes the connection.

Make sure to also have an Event stream open, to be notified when a game starts. We recommend opening the Event stream first, then the seek stream. This way, you won't miss the game event if the seek is accepted immediately.

AUTHORIZATIONS: OAuth2 (`board:play`)

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

Parameters of the seek

`rated`

boolean
Default: `false`
Whether the game is rated and impacts players ratings.

`time` required

number `[ 0 .. 180 ]`
Clock initial time in minutes.

`increment` required

integer `[ 0 .. 180 ]`
Clock increment in seconds.

`variant`

string
Default: `"standard"`
Enum: `"standard"` `"chess960"` `"crazyhouse"` `"antichess"` `"atomic"` `"horde"` `"kingOfTheHill"` `"racingKings"` `"threeCheck"`
The variant of the game.

`color`

string
Default: `"random"`
Enum: `"random"` `"white"` `"black"`
The color to play. Better left empty to automatically get 50% white.

`ratingRange` string
The rating range of potential opponents. Better left empty. Example: 1500-1800

Responses

— 200

> 400

POST `/api/board/seek`

Response samples

200 400

Content type
text/plain

Copy

Stream Board game state

Stream the state of a game being played with the Board API, as ndjson.

Use this endpoint to get updates about the game in real-time, with a single request.

Each line is a JSON object containing a `type` field. Possible values are:

- `gameFull` Full game data. All values are immutable, except for the `state` field.
- `gameState` Current state of the game. Immutable values not included. Sent when a move is

played, a draw is offered, or when the game ends.

- `chatLine` Chat message sent by a user in the `room` "player" or "spectator".

The first line is always of type `gameFull`.

AUTHORIZATIONS: OAuth2 (`board:play`)

PATH PARAMETERS

`gameId` `required`	string Example: `5IrD6Gzz`

Responses

› 200

GET `/api/board/game/stream/{gameId}`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

[

```
  - {
        "type": "gameFull",
        "id": "5IrD6Gzz",
        "rated": true,
      + "variant": { … },
      + "clock": { … },
        "speed": "classical",
      + "perf": { … },
        "createdAt": 1523825103562,
      + "white": { … },
      + "black": { … },
        "initialFen": "startpos",
      + "state": { … }
    },
  - {
        "type": "gameState",
        "moves": "e2e4 c7c5 f2f4 d7d6 g1f3 b8c6 f1c4 g8f6 d2d3 g7g6 e1g1 f8
        "wtime": 7598040,
        "btime": 8395220,
        "winc": 10000,
        "binc": 10000,
        "status": "started"
    },
  - {
        "type": "chatLine",
        "username": "thibault",
        "text": "Good luck, have fun",
        "room": "player"
    },
  - {
        "type": "chatLine",
        "username": "lovlas",
        "text": "!eval",
        "room": "spectator"
    },
  - {
        "type": "gameState",
        "moves": "e2e4 c7c5 f2f4 d7d6 g1f3 b8c6 f1c4 g8f6 d2d3 g7g6 e1g1 f8
        "wtime": 7598040,
        "btime": 8395220,
        "winc": 10000,
        "binc": 10000,
        "status": "resign",
        "winner": "black"
    }
 ]
```

Make a Board move

Make a move in a game being played with the Board API.

The move can also contain a draw offer/agreement.

AUTHORIZATIONS: OAuth2 (`board:play`)

PATH PARAMETERS

`gameId` required	string Example: `5IrD6Gzz`
`move` required	string Example: `e2e4` The move to play, in UCI format

QUERY PARAMETERS

`offeringDraw`	boolean Whether to offer (or agree to) a draw

Responses

> 200

> 400

POST /api/board/game/{gameId}/move/{move}

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Write in the chat

Post a message to the player or spectator chat, in a game being played with the Board API.

AUTHORIZATIONS: OAuth2 (`board:play`)

PATH PARAMETERS

`gameId` `required`	string Example: `5IrD6Gzz`

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`room`	string

required	Enum: `"player"` `"spectator"`
text required	string

Responses

› 200

› 400

POST `/api/board/game/{gameId}/chat`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Abort a game

Abort a game being played with the Board API.

AUTHORIZATIONS: OAuth2 (`board:play`)

PATH PARAMETERS

`gameId` required	string Example: `5IrD6Gzz`

Responses

› 200

› 400

POST `/api/board/game/{gameId}/abort`

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Resign a game

Resign a game being played with the Board API.

AUTHORIZATIONS: OAuth2 (`board:play`)

PATH PARAMETERS

`gameId` `required`	string Example: `5IrD6Gzz`

Responses

› 200

› 400

POST `/api/board/game/{gameId}/resign`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Handle draw offers

Create/accept/decline draw offers.

- `yes` : Offer a draw, or accept the opponent's draw offer.
- `no` : Decline a draw offer from the opponent.

AUTHORIZATIONS: OAuth2 (`board:play`)

PATH PARAMETERS

`gameId` required	string Example: `5IrD6Gzz`
`accept` required	boolean Example: `yes`

Responses

> 200

> 400

POST `/api/board/game/{gameId}/draw/{accept}`

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Bot

Play on Lichess as a bot. Allows engine play. Read the blog post announcement of lichess bots.

Only works with Bot accounts.

Features

- Stream incoming chess moves
- Play chess moves
- Read and write in the player and spectator chats
- Receive, create and accept (or decline) challenges
- Abort and resign games
- Engine assistance is allowed

Restrictions

- Bots can only play challenge games: pools and tournaments are off-limits
- Bots cannot play UltraBullet (¼+0) because it requires making too many requests. But 0+1 and ½+0 are allowed.

Integrations

- Python3 lichess-bot (official)
- JavaScript bot-o-tron
- Golang lichess-bot
- Yours? Please make an issue or pull request.

Links

- Announcement
- Join the Lichess Bots team with your bot account
- Get help in the discord channel
- Watch Lichess Bot TV

Stream incoming events

Stream the events reaching a lichess user in real time as ndjson.

Each line is a JSON object containing a `type` field. Possible values are:

- `gameStart` Start of a game
- `gameFinish` Completion of a game
- `challenge` A player sends you a challenge
- `challengeCanceled` A player cancels their challenge to you
- `challengeDeclined` The opponent declines your challenge

When the stream opens, all current challenges and games are sent.

AUTHORIZATIONS: OAuth2 (`challenge:read,` `bot:play,` `board:play`)

Responses

— 200

GET /api/stream/event

Response samples

200

Content type

text/plain

Copy

```
{"type":"challenge","challenge":{"id":"7pGLxJ4F","status":"created","chall
{"type":"gameStart","game":{"id":"1lsvP62l"}}
```

Upgrade to Bot account

Upgrade a lichess player account into a Bot account. Only Bot accounts can use the Bot API.

The account **cannot have played any game** before becoming a Bot account. The upgrade is **irreversible**. The account will only be able to play as a Bot.

To upgrade an account to Bot, use the official lichess-bot client, or follow these steps:

- Create an API access token with "Play bot moves" permission.
- `curl -d '' https://lichess.org/api/bot/account/upgrade -H "Authorization: Bearer <yourTokenHere>"`

To know if an account has already been upgraded, use the Get my profile API: the `title` field should be set to `BOT`.

AUTHORIZATIONS: OAuth2 (`bot:play`)

Responses

› 200

› 400

POST /api/bot/account/upgrade

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

{

```
    "ok": true
}
```

Stream Bot game state

Stream the state of a game being played with the Bot API, as ndjson.

Use this endpoint to get updates about the game in real-time, with a single request.

Each line is a JSON object containing a `type` field. Possible values are:

- `gameFull` Full game data. All values are immutable, except for the `state` field.
- `gameState` Current state of the game. Immutable values not included.
- `chatLine` Chat message sent by a user (or the bot itself) in the `room` "player" or "spectator".

The first line is always of type `gameFull`.

AUTHORIZATIONS: OAuth2 (`bot:play`)

PATH PARAMETERS

`gameId`
 `required`
	string
	 Example: `5IrD6Gzz`

Responses

> 200

GET `/api/bot/game/stream/{gameId}`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
 - {
       "type": "gameFull",
       "id": "5IrD6Gzz",
       "rated": true,
     + "variant": { … },
     + "clock": { … },
       "speed": "classical",
     + "perf": { … },
       "createdAt": 1523825103562,
     + "white": { … },
     + "black": { … },
       "initialFen": "startpos",
     + "state": { … }
   },
 - {
       "type": "gameState",
       "moves": "e2e4 c7c5 f2f4 d7d6 g1f3 b8c6 f1c4 g8f6 d2d3 g7g6 e1g1 f8
       "wtime": 7598040,
       "btime": 8395220,
       "winc": 10000,
       "binc": 10000,
       "status": "started"
   },
 - {
       "type": "chatLine",
       "username": "thibault",
       "text": "Good luck, have fun",
       "room": "player"
   },
    - {
          "type": "chatLine",
          "username": "lovlas",
          "text": "!eval",
          "room": "spectator"
      },
    - {
          "type": "gameState",
          "moves": "e2e4 c7c5 f2f4 d7d6 g1f3 b8c6 f1c4 g8f6 d2d3 g7g6 e1g1 f8
          "wtime": 7598040,
          "btime": 8395220,
          "winc": 10000,
          "binc": 10000,
          "status": "resign",
          "winner": "black"
      }
  ]
```

Make a Bot move

Make a move in a game being played with the Bot API.

The move can also contain a draw offer/agreement.

AUTHORIZATIONS: OAuth2 (`bot:play`)

PATH PARAMETERS

`gameId` `required`

string

Example: `5IrD6Gzz`

`move` `required`

string

Example: `e2e4`

The move to play, in UCI format

QUERY PARAMETERS

`offeringDraw` boolean
Whether to offer (or agree to) a draw

Responses

› 200

› 400

POST `/api/bot/game/{gameId}/move/{move}`

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Write in the chat

Post a message to the player or spectator chat, in a game being played with the Bot API.

AUTHORIZATIONS: OAuth2 (`bot:play`)

PATH PARAMETERS

`gameId` required

string

Example: `5IrD6Gzz`

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`room` required

string

Enum: `"player"` `"spectator"`

`text` required

string

Responses

› 200

› 400

POST `/api/bot/game/{gameId}/chat`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Abort a game

Abort a game being played with the Bot API.

AUTHORIZATIONS: OAuth2 (`bot:play`)

PATH PARAMETERS

`gameId` `required`	string Example: `5IrD6Gzz`

Responses

› 200

› 400

POST `/api/bot/game/{gameId}/abort`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Resign a game

Resign a game being played with the Bot API.

AUTHORIZATIONS: OAuth2 (`bot:play`)

PATH PARAMETERS

`gameId` `required`	string Example: `5IrD6Gzz`

Responses

› 200

› 400

POST `/api/bot/game/{gameId}/resign`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Challenges

Send and receive challenges to play.

Stream incoming events

Stream the events reaching a lichess user in real time as ndjson.

Each line is a JSON object containing a `type` field. Possible values are:

- `gameStart` Start of a game
- `gameFinish` Completion of a game
- `challenge` A player sends you a challenge
- `challengeCanceled` A player cancels their challenge to you
- `challengeDeclined` The opponent declines your challenge

  When the stream opens, all current challenges and games are sent.

AUTHORIZATIONS: OAuth2 (`challenge:read,` `bot:play,` `board:play`)

Responses

— 200

GET `/api/stream/event`

Response samples

200

Content type

text/plain

Copy

```
{"type":"challenge","challenge":{"id":"7pGLxJ4F","status":"created","chall
{"type":"gameStart","game":{"id":"1lsvP62l"}}
```

Create a challenge

Challenge someone to play. The targeted player can choose to accept or decline.

If the challenge is accepted, you will be notified on the event stream that a new game has started. The game ID will be the same as the challenge ID.

If you also have an OAuth token with `challenge:write` scope for the receiving user, you can make them accept the challenge immediately by setting the `acceptByToken` field.

AUTHORIZATIONS: OAuth2 (`challenge:write,` `bot:play,` `board:play`)

PATH PARAMETERS

`username` required
string
Example: `LeelaChess`

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

Parameters of the challenge

`rated`
boolean
Game is rated and impacts players ratings

`clock.limit`
number `[ 0 .. 10800 ]`
Clock initial time in seconds. If empty, a correspondence game is created.

`clock.increment`
integer `[ 0 .. 60 ]`

Clock increment in seconds. If empty, a correspondence game is created.

`days`

integer `[ 1 .. 15 ]`

Days per move, for correspondence games. Clock settings must be omitted.

`color`

string

Default: `"random"`

Enum: `"random"` `"white"` `"black"`

Which color you get to play

`variant`

string

Default: `"standard"`

Enum: `"standard"` `"chess960"` `"crazyhouse"` `"antichess"` `"atomic"` `"horde"` `"kingOfTheHill"` `"racingKings"` `"threeCheck"`

The variant of the game

`fen`

string

Default: `"rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1"`

Custom initial position (in FEN). Variant must be standard, and the game cannot be rated.

`acceptByToken`

string

Immediately accept the challenge and create the game. Pass in an OAuth token (with the `challenge:write` scope) for the receiving user. On success, the response will contain a `game` field instead of a `challenge` field.

Responses

> 200

> 400

POST `/api/challenge/{username}`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "id": "VU0nyvsW",
    "url": "https://lichess.org/VU0nyvsW",
    "color": "random",
    "direction": "out",
  - "timeControl": {
        "increment": 2,
        "limit": 300,
        "show": "5+2",
        "type": "clock"
    },
  - "variant": {
        "key": "standard",
        "name": "Standard",
        "short": "Std"
    },
  - "challenger": {
        "id": "thibot",
        "name": "thibot",
        "online": true,
        "provisional": false,
        "rating": 1940,
        "title": "BOT"
    },
  - "destUser": {
        "id": "leelachess",
        "name": "LeelaChess",
        "online": true,
        "provisional": true,
        "rating": 2670,
        "title": "BOT"
    },
  - "perf": {
        "icon": ";",
        "name": "Correspondence"
    },
    "rated": true,
    "speed": "blitz",
    "status": "created"
}
```

Accept a challenge

Accept an incoming challenge.

You should receive a `gameStart` event on the incoming events stream.

AUTHORIZATIONS: OAuth2 (`challenge:write,` `bot:play,` `board:play`)

PATH PARAMETERS

`challengeId` required	string Example: `5IrD6Gzz`

Responses

› 200

> 404

POST /api/challenge/{challengeId}/accept

Response samples

200 404

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Decline a challenge

Decline an incoming challenge.

AUTHORIZATIONS: OAuth2 (`challenge:write,` `bot:play,` `board:play`)

PATH PARAMETERS

`challengeId` required	string Example: `5IrD6Gzz`

Responses

> 200

> 404

POST /api/challenge/{challengeId}/decline

Response samples

200 404

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Cancel a challenge

Cancel a challenge you sent, or aborts the game if the challenge was accepted, but the game was not yet played. Note that the ID of a game is the same as the ID of the challenge that created it.

AUTHORIZATIONS: OAuth2 (`challenge:write,` `bot:play,` `board:play`)

PATH PARAMETERS

`challengeId` required	string Example: `5IrD6Gzz`

Responses

> 200

> 404

POST `/api/challenge/{challengeId}/cancel`

Response samples

200 404

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Challenge the AI

Start a game with Lichess AI.

You will be notified on the event stream that a new game has started.

AUTHORIZATIONS: OAuth2 (`challenge:write,` `bot:play,` `board:play`)

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

Parameters of the game

`level` number `[ 1 .. 8 ]`
AI strength

`clock.limit` number `[ 0 .. 10800 ]`

Clock initial time in seconds. If empty, a correspondence game is created.

`clock.increment`

integer `[ 0 .. 60 ]`
Clock increment in seconds. If empty, a correspondence game is created.

`days`

integer `[ 1 .. 15 ]`
Days per move, for correspondence games. Clock settings must be omitted.

`color`

string
Default: `"random"`
Enum: `"random"` `"white"` `"black"`
Which color you get to play

`variant`

string
Default: `"standard"`
Enum: `"standard"` `"chess960"` `"crazyhouse"` `"antichess"` `"atomic"` `"horde"` `"kingOfTheHill"` `"racingKings"` `"threeCheck"`
The variant of the game

`fen`

string
Default: `"rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1"`
Custom initial position (in FEN). Variant must be standard, and the game cannot be rated.

Responses

> 200

> 400

POST `/api/challenge/ai`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "id": "q7ZvsdUF",
    "rated": true,
    "variant": "standard",
    "speed": "blitz",
    "perf": "blitz",
    "createdAt": 1514505150384,
    "lastMoveAt": 1514505592843,
    "status": "draw",
  - "players": {
      + "white": { … },
      + "black": { … }
    },
  - "opening": {
        "eco": "D31",
        "name": "Semi-Slav Defense: Marshall Gambit",
        "ply": 7
    },
    "moves": "d4 d5 c4 c6 Nc3 e6 e4 Nd7 exd5 cxd5 cxd5 exd5 Nxd5 Nb6 Bb5+ E
  - "clock": {
        "initial": 300,
        "increment": 3,
        "totalTime": 420
    }
}
```

Open-ended challenge

Create a challenge that any 2 players can join.

Share the URL of the challenge. the first 2 players to click it will be paired for a game.

The response body also contains `whiteUrl` and `blackUrl`. You can control which color each player gets by giving them these URLs, instead of the main challenge URL.

Open challenges expire after 24h.

To directly pair 2 known players, use this endpoint instead, with the `acceptByToken` parameter.

AUTHORIZATIONS: None

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

Parameters of the game

`clock.limit` number `[ 0 .. 10800 ]`
Clock initial time in seconds. If empty, a correspondence game is created.

`clock.increment` integer `[ 0 .. 60 ]`
Clock increment in seconds. If empty, a correspondence game is created.

`variant` string
Default: `"standard"`
Enum: `"standard"` `"chess960"` `"crazyhouse"` `"antichess"` `"atomic"` `"horde"` `"kingOfTheHill"` `"racingKings"` `"threeCheck"`
The variant of the game

`fen` string
Default: `"rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1"`
Custom initial position (in FEN). Variant must be standard, and the game cannot be rated.

Responses

> 200

> 400

POST /api/challenge/open

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "id": "VU0nyvsW",
    "url": "https://lichess.org/VU0nyvsW",
    "color": "random",
    "direction": "out",
  - "timeControl": {
        "increment": 2,
        "limit": 300,
        "show": "5+2",
        "type": "clock"
    },
  - "variant": {
        "key": "standard",
        "name": "Standard",
        "short": "Std"
    },
  - "challenger": {
        "id": "thibot",
        "name": "thibot",
        "online": true,
        "provisional": false,
        "rating": 1940,
        "title": "BOT"
    },
  - "destUser": {
        "id": "leelachess",
        "name": "LeelaChess",
        "online": true,
        "provisional": true,
        "rating": 2670,
        "title": "BOT"
    },
  - "perf": {
        "icon": ";",
        "name": "Correspondence"
    },
    "rated": true,
    "speed": "blitz",
    "status": "created"
}
```

Start clocks of a game

Start the clocks of a game immediately, even if a player has not yet made a move.

Requires the OAuth tokens of both players with `challenge:write` scope.

If the clocks have already started, the call will have no effect.

AUTHORIZATIONS: OAuth2 (`challenge:write`)

PATH PARAMETERS

`gameId` `required`
string
ID of the game

QUERY PARAMETERS

`token1`
string
OAuth token of a player

`token2` string

OAuth token of the other player

Responses

> 200

POST `/api/challenge/{gameId}/start-clocks`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Arena tournaments

Access Arena tournaments played on Lichess. Official Arena tournaments are maintained by Lichess, but you can create your own Arena tournaments as well.

Get current tournaments

Get recently finished, ongoing, and upcoming tournaments.

This API is used to display the Lichess tournament schedule.

AUTHORIZATIONS: None

Responses

> 200

GET /api/tournament

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
  - "created": [
      + { … }
    ],
  - "started": [
      + { … }
    ],
  - "finished": [
      + { … }
    ]
}
```

Create a new Arena tournament

Create a public or private Arena tournament.

This endpoint mirrors the form on https://lichess.org/tournament/new.

You can create up to 12 public tournaments per day, or 24 private tournaments.

AUTHORIZATIONS: OAuth2 (`tournament:write`)

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

Parameters of the tournament

`name`	string The tournament name. Leave empty to get a random Grandmaster name
`clockTime` required	number `[ 0 .. 60 ]` Enum: `0` `0.25` `0.5` `0.75` `1` `1.5` `2` `3` `4` `5` `6` `7` Clock initial time in minutes
`clockIncrement` required	integer `[ 0 .. 60 ]` Clock increment in seconds
`minutes` required	integer `[ 0 .. 360 ]` How long the tournament lasts, in minutes
`waitMinutes`	integer `[ 0 .. 360 ]` Default: `5` How long to wait before starting the tournament, from now, in minutes
`startDate`	integer Timestamp to start the tournament at a given date and time. Overrides the `waitMinutes` setting
`variant`	string Default: `"standard"`

Enum: `"standard"` `"chess960"` `"crazyhouse"` `"antichess"` `"atomic"` `"horde"` `"kingOfTheHill"` `"racingKings"` `"threeCheck"`

The variant to use in tournament games

`rated`

boolean

Default: `true`

Games are rated and impact players ratings

`position`

string

Custom initial position (in FEN) for all games of the tournament. Must be a legal chess position. Only works with standard chess, not variants.

`berserkable`

boolean

Default: `true`

Whether the players can use berserk

`streakable`

boolean

Default: `true`

After 2 wins, consecutive wins grant 4 points instead of 2.

`hasChat`

boolean

Default: `true`

Whether the players can discuss in a chat

`description`

string

Anything you want to tell players about the tournament

`password`

string

Make the tournament private, and restrict access with a password

`conditions.teamMember.teamId`

string

Restrict entry to members of a team.

The teamId is the last part of a team URL, e.g. https://lichess.org/team/coders has teamId = `coders`.

Leave empty to let everyone join the tournament.

`conditions.minRating.rating`

integer

Minimum rating to join. Leave empty to let everyone join the tournament.

`conditions.maxRating.rating`

integer

Maximum rating to join. Based on best rating reached in the last 7 days. Leave empty to let everyone join the tournament.

`conditions.nbRatedGame.nb`

integer

Minimum number of rated games required to join.

Responses

> 200

> 400

POST `/api/tournament`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "id": "2VoKhb8g",
    "createdBy": "lichess",
    "system": "arena",
    "minutes": 57,
  - "clock": {
        "limit": 0,
        "increment": 0
    },
    "rated": true,
       "fullName": "Petrov's Defence Blitz Arena",
       "nbPlayers": 156,
     - "variant": {
           "key": "standard",
           "short": "Std",
           "name": "Standard"
       },
       "startsAt": 1599886846000,
       "finishesAt": 1599890266000,
       "status": 30,
     - "perf": {
           "icon": ")",
           "key": "blitz",
           "name": "Blitz",
           "position": 1
       },
       "secondsToStart": 0,
       "hasMaxRating": false,
       "private": false,
     - "position": {
           "eco": "C43",
           "name": "Petrov's Defence: Steinitz Attack",
           "wikiPath": "Petrov's_Defence#3.d4",
           "fen": "string"
       },
     - "schedule": {
           "freq": "hourly",
           "speed": "blitz"
       },
     - "winner": {
           "name": "Chess-Network",
           "title": "NM",
           "patron": true,
           "id": "chess-network"
       }
   }
```

Get info about a tournament

Get detailed info about recently finished, current, or upcoming tournament's duels, player standings, and other info.

AUTHORIZATIONS: None

PATH PARAMETERS

`id` `required`	string The tournament ID.

QUERY PARAMETERS

`page`	number `[ 1 .. 200 ]` Default: `1` Example: `page=1` Specify which page of player standings to view.

Responses

> 200

GET `/api/tournament/{id}`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
      "id": "QITRjufu",
      "fullName": "U1700 SuperBlitz Arena",
    - "clock": {
          "increment": 0,
          "limit": 180
      },
      "minutes": 57,
      "createdBy": "lichess",
      "system": "arena",
      "secondsToStart": 0,
      "secondsToFinish": 36000,
      "isFinished": true,
      "isRecentlyFinished": true,
      "pairingsClosed": true,
      "startsAt": "2018-04-04T01:00:00.000Z",
      "nbPlayers": 154,
    - "perf": {
          "icon": ")",
          "key": "blitz",
          "name": "Blitz",
          "position": 1
      },
    - "schedule": {
          "freq": "hourly",
          "speed": "superblitz"
      },
    - "variant": {
          "key": "standard",
          "name": "Standard",
          "short": "Std"
      },
    - "duels": [
        + { … }
      ],
    - "standings": {
          "page": 1,
        + "players": [ … ]
      },
    - "featured": {
          "id": "khe72Fer",
          "fen": "rn1qkb1r/pQ3ppp/2b2n2/8/5P2/4P3/PP4PP/RNB1KBNR",
          "color": "black",
          "lastMove": "d7c6",
        + "white": { … },
        + "black": { … }
      },
    - "podium": [
        + { … }
      ],
    - "stats": {
          "games": 454,
          "moves": 27542,
          "whiteWins": 236,
          "blackWins": 207,
          "draws": 11,
          "berserks": 0,
          "averageRating": 1320
      }
  }
```

Export games of an Arena tournament

Download games of a tournament in PGN or ndjson format.

Games are sorted by reverse chronological order (most recent first)

AUTHORIZATIONS: None

PATH PARAMETERS

`id` required	string The tournament ID.

QUERY PARAMETERS

`moves`

boolean

Default: `true`

Include the PGN moves.

`pgnInJson`

boolean

Default: `false`

Include the full PGN within the JSON response, in a `pgn` field.

`tags`

boolean

Default: `true`

Include the PGN tags.

`clocks`

boolean

Default: `false`

Include clock comments in the PGN moves, when available.

Example: `2. exd5 { [%clk 1:01:27] } e5 { [%clk 1:01:28] }`

`evals`

boolean

Default: `false`

Include analysis evaluation comments in the PGN, when available.

Example: `12. Bxf6 { [%eval 0.23] } a3 { [%eval -1.09] }`

`opening`

boolean

Default: `false`

Include the opening name.

Example: `[Opening "King's Gambit Accepted, King's Knight Gambit"]`

Responses

> 200

GET `/api/tournament/{id}/games`

Response samples

200

Content type

application/x-chess-pgn

```
No sample
```

Get results of an arena tournament

Players of an arena tournament, with their score and performance, sorted by rank (best first).

Players are streamed as ndjson, i.e. one JSON object per line.

If called on an ongoing tournament, results can be inconsistent due to ranking changes while the players are being streamed. Use on finished tournaments for guaranteed consistency.

AUTHORIZATIONS: None

PATH PARAMETERS

`id` required	string The tournament ID.

QUERY PARAMETERS

`nb`	integer `>= 1` Max number of players to fetch

Responses

> 200

GET /api/tournament/{id}/results

Response samples

200

Content type

application/x-ndjson

Copy Expand all Collapse all

```
{
    "rank": 4,
    "score": 389,
    "rating": 2618,
    "username": "opperwezen",
    "title": "IM",
    "performance": 2423,
    "team": "coders"
}
```

Get team standing of a team battle

Teams of a team battle tournament, with top players, sorted by rank (best first).

AUTHORIZATIONS: None

PATH PARAMETERS

id required	string The tournament ID.

Responses

> 200

GET /api/tournament/{id}/teams

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "id": "CdPg1ey4",
  - "teams": [
      + { … }
    ]
}
```

Get tournaments created by a user

Get all tournaments created by a given user.

Tournaments are sorted by reverse chronological order of start date (last starting first).

Tournaments are streamed as ndjson, i.e. one JSON object per line.

AUTHORIZATIONS: None

QUERY PARAMETERS

nb integer >= 1

Max number of tournaments to fetch

Responses

> 200

GET /api/user/{username}/tournament/created

Response samples

200

Content type

application/x-ndjson

Copy Expand all Collapse all

```
{
    "id": "2VoKhb8g",
    "createdBy": "lichess",
    "system": "arena",
    "minutes": 57,
  - "clock": {
        "limit": 0,
        "increment": 0
    },
    "rated": true,
    "fullName": "Petrov's Defence Blitz Arena",
    "nbPlayers": 156,
  - "variant": {
        "key": "standard",
        "short": "Std",
        "name": "Standard"
    },
    "startsAt": 1599886846000,
        "finishesAt": 1599890266000,
        "status": 30,
      - "perf": {
            "icon": ")",
            "key": "blitz",
            "name": "Blitz",
            "position": 1
        },
        "secondsToStart": 0,
        "hasMaxRating": false,
        "private": false,
      - "position": {
            "eco": "C43",
            "name": "Petrov's Defence: Steinitz Attack",
            "wikiPath": "Petrov's_Defence#3.d4",
            "fen": "string"
        },
      - "schedule": {
            "freq": "hourly",
            "speed": "blitz"
        },
      - "winner": {
            "name": "Chess-Network",
            "title": "NM",
            "patron": true,
            "id": "chess-network"
        }
    }
```

Get team arena tournaments

Get all arena tournaments relevant to a team.

Tournaments are sorted by reverse chronological order of start date (last starting first).

Tournaments are streamed as ndjson, i.e. one JSON object per line.

AUTHORIZATIONS: None

QUERY PARAMETERS

`max`	integer `>= 1` Default: `100` How many tournaments to download.

Responses

› 200

GET `/api/team/{teamId}/arena`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
```

```
  - {
        "id": "2VoKhb8g",
        "createdBy": "lichess",
        "system": "arena",
        "minutes": 57,
      + "clock": { … },
        "rated": true,
        "fullName": "Petrov's Defence Blitz Arena",
        "nbPlayers": 156,
      + "variant": { … },
        "startsAt": 1599886846000,
        "finishesAt": 1599890266000,
        "status": 30,
      + "perf": { … },
        "secondsToStart": 0,
        "hasMaxRating": false,
        "private": false,
      + "position": { … },
      + "schedule": { … },
      + "winner": { … }
    }
]
```

Swiss tournaments

Access Swiss tournaments played on Lichess. Read more about Swiss tournaments..

Create a new Swiss tournament

Create a Swiss tournament for your team.

This endpoint mirrors the Swiss tournament form from your team pagee.

You can create up to 12 tournaments per day.

AUTHORIZATIONS: OAuth2 (`tournament:write`)

QUERY PARAMETERS

`teamId`
string
ID of the team

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

Parameters of the tournament

`name`
string
The tournament name. Leave empty to get a random Grandmaster name

`clock.limit`
required
number `[ 0 .. 3600 ]`
Clock initial time in seconds

`clock.increment`
required
integer `[ 0 .. 600 ]`
Clock increment in seconds

`nbRounds`
required
integer `[ 3 .. 100 ]`
Default: `8`
Maximum number of rounds to play

`startsAt`
integer
Timestamp in milliseconds to start the tournament at a given date and time. By default, it starts 10 minutes after creation.

`roundInterval`
integer `[ 0 .. 86400 ]`
Default: `"auto"`
How long to wait between each round, in seconds.

Set to 99999999 to manually schedule each round from the tournament UI.

`variant`
string
Default: `"standard"`
Enum: `"standard"` `"chess960"` `"crazyhouse"` `"antichess"` `"atomic"` `"horde"` `"kingOfTheHill"` `"racingKings"` `"threeCheck"`
The variant to use in tournament games

`description`
string

Anything you want to tell players about the tournament

`rated`

boolean

Default: `true`

Games are rated and impact players ratings

`chatFor`

number

Default: `20`

Who can read and write in the chat.

- 0 = No-one
- 10 = Only team leaders
- 20 = Only team members
- 30 = All Lichess players

Responses

› 200

› 400

POST `/api/swiss/new/{teamId}`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "rated": true,
    - "clock": {
          "increment": 0,
          "limit": 300
      },
      "createdBy": "thibault",
    - "greatPlayer": {
          "name": "Wang",
          "url": "https://wikipedia.org/wiki/Wang_Hao_(chess_player)"
      },
      "id": "ZmKWCOye",
      "name": "Wang",
      "nbOngoing": 0,
      "nbPlayers": 0,
      "nbRounds": 2,
    - "nextRound": {
          "at": "2020-05-11T12:23:18.233-06:00",
          "in": 600
      },
    - "quote": {
          "author": "Emanuel Lasker",
          "text": "By what right does White, in an absolutely even position,
      },
      "round": 0,
      "startsAt": "2020-05-11T12:23:18.233-06:00",
      "status": "created",
      "variant": "standard"
  }
```

Export TRF of a Swiss tournament

Download a tournament in the Tournament Report File format, the FIDE standard.

Documentation: https://www.fide.com/FIDE/handbook/C04Annex2_TRF16.pdf

Example: https://lichess.org/swiss/j8rtJ5GL.trf

AUTHORIZATIONS: None

PATH PARAMETERS

id required	string The tournament ID.

Responses

— 200

GET /swiss/{id}.trf

Response samples

200

Content type
text/plain

Copy

```
null
```

Export games of a Swiss tournament

Download games of a swiss tournament in PGN or ndjson format.

Games are sorted by reverse chronological order (last round first)

AUTHORIZATIONS: None

PATH PARAMETERS

`id` required

string

The tournament ID.

QUERY PARAMETERS

`moves`

boolean

Default: `true`

Include the PGN moves.

`pgnInJson`

boolean

Default: `false`

Include the full PGN within the JSON response, in a `pgn` field.

`tags`

boolean

Default: `true`

Include the PGN tags.

`clocks`

boolean

Default: `false`

Include clock comments in the PGN moves, when available.

Example: `2. exd5 { [%clk 1:01:27] } e5 { [%clk 1:01:28] }`

`evals`

boolean

Default: `false`

Include analysis evaluation comments in the PGN, when available.

Example: `12. Bxf6 { [%eval 0.23] } a3 { [%eval -1.09] }`

`opening`

boolean

Default: `false`
Include the opening name.

Example: `[Opening "King's Gambit Accepted, King's Knight Gambit"]`

Responses

> 200

GET /api/swiss/{id}/games

Response samples

200

Content type
application/x-chess-pgn

```
No sample
```

Get results of a swiss tournament

Players of a swiss tournament, with their score and performance, sorted by rank (best first).

Players are streamed as ndjson, i.e. one JSON object per line.

If called on an ongoing tournament, results can be inconsistent due to ranking changes while the players are being streamed. Use on finished tournaments for guaranteed consistency.

AUTHORIZATIONS: None

PATH PARAMETERS

`id` `required`	string The tournament ID.

QUERY PARAMETERS

`nb`	integer `>= 1` Max number of players to fetch

Responses

› **200**

GET `/api/swiss/{id}/results`

Response samples

200

Content type

application/x-ndjson

Copy Expand all Collapse all

```
{
    "rank": 4,
    "score": 24,
    "tieBreak": 77,
    "rating": 2618,
    "username": "opperwezen",
    "title": "IM",
    "performance": 2423
}
```

Get team swiss tournaments

Get all swiss tournaments of a team.

Tournaments are sorted by reverse chronological order of start date (last starting first).

Tournaments are streamed as ndjson, i.e. one JSON object per line.

AUTHORIZATIONS: None

PATH PARAMETERS

`teamId` required	string Example: `coders`

QUERY PARAMETERS

`max`	integer `>= 1` Default: `100` How many tournaments to download.

Responses

> 200

```
GET /api/team/{teamId}/swiss
```

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
[
 - {
      "rated": true,
    + "clock": { … },
      "createdBy": "thibault",
    + "greatPlayer": { … },
      "id": "ZmKWCOye",
      "name": "Wang",
      "nbOngoing": 0,
      "nbPlayers": 0,
      "nbRounds": 2,
    + "nextRound": { … },
    + "quote": { … },
      "round": 0,
      "startsAt": "2020-05-11T12:23:18.233-06:00",
      "status": "created",
      "variant": "standard"
   }
]
```

Simuls

Access simuls played on Lichess. https://lichess.org/simul

Get current simuls

Get recently finished, ongoing, and upcoming simuls.

AUTHORIZATIONS: None

Responses

› 200

GET /api/simul

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
 - {
       "fullName": "GM ChessWeeb simul",
     + "host": { … },
       "id": "pDGbxhUe",
       "isCreated": false,
       "isFinished": true,
       "isRunning": false,
       "name": "GM ChessWeeb",
       "nbApplicants": 0,
       "nbPairings": 24,
       "text": "",
     + "variants": [ … ]
   }
]
```

Studies

Access Lichess studies. https://lichess.org/study

Export one study chapter

Download one study chapter in PGN format.

AUTHORIZATIONS: SameOrigin

PATH PARAMETERS

`studyId` required
string
The study ID (8 characters).

`chapterId` required
string
The chapter ID (8 characters).

QUERY PARAMETERS

`clocks`
boolean
Default: `true`
Include clock comments in the PGN moves, when available.

Example: `2. exd5 { [%clk 1:01:27] } e5 { [%clk 1:01:28] }`

`comments`
boolean
Default: `true`
Include analysis and annotator comments in the PGN moves, when available.

Example: `12. Bxf6 { [%eval 0.23] } a3 { White is in a pickle. }`

`variations`

boolean

Default: `true`

Include non-mainline moves, when available.

Example: `4. d4 Bb4+ (4... Nc6 5. Nf3 Bb4+ 6. Bd2 (6. Nbd2 O-O 7. O-O) 6... Bd6) 5. Nd2`

Responses

› 200

GET `/study/{studyId}/{chapterId}.pgn`

Export all chapters

Download all chapters of a study in PGN format.

AUTHORIZATIONS: SameOrigin

PATH PARAMETERS

`studyId` `required`

string

The study ID (8 characters).

QUERY PARAMETERS

`clocks`

boolean

Default: `true`

Include clock comments in the PGN moves, when available.

Example: `2. exd5 { [%clk 1:01:27] } e5 { [%clk 1:01:28] }` `comments` boolean

Default: `true`

Include analysis and annotator comments in the PGN moves, when available.

Example: `12. Bxf6 { [%eval 0.23] } a3 { White is in a pickle. }`

`variations` boolean

Default: `true`

Include non-mainline moves, when available.

Example: `4. d4 Bb4+ (4... Nc6 5. Nf3 Bb4+ 6. Bd2 (6. Nbd2 O-O 7. O-O) 6... Bd6) 5. Nd2`

Responses

> 200

GET `/study/{studyId}.pgn`

Messaging

Private messages with other players. https://lichess.org/inbox

Send a private message

Send a private message to another player.

AUTHORIZATIONS: OAuth2 (`msg:write`) **AND** SameOrigin

PATH PARAMETERS

`username` `required` string

Example: `someplayer`

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`text` `required` string

Responses

> 200

> 400

POST `/inbox/{username}`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Broadcasts

Relay chess events on Lichess. Official broadcasts are maintained by Lichess, but you can create your own broadcasts to cover any live game or chess event. You will need to publish PGN on a public URL so that Lichess can pull updates from it. Alternatively, you can push PGN updates to Lichess using this API.

Get official broadcasts

Get all incoming, ongoing, and finished official broadcasts. The broadcasts are sorted by start date, most recent first.

Broadcasts are streamed as ndjson, i.e. one JSON object per line.

QUERY PARAMETERS

`nb`	integer `>= 1` Default: `20` Max number of broadcasts to fetch

Responses

> 200

GET `/api/broadcast`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
  - {
        "description": "An 11 round classical tournament featuring the 9 hi
        "url": "https://lichess.org/broadcast/sinquefield-cup-round-10/wje0
        "id": "wje0786Q",
        "markup": "<p>Starts August 16th at 18:00 UTC</p>\n<table class=\"s
        "name": "Sinquefield Cup - Round 10",
        "ownerId": "broadcaster",
        "startedAt": 1566928867494,
        "startsAt": 1566928800000,
      + "sync": { … }
    }
]
```

Create a broadcast

Create a new broadcast to relay external games. This endpoint accepts the same form data as the web form.

AUTHORIZATIONS: OAuth2 (`study:write`)

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`name` required

string

Name of the broadcast. Length must be between 3 and 80 characters.

Example: `Sinquefield Cup - Round 10`

`description` required

string

Short description of the broadcast. Length must be between 3 and 400 characters.

Example: `An 11 round classical tournament featuring the 9 highest rated players in the world. Including Carlsen, Caruana, Ding, Aronian, Nakamura and more.`

`syncUrl`

string

URL that Lichess will poll to get updates about the games. It must be publicly accessible from the Internet.

Example: `http://myserver.org/myevent/round-10/games.pgn`

If the syncUrl is missing, then the broadcast needs to be fed by pushing PGN to it.

`markdown`

string

Optional long description of the broadcast. Markdown is supported. Length must be less than 20,000 characters.

`credit`

string

Optional short text to give credit to the source provider.

`startsAt`

integer

Timestamp in milliseconds of broadcast start. Leave empty to manually start the broadcast.

Example: `1356998400070`

`official`

boolean

For Lichess internal usage only. You are not allowed to use this flag. If you do it, we will have to call the police.

`throttle`

integer

For Lichess internal usage only. You are not allowed to use this flag. If you do it, chess24 will display ads on your broadcast.

Responses

› 200

› 400

POST /broadcast/new

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

```
{
    "description": "An 11 round classical tournament featuring the 9 highes
    "url": "https://lichess.org/broadcast/sinquefield-cup-round-10/wje0786Q
    "id": "wje0786Q",
    "markup": "<p>Starts August 16th at 18:00 UTC</p>\n<table class=\"slist
    "name": "Sinquefield Cup - Round 10",
    "ownerId": "broadcaster",
    "startedAt": 1566928867494,
    "startsAt": 1566928800000,
  - "sync": {
        "log": [ ],
        "ongoing": false,
        "url": "http://0.0.0.0/sinquefield2019/10/games.pgn"
    }
}
```

Get your broadcast

Get information about a broadcast that you created. You will need it if you want to update that broadcast.

AUTHORIZATIONS: OAuth2 (`study:read`) **AND** SameOrigin

PATH PARAMETERS

`slug`
`required`

string

Default: `"-"`

The broadcast slug. Only used for SEO, the slug can be safely replaced by `-`. Only the `broadcastId` is actually used.

`broadcastId`
`required`

string

The broadcast ID (8 characters).

Responses

› 200

GET `/broadcast/{slug}/{broadcastId}`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
  - {
        "description": "An 11 round classical tournament featuring the 9 hi
        "url": "https://lichess.org/broadcast/sinquefield-cup-round-10/wje0
        "id": "wje0786Q",
        "markup": "<p>Starts August 16th at 18:00 UTC</p>\n<table class=\"s
        "name": "Sinquefield Cup - Round 10",
        "ownerId": "broadcaster",
        "startedAt": 1566928867494,
        "startsAt": 1566928800000,
      + "sync": { … }
    }
]
```

Update your broadcast

Update information about a broadcast that you created. This endpoint accepts the same form data as the web form. All fields must be populated with data. Missing fields will override the broadcast with empty data. For instance, if you omit `startDate`, then any pre-existing start date will be removed.

AUTHORIZATIONS: OAuth2 (`study:write`)

PATH PARAMETERS

`slug` required

string

Default: `"-"`

The broadcast slug. Only used for SEO, the slug can be safely replaced by `-`. Only the `broadcastId` is actually used.

`broadcastId` required

string

The broadcast ID (8 characters).

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`name` required

string

Name of the broadcast. Length must be between 3 and 80 characters.

Example: `Sinquefield Cup - Round 10`

`description`
required

string

Short description of the broadcast. Length must be between 3 and 400 characters.

Example: `An 11 round classical tournament featuring the 9 highest rated players in the world. Including Carlsen, Caruana, Ding, Aronian, Nakamura and more.`

`syncUrl`
required

string

URL that Lichess will poll to get updates about the games. It must be publicly accessible from the Internet.

Example: `http://myserver.org/myevent/round-10/games.pgn`

`markdown`

string

Optional long description of the broadcast. Markdown is supported. Length must be less than 20,000 characters.

`credit`

string

Optional short text to give credit to the source provider.

`startsAt`

integer

Timestamp in milliseconds of broadcast start. Leave empty to manually start the broadcast.

Example: `1356998400070`

`official`

boolean

For Lichess internal usage only. You are not allowed to use this flag. If you do it, we will have to call the police.

`throttle`

integer

For Lichess internal usage only. You are not allowed to use this flag. If you do it, chess24 will display ads on your broadcast.

Responses

› 200

› 400

POST /broadcast/{slug}/{broadcastId}

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

```
{
    "description": "An 11 round classical tournament featuring the 9 highes
    "url": "https://lichess.org/broadcast/sinquefield-cup-round-10/wje0786Q
    "id": "wje0786Q",
    "markup": "<p>Starts August 16th at 18:00 UTC</p>\n<table class=\"slist
    "name": "Sinquefield Cup - Round 10",
    "ownerId": "broadcaster",
    "startedAt": 1566928867494,
    "startsAt": 1566928800000,
  - "sync": {
        "log": [ ],
        "ongoing": false,
        "url": "http://0.0.0.0/sinquefield2019/10/games.pgn"
    }
}
```

Push PGN to your broadcast

Update your broadcast with new PGN. Only for broadcast without a source URL.

AUTHORIZATIONS: OAuth2 (`study:write`) **AND** SameOrigin

PATH PARAMETERS

`slug` `required`	string Default: `"-"` The broadcast slug. Only used for SEO, the slug can be safely replaced by `-`. Only the `broadcastId` is actually used.
`broadcastId` `required`	string The broadcast ID (8 characters).

REQUEST BODY SCHEMA: text/plain

The PGN. It can contain up to 64 games, separated by a double new line.

string

Responses

› 200

POST `/broadcast/{slug}/{broadcastId}/push`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
  - {
        "ok": true
    }
]
```

OAuth

Requests taking advantage of OAuth authentication. https://lichess.org/account/oauth/app

Get your puzzle activity

Download your puzzle activity in ndjson format.

Puzzle activity is sorted by reverse chronological order (most recent first)

We recommend streaming the response, for it can be very long.

AUTHORIZATIONS: OAuth2 (`puzzle:read`)

QUERY PARAMETERS

`max`	integer `>= 1` Default: `null` How many entries to download. Leave empty to download all activity.

Responses

› 200

GET `/api/user/puzzle-activity`

Response samples

200

Content type
application/x-ndjson

Copy Expand all Collapse all

```
{
    "id": 70232,
    "date": 1514505150384,
    "rating": 1982,
    "ratingDiff": 5,
    "puzzleRating": 1877
}
```

Get my profile

Public informations about the logged in user.

AUTHORIZATIONS: OAuth2

Responses

> 200

GET /api/account

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "id": "georges",
    "username": "Georges",
    "online": true,
  - "perfs": {
      + "chess960": { … },
      + "atomic": { … },
      + "racingKings": { … },
      + "ultraBullet": { … },
      + "blitz": { … },
      + "kingOfTheHill": { … },
      + "bullet": { … },
      + "correspondence": { … },
      + "horde": { … },
      + "puzzle": { … },
      + "classical": { … },
      + "rapid": { … }
    },
    "createdAt": 1290415680000,
    "disabled": false,
    "engine": false,
    "booster": false,
  - "profile": {
        "country": "EC",
        "location": "string",
        "bio": "Free bugs!",
        "firstName": "Thibault",
        "lastName": "Duplessis",
        "fideRating": 1500,
        "uscfRating": 1500,
        "ecfRating": 1500,
        "links": "github.com/ornicar\r\ntwitter.com/ornicar"
    },
        "seenAt": 1522636452014,
        "patron": true,
      - "playTime": {
            "total": 3296897,
            "tv": 12134
        },
        "language": "en-GB",
        "title": "NM",
        "url": "https://lichess.org/@/georges",
        "playing": "https://lichess.org/yqfLYJ5E/black",
        "nbFollowing": 299,
        "nbFollowers": 2735,
        "completionRate": 97,
      - "count": {
            "all": 9265,
            "rated": 7157,
            "ai": 531,
            "draw": 340,
            "drawH": 331,
            "loss": 4480,
            "lossH": 4207,
            "win": 4440,
            "winH": 4378,
            "bookmark": 71,
            "playing": 6,
            "import": 66,
            "me": 0
        },
        "streaming": false,
        "followable": true,
        "following": false,
        "blocking": false,
        "followsYou": false
    }
```

Get my email address

Read the email address of the logged in user.

AUTHORIZATIONS: OAuth2 (`email:read`)

Responses

> 200

GET `/api/account/email`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "email": "abathur@mail.org"
}
```

Get my preferences

Read the preferences of the logged in user.

- https://lichess.org/account/preferences/game-display
- https://github.com/ornicar/lila/blob/master/modules/pref/src/main/Pref.scala

AUTHORIZATIONS: OAuth2 (`preference:read`)

Responses

> 200

GET `/api/account/preferences`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    - "prefs": {
          "dark": true,
          "transp": false,
          "bgImg": "http://example.com",
          "is3d": false,
          "theme": "blue",
          "pieceSet": "cburnett",
          "theme3d": "Black-White-Aluminium",
          "pieceSet3d": "Basic",
          "soundSet": "silent",
          "blindfold": 0,
          "autoQueen": 2,
          "autoThreefold": 2,
          "takeback": 3,
          "moretime": 3,
          "clockTenths": 1,
          "clockBar": true,
          "clockSound": true,
          "premove": true,
          "animation": 2,
          "captured": true,
          "follow": true,
          "highlight": true,
          "destination": true,
          "coords": 2,
          "replay": 2,
          "challenge": 4,
          "message": 3,
          "coordColor": 2,
          "submitMove": 4,
          "confirmResign": 1,
          "insightShare": 1,
          "keyboardMove": 0,
          "zen": 0,
          "moveEvent": 2,
          "rookCastle": 1
      }
  }
```

Get my kid mode status

Read the kid mode status of the logged in user.

- https://lichess.org/account/kid

AUTHORIZATIONS: OAuth2 (`preference:read`)

Responses

› 200

GET `/api/account/kid`

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "kid": false
}
```

Set my kid mode status

Set the kid mode status of the logged in user.

- https://lichess.org/account/kid

AUTHORIZATIONS: OAuth2 (`preference:write`)

QUERY PARAMETERS

`v` `required`	boolean Example: `v=true` Kid mode status

Responses

› 200

POST `/api/account/kid`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Export games of a user

Download all games of any user in PGN or ndjson format.

Games are sorted by reverse chronological order (most recent first)

We recommend streaming the response, for it can be very long. https://lichess.org/@/german11 for instance has more than 320,000 games.

The game stream is throttled, depending on who is making the request:

- Anonymous request: 20 games per second
- OAuth2 authenticated request: 30 games per second
- Authenticated, downloading your own games: 60 games per second

AUTHORIZATIONS: (None) **OR** (OAuth2)

PATH PARAMETERS

`username` required

string

The user name.

QUERY PARAMETERS

`since`

integer `>= 1356998400070`

Default: `"Account creation date"`

Download games played since this timestamp.

`until`

integer `>= 1356998400070`

Default: `"Now"`

Download games played until this timestamp.

`max`

integer `>= 1`

Default: `null`

How many games to download. Leave empty to download all games.

`vs`

string

[Filter] Only games played against this opponent

`rated`

boolean

Default: `null`

[Filter] Only rated (`true`) or casual (`false`) games

`perfType`

string

Default: `null`
Enum: `"ultraBullet"` `"bullet"` `"blitz"` `"rapid"` `"classical"` `"correspon`
`"atomic"` `"horde"` `"kingOfTheHill"` `"racingKings"` `"threeCheck"`
[Filter] Only games in these speeds or variants.

Multiple perf types can be specified, separated by a comma.

Example: blitz,rapid,classical

`color`

string
Default: `null`
Enum: `"white"` `"black"`
[Filter] Only games played as this color.

`analysed`

boolean
Default: `null`
[Filter] Only games with or without a computer analysis available

---

`ongoing`

boolean
Default: `false`
[Filter] Also include ongoing games

`moves`

boolean
Default: `true`
Include the PGN moves.

`pgnInJson`

boolean
Default: `false`
Include the full PGN within the JSON response, in a `pgn` field. The response type must be
header.

`tags`

boolean
Default: `true`
Include the PGN tags.

`clocks`

boolean
Default: `false`
Include clock comments in the PGN moves, when available.

Example: `2. exd5 { [%clk 1:01:27] } e5 { [%clk 1:01:28] }`

`evals`

boolean
Default: `false`
Include analysis evaluation comments in the PGN, when available.

Example: `12. Bxf6 { [%eval 0.23] } a3 { [%eval -1.09] }`

`opening` boolean

Default: `false`

Include the opening name.

Example: `[Opening "King's Gambit Accepted, King's Knight Gambit"]`

`players` string

URL of a text file containing real names and ratings, to replace Lichess usernames and rat

https://gist.githubusercontent.com/ornicar/6bfa91eb61a2dcae7bcd14cce1b2a4eb/raw/7

Responses

› 200

GET /api/games/user/{username}

Response samples

200

Content type

application/x-chess-pgn

`No sample`

Get ongoing games

Get the ongoing games of the current user. Real-time and correspondence games are included. The most urgent games are listed first.

AUTHORIZATIONS: OAuth2

QUERY PARAMETERS

`nb`	integer `[ 1 .. 50 ]` Default: `9` Max number of games to fetch

Responses

> 200

GET `/api/account/playing`

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
  - "nowPlaying": [
      + { … }
    ]
}
```

Import one game

Import a game from PGN. See https://lichess.org/paste.

Rate limiting: 200 games per hour for OAuth requests, 100 games per hour for anonymous requests.

To broadcast ongoing games, consider pushing to a broadcast instead.

AUTHORIZATIONS: (None) **OR** (OAuth2)

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

A single game to import

`pgn`	string The PGN. It can contain only one game. Most standard tags are supported.

Responses

› **200**

POST `/api/import`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "id": "R6iLjwz5",
    "url": "https://lichess.org/R6iLjwz5"
}
```

Create a broadcast

Create a new broadcast to relay external games. This endpoint accepts the same form data as the web form.

AUTHORIZATIONS: OAuth2 (`study:write`)

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`name`
`required`

string

Name of the broadcast. Length must be between 3 and 80 characters.

Example: `Sinquefield Cup - Round 10`

`description`
`required`

string

Short description of the broadcast. Length must be between 3 and 400 characters.

Example: `An 11 round classical tournament featuring the 9 highest rated players in the world. Including Carlsen, Caruana, Ding, Aronian, Nakamura and more.`

`syncUrl`

string

URL that Lichess will poll to get updates about the games. It must be publicly accessible from the Internet.

Example: `http://myserver.org/myevent/round-10/games.pgn`

If the syncUrl is missing, then the broadcast needs to be fed by pushing PGN to it.

`markdown`

string

Optional long description of the broadcast. Markdown is supported. Length must be less than 20,000 characters.

`credit`

string

Optional short text to give credit to the source provider.

`startsAt`

integer

Timestamp in milliseconds of broadcast start. Leave empty to manually start the broadcast.

Example: `1356998400070`

`official`

boolean

For Lichess internal usage only. You are not allowed to use this flag. If you do it, we will have to call the police.

`throttle`

integer

For Lichess internal usage only. You are not allowed to use this flag. If you do it, chess24 will display ads on your broadcast.

Responses

› 200

› 400

POST `/broadcast/new`

Response samples

200 400

Content type

application/json

Copy Expand all Collapse all

```
{
    "description": "An 11 round classical tournament featuring the 9 highes
    "url": "https://lichess.org/broadcast/sinquefield-cup-round-10/wje0786Q
    "id": "wje0786Q",
    "markup": "<p>Starts August 16th at 18:00 UTC</p>\n<table class=\"slist
    "name": "Sinquefield Cup - Round 10",
    "ownerId": "broadcaster",
    "startedAt": 1566928867494,
    "startsAt": 1566928800000,
      - "sync": {
            "log": [ ],
            "ongoing": false,
            "url": "http://0.0.0.0/sinquefield2019/10/games.pgn"
        }
    }
```

Get your broadcast

Get information about a broadcast that you created. You will need it if you want to update that broadcast.

AUTHORIZATIONS: OAuth2 (`study:read`) **AND** SameOrigin

PATH PARAMETERS

`slug` required

string

Default: `"-"`

The broadcast slug. Only used for SEO, the slug can be safely replaced by `-`. Only the `broadcastId` is actually used.

`broadcastId` required

string

The broadcast ID (8 characters).

Responses

> **200**

```
GET    /broadcast/{slug}/{broadcastId}
```

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[
  - {
        "description": "An 11 round classical tournament featuring the 9 hi
        "url": "https://lichess.org/broadcast/sinquefield-cup-round-10/wje0
        "id": "wje0786Q",
        "markup": "<p>Starts August 16th at 18:00 UTC</p>\n<table class=\"s
        "name": "Sinquefield Cup - Round 10",
        "ownerId": "broadcaster",
        "startedAt": 1566928867494,
        "startsAt": 1566928800000,
      + "sync": { … }
    }
]
```

Update your broadcast

Update information about a broadcast that you created. This endpoint accepts the same form data as the web form. All fields must be populated with data. Missing fields will override the broadcast with empty data. For instance, if you omit `startDate`, then any pre-existing start date will be removed.

AUTHORIZATIONS: OAuth2 (`study:write`)

PATH PARAMETERS

`slug` required	string Default: `"-"` The broadcast slug. Only used for SEO, the slug can be safely replaced by

`-`. Only the `broadcastId` is actually used.

`broadcastId` required

string

The broadcast ID (8 characters).

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`name` required

string

Name of the broadcast. Length must be between 3 and 80 characters.

Example: `Sinquefield Cup - Round 10`

`description` required

string

Short description of the broadcast. Length must be between 3 and 400 characters.

Example: `An 11 round classical tournament featuring the 9 highest rated players in the world. Including Carlsen, Caruana, Ding, Aronian, Nakamura and more.`

`syncUrl` required

string

URL that Lichess will poll to get updates about the games. It must be publicly accessible from the Internet.

Example: `http://myserver.org/myevent/round-10/games.pgn`

`markdown`

string

Optional long description of the broadcast. Markdown is supported. Length must be less than 20,000 characters.

`credit`

string

Optional short text to give credit to the source provider.

`startsAt`

integer

Timestamp in milliseconds of broadcast start. Leave empty to manually start the broadcast.

Example: `1356998400070`

`official`

boolean

For Lichess internal usage only. You are not allowed to use this flag. If you do it, we will have to call the police.

`throttle`

integer

For Lichess internal usage only. You are not allowed to use this flag. If you do it, chess24 will display ads on your broadcast.

Responses

> 200

> 400

POST /broadcast/{slug}/{broadcastId}

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

```
{
    "description": "An 11 round classical tournament featuring the 9 highes
    "url": "https://lichess.org/broadcast/sinquefield-cup-round-10/wje0786Q
    "id": "wje0786Q",
    "markup": "<p>Starts August 16th at 18:00 UTC</p>\n<table class=\"slist
    "name": "Sinquefield Cup - Round 10",
    "ownerId": "broadcaster",
    "startedAt": 1566928867494,
    "startsAt": 1566928800000,
  - "sync": {
        "log": [ ],
        "ongoing": false,
        "url": "http://0.0.0.0/sinquefield2019/10/games.pgn"
    }
}
```

Push PGN to your broadcast

Update your broadcast with new PGN. Only for broadcast without a source URL.

AUTHORIZATIONS: OAuth2 (`study:write`) **AND** SameOrigin

PATH PARAMETERS

`slug` `required`	string Default: `"-"` The broadcast slug. Only used for SEO, the slug can be safely replaced by `-`. Only the `broadcastId` is actually used.
`broadcastId` `required`	string The broadcast ID (8 characters).

REQUEST BODY SCHEMA: text/plain

The PGN. It can contain up to 64 games, separated by a double new line.

string

Responses

› 200

POST `/broadcast/{slug}/{broadcastId}/push`

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
[
  - {
        "ok": true
    }
]
```

Join a team

Join a team. If the team join policy requires a confirmation, and the team owner is not the oAuth app owner, and the `message` field is not set, then the call fails with `403 Forbidden`.

AUTHORIZATIONS: OAuth2 (`team:write`) **AND** SameOrigin

PATH PARAMETERS

`teamId` `required`	string Example: `coders`

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`message`	string Optional request message, if the team is not open.

Responses

> **200**

POST `/team/{teamId}/join`

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Leave a team

Leave a team.

- https://lichess.org/team

AUTHORIZATIONS: OAuth2 (`team:write`) **AND** SameOrigin

PATH PARAMETERS

`teamId` required	string Example: `coders`

Responses

> 200

POST `/team/{teamId}/quit`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Kick a user from your team

Kick a member out of one of your teams.

- https://lichess.org/team

AUTHORIZATIONS: OAuth2 (`team:write`) **AND** SameOrigin

PATH PARAMETERS

`teamId` `required`	string Example: `coders`
`userId` `required`	string Example: `neio`

Responses

> **200**

POST `/team/{teamId}/kick/{userId}`

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Message all members

Send a private message to all members of a team. You must own the team.

AUTHORIZATIONS: OAuth2 (`team:write`)

PATH PARAMETERS

`teamId` `required`	string Example: `coders`

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

`message`	string The message to send to all your team members.

Responses

› 200

› 400

POST /team/{teamId}/pm-all

Response samples

200 400

Content type
application/json

Copy Expand all Collapse all

```
{
    "ok": true
}
```

Analysis

Access Lichess cloud evaluations database. https://lichess.org/analysis

Get cloud evaluation of a position.

Get the cached evaluation of a position, if available.

Opening positions have more chances of being available. There are about 7 million positions in the database.

Up to 5 variations may be available. Variants are supported.

AUTHORIZATIONS: None

QUERY PARAMETERS

`fen`
`required`

string
Example: `fen=rnbqkbnr/ppp1pppp/8/3pP3/8/8/PPPP1PPP/RNBQKBNR b KQkq - 0 2`
FEN of the position

`multiPv`

number
Default: `1`
Number of variations

`variant`

string
Default: `"standard"`
Variant

Responses

— 200

GET `/api/cloud-eval`

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "fen": "rnbqkbnr/ppp1pppp/8/3pP3/8/8/PPPP1PPP/RNBQKBNR b KQkq - 0 2",
    "knodes": 13683,
    "depth": 22,
  - "pvs": [
      + { … },
      + { … },
      + { … }
    ]
}
```