

webDiplomacy API - No-press Bot Quick-start

Kestas Kuliukas – 2022-07-24

Overview

webDiplomacy has a simple RESTful style JSON based API, called via web requests, which is used to allow bots to interact with the site, as well as allowing client-side user interfaces.'

This document is intended to be a quick-start guide on the basic functionality used by no-press bots, as no-press bots have only three API calls they need to make to be fully functional.

A press-bot that reads and sends messages, or a human user, does all of the below but performs other operations to send messages, cast votes, etc. This makes no-press bots a good way to understand the API very quickly.

Note that the source code for the API is available at <https://github.com/kestasjk/webDiplomacy/blob/master/api.php>

Contents

Overview	1
Authentication	1
Available operations	2
Required operations	2
ListGamesWithMissingOrders: /api.php?route=players/missing_orders	3
GetGameStates - /api.php?route=game/status	4
SetOrders - /api.php?route=game/orders.....	6

Authentication

In order to interface with the API you need a user account and a key. First create an account on the site for the bot like a regular account, via the registration page, and then ask a site administrator to provide an API key to use for the bot.

*Please note: Using the API with web-browser cookie-based authentication as a bot is **not allowed**, as moderators cannot tell bots from users. Depending on the circumstances this could be considered cheating or data-scraping. Bots receive messages that have gone through an automatic word writelist.*

The site administrator will create a random secret key for you, stored in the wD_ApiKeys table, which links the key to the bot account user ID. The administrator will also create a record in wD_ApiPermissions, which is used to allow bot accounts to be able to get the state of all games, submit orders on behalf of users who are inactive, or list and join games that players have left.

Once you have been given an API key you can make requests against the API using any HTTP client by using an Authorization: Bearer header. E.g. given an API key of "SECRET123" you can use curl to make a request against the API like so:

```
# curl -H 'Authorization: Bearer SECRET123' http://localhost:43000/api.php?route=players/missing\_orders
```

Available operations

The API is fairly simple, with just a few endpoints defined:

```
// List players in civil disorder who have left a game, and can be taken over
    $api->load(new ListGamesWithPlayersInCD());
// List games where there are orders that need to be submitted
    $api->load(new ListGamesWithMissingOrders());
// List active / joined games
    $api->load(new ListActiveGamesForUser());
// Get the full data about a game state; unit locations, orders submitted, etc
    $api->load(new GetGamesStates());
// Get an overview of a game state; member information, supply center counts, etc
    $api->load(new GetGameOverview());
// Get the full data about a game state; unit locations, orders submitted, etc
    $api->load(new GetGameData());
// Get info about game members
    $api->load(new GetGameMembers());
// Submit orders
    $api->load(new SetOrders());
// Toggle a vote
    $api->load(new ToggleVote());
// Set a vote
    $api->load(new SetVote());
// Send a message
    $api->load(new SendMessage());
// Fetch messages
    $api->load(new GetMessages());
// Flag messages as seen
    $api->load(new MessagesSeen());
// Notify that you are present
    $api->load(new MarkBackFromLeft());
```

And for no-press functionality being covered here it is especially simple, as there are only three operations required:

```
// List games where there are orders that need to be submitted
    $api->load(new ListGamesWithMissingOrders());
// Get the full data about a game state; unit locations, orders submitted, etc
    $api->load(new GetGamesStates());
// Submit orders
    $api->load(new SetOrders());
```

Required operations

JSON request / responses below are formatted like so:

```
Sunday 24th of July 2022 12:48:56 AM ← Time of request
-----
/api.php?route=players%2Fmissing_orders ← HTTP Request
-----
[] ← Input data submitted (can be GET, POST, or JSON)
-----
[] ← Output data (JSON)
```

ListGamesWithMissingOrders: /api.php?route=players/missing_orders

This request will return a list of gameId/countryID pairs for any games that the bot needs to submit orders for.

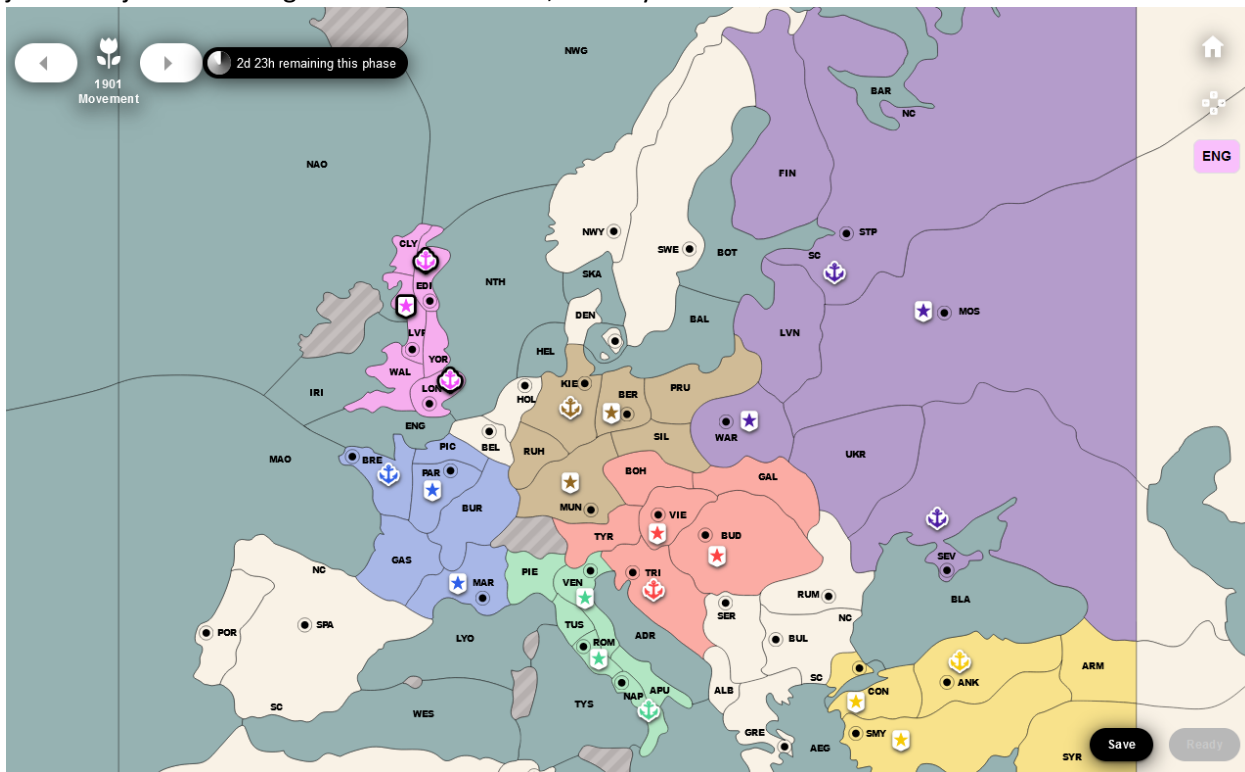
If the bot has no orders that need to be submitted an empty array is returned:

```
Sunday 24th of July 2022 12:48:56 AM
-----
/api.php?route=players%2Fmissing_orders
-----
[]
-----
[]
-----
```

If the bot does have orders to submit JSON like below is returned:

```
Sunday 24th of July 2022 12:49:01 AM
-----
/api.php?route=players%2Fmissing_orders
-----
[
  {
    "gameID": 3,
    "countryID": 2
  }
]
-----
```

Note that normally bots are assigned to games by the system automatically, so the bot is essentially notified that it has joined a game when it checks for orders it needs to submit and sees a new game in there. In this example this bot has just been joined to the game below as France, country 2:



This request should be made at least once every few seconds to ensure players don't wait for bots to submit orders.

GetGameStates - /api.php?route=game/status

This request requires a gameId and countryID be provided (typically from a ListGamesWithMissingOrders request), and will return the full game board state, including all previous phases in the game.

If making a client application that doesn't have built-in map data you can fetch the map data from [https://webdiplomacy.net/variants/\[variant\]/cache/territories.js](https://webdiplomacy.net/variants/[variant]/cache/territories.js)

Sunday 24th of July 2022 12:49:01 AM

/api.php?route=game%2Fstatus&gameID=3&countryID=2

```
{
  "gameID": "3",          ← Input gameId and countryID
  "countryID": "2"        ← CountryIDs are defined in the variants/[variant]/variant.php file.
                           e.g. Classic:'England','France','Italy','Germany','Austria','Turkey','Russia'
}
-----
{
  "gameID": 3,            ← Returned JSON object will full details
  "countryID": 2,
  "variantID": 1,         ← Variant ID, classic is most common
  "potType": "Unranked",  ← Ranking system used in game
  "turn": 0,              ← Which turn; 0 is Spring 1901, 1 is Autumn 1901, 2 is Spring 1902, etc
  "phase": "Diplomacy",   ← The phase of the turn; Pre-game/Diplomacy/Retreats/Builds/Finished
  "gameOver": "No",       ← Is the game over
  "pressType": "Regular", ← Is this an full-press game, global-chat only, or no-press game
  "phases": [             ← The board state for each phase
    {
      "units": [          ← The units on the board
        {
          "unitType": "Fleet",
          "retreating": "No", ← If in Retreats phase specifies whether the unit is retreating
          "terrID": 2,        ← The territory ID the unit is in, territories can be
                               fetched in JSON format from
                               variants/[variant]/cache/territories.js
          "countryID": 1
        },
        {
          "unitType": "Army",
          "retreating": "No",
          "terrID": 3,
          "countryID": 1
        },
        [.....]
        {
          "unitType": "Army",
          "retreating": "No",
          "terrID": 74,
          "countryID": 5
        }
      ],
      "centers": [        ← The supply centers territories and the owning country
        {
          "terrID": 1,
          "countryID": 1
        },
        {
          "terrID": 2,
          "countryID": 1
        },
        [.....]
        {
          "terrID": 81,
          "countryID": 0
        }
      ],
      "turn": 0,          ← The turn and phase of this board state array
    }
  ]
}
```

<pre> "phase": "Diplomacy", "orders": [] }], "standoffs": [], "occupiedFrom": [], "votes": null, "orderStatus": "", "status": "Playing", "drawType": "draw-votes-public", "processTime": "1658882937", "phaseLengthInMinutes": "4320", "orderStatuses": { "1": "", "2": "", "5": "", "3": "", "6": "", "4": "", "7": "" }, "publicVotes": { "1": null, "2": null, "5": null, "3": null, "6": null, "4": null, "7": null } } } </pre>	<pre> ← Orders each player has made (none present for the first turn) ← Standoffs caused by units that bounced in a Diplomacy phase ← A list of newly occupied territories during the last Diplomacy phase, and where from. ← The requesting account's votes ← The requesting account's order status. Ready means ready for next phase ← The requesting account's status Playing/Left/Defeated/Won ← The draw setting; are draw votes public or hidden etc ← A unix timestamp for when the next process will occur (assuming players don't set their orders to Ready) ← The maximum length in minutes per phase ← Order statuses for each other country ← Votes submitted for each other country </pre>
---	---

SetOrders - /api.php?route=game/orders

The final operation no-press bots are required to implement to function; order submission. After being notified of a game missing orders, then fetching the state of the game, the bot processes the game state and generates its orders to submit.

Sunday 24th of July 2022 12:49:01 AM

/api.php?route=game%2Forders

```
{
  "gameID": 3,                ← Game ID, turn, phase, country
  "turn": 0,
  "phase": "Diplomacy",
  "countryID": 2,
  "orders": [                ← List of orders
    {
      "terrID": 46,           ← Territory the order is starting from
      "unitType": "Fleet",    ← Unit type in the territory
      "type": "Move",         ← Order type: Hold/Move/Support Hold/Support Move/Convoy
      "toTerrID": 61,         ← To territory ID, if applicable (e.g. move to here)
      "fromTerrID": "",       ← From territory ID, if applicable (e.g. support a unit from here)
      "viaConvoy": "No"       ← If an army and a move, is it going via convoy
    },
    {
      "terrID": 47,
      "unitType": "Army",
      "type": "Move",
      "toTerrID": 48,
      "fromTerrID": "",
      "viaConvoy": "No"
    },
    {
      "terrID": 49,
      "unitType": "Army",
      "type": "Move",
      "toTerrID": 14,
      "fromTerrID": "",
      "viaConvoy": "No"
    }
  ],
  "ready": "Yes"             ← If true this country is ready for the game to proceed. When all countries are
                              ready the game will move straight to the next phase.
}
-----
[
  {
    "unitType": "Fleet",      ← The returned saved orders, that have been validated by the server.
    "terrID": 46,
    "type": "Move",
    "fromTerrID": null,
    "toTerrID": 61,
    "viaConvoy": "No"
  },
  {
    "unitType": "Army",
    "terrID": 47,
    "type": "Move",
    "fromTerrID": null,
    "toTerrID": 48,
    "viaConvoy": "No"
  },
  {
    "unitType": "Army",
    "terrID": 49,
    "type": "Move",
    "fromTerrID": null,
    "toTerrID": 14,
    "viaConvoy": "No"
  }
]
-----
```