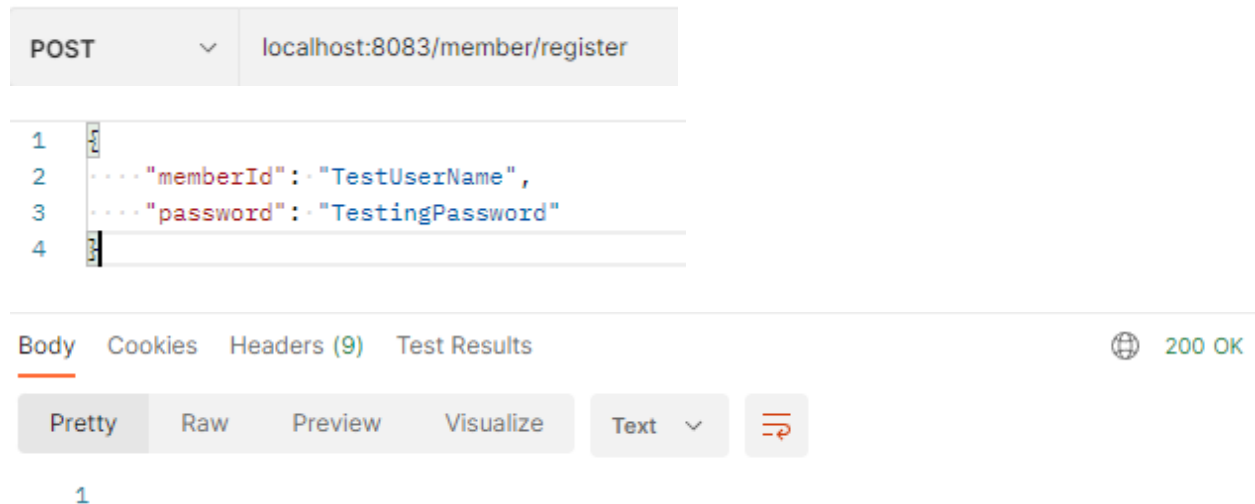


Registering

Type your credentials at registration



The screenshot shows a REST client interface. The top bar indicates a POST request to `localhost:8083/member/register`. The request body is a JSON object with the following content:

```
1 {
2   ... "memberId": "TestUserName",
3   ... "password": "TestingPassword"
4 }
```

Below the request body, the response is shown. The status is `200 OK`. The response body is empty, and the format is set to `Text`.

Trying to do register a second time (with same memberId, password is fine) throws a bad request



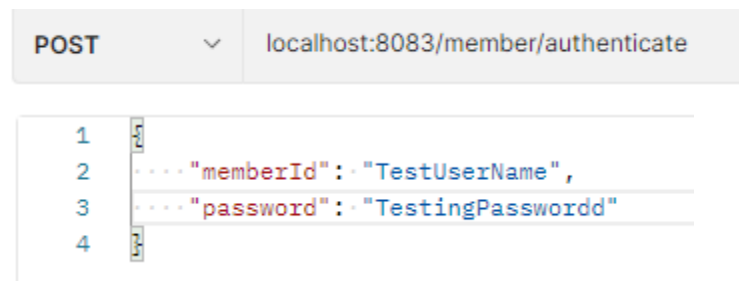
The screenshot shows a REST client interface. The top bar indicates a POST request to `localhost:8083/member/register`. The response status is `400 Bad Request`. The response body is a JSON object with the following content:

```
1 {
2   "timestamp": "2022-12-23T08:02:41.263+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Member already exists",
6   "path": "/member/register"
7 }
```

The response body is displayed in JSON format.

Authentication

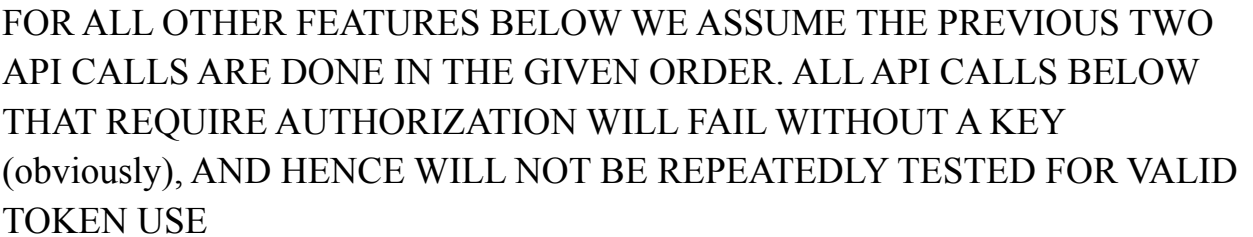
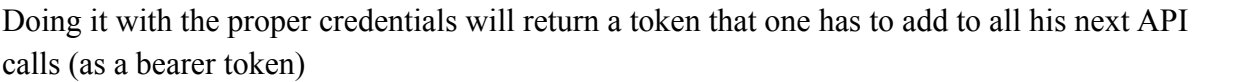
First trying to do it with wrong credentials - throws an invalid credential error (401, expected over 404 or 400 to not disclose a user's availability)



The screenshot shows a REST client interface. The top bar indicates a POST request to `localhost:8083/member/authenticate`. The request body is a JSON object with the following content:

```
1 {
2   ... "memberId": "TestUserName",
3   ... "password": "TestingPasswordd"
4 }
```

The request body is displayed in JSON format.




Changing a password

Trying to update password without an auth token throws an unauthorized error

POST	localhost:8083/member/updatePassword
------	--------------------------------------


```
1 {
2   ... "memberId": "TestUserName",
3   ... "password": "TestingPassword"
4 }
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON 

```
1 {
2   "timestamp": "2022-12-23T08:10:25.444+00:00",
3   "status": 401,
4   "error": "Unauthorized",
5   "message": "INVALID_CREDENTIALS",
6   "path": "/member/authenticate"
7 }
```

Database overview:

SELECT * FROM MEMBERS;

MEMBERID	PASSWORD_HASH
TestUserName	\$2a\$10\$b8ffukhGoRppoQs/owj4zOSTqwQm6PK.BMr/gP/MGkJAi2UH1/IRe
TestUserName2	\$2a\$10\$1R1oPivr2KKIDdEH3M.mquEfGLJbtjJBe10DPpeW6.55Gn6VtxSUM

(2 rows, 2 ms)

Creating an HOA

POST localhost:8084/hoa/create

```
1 {
2   ... "country": "Germany",
3   ... "city": "Hamburg",
4   ... "name": "RandomAssoc"
5 }
```

```
Body Cookies Headers (10) Test Results 200 OK
Pretty Raw Preview Visualize JSON
1 {
2   "id": 1,
3   "country": "Germany",
4   "city": "Hamburg",
5   "name": "RandomAssoc",
6   "reports": {},
7   "notifications": {}
8 }
```

Trying to create an HOA with a name that is already taken fails

```
Body Cookies Headers (9) Test Results 400 Bad Request
Pretty Raw Preview Visualize JSON
1 {
2   "timestamp": "2022-12-23T08:19:11.744+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Hoa with name RandomAssoc is already taken.",
6   "path": "/hoa/create"
7 }
```

And one with bad credentials also

```
1 {
2   ... "country": "Germany",
3   ... "city": "Hamburg",
4   ... "name": ""
5 }
Body Cookies Headers (9) Test Results 400 Bad Request
Pretty Raw Preview Visualize JSON
1 {
2   "timestamp": "2022-12-23T08:19:46.521+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Wrong format for the country, city, or name!",
6   "path": "/hoa/create"
7 }
```

Database overview:

SELECT * FROM HOA;

ID	CITY	COUNTRY	NAME	NOTIFICATIONS	REPORTS
1	Hamburg	Germany	RandomAssoc		
2	Hamburgg	Germanyy	RandomAssocc		

Fetch an HOA

Fetching an HOA requires a 'GET' call to the proper endpoint

GET	▼	localhost:8084/hoa/getById/2
-----	---	------------------------------

Body Cookies Headers (10) Test Results

200 OK

Pretty	Raw	Preview	Visualize	JSON ▼	≡
--------	-----	---------	-----------	--------	---

```
1 {
2   "id": 2,
3   "country": "Germanyy",
4   "city": "Hamburgg",
5   "name": "RandomAssocc",
6   "reports": {},
7   "notifications": {}
8 }
```

Trying to fetch a non-existent HOA fails

GET	▼	localhost:8084/hoa/getById/3
-----	---	------------------------------

```
1 {
2   "timestamp": "2022-12-23T08:33:13.145+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Could not fetch HOA",
6   "path": "/hoa/getById/3"
7 }
```

Join a HOA

POST	▼	localhost:8083/member/joinHOA
------	---	-------------------------------

Required request model format:

```

{
  .... "memberId": "TestUserName",
  .... "hoaId": 1,
  .... "address": {
    .... "country": "Germany",
    .... "city": "Hamburg",
    .... "street": "Randomstreeter",
    .... "houseNumber": "244",
    .... "postalCode": "2222AB"
  }
}

```


Non-existent id

```

1  {
2  .... "memberId": "TestUserName",
3  .... "hoaId": 3,
4  .... "address": {
5  .... |     "country": "Germany",
6  .... |     "city": "Hamburg",
7  .... |     "street": "Randomstreeter",
8  .... |     "houseNumber": "244",
9  .... |     "postalCode": "2222AB"
10 .... }
11 }

```

ody Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON 

```

1  {
2  "timestamp": "2022-12-23T08:42:48.705+00:00",
3  "status": 400,
4  "error": "Bad Request",
5  "message": "HTTP 400 Bad Request",
6  "path": "/member/joinHOA"
7  }

```

Invalid address

```

1  ✓
2  .... "memberId": "TestUserName",
3  .... "hoaId": 2,
4  ✓  .... "address": {
5  |    .... "country": "Germany",
6  |    .... "city": "Hamburg",
7  |    .... "street": "Randomstreeter",
8  |    .... "houseNumber": "244231312312",
9  |    .... "postalCode": "2222AB"
10 |  .... }
11

```

```

"timestamp": "2022-12-23T08:46:13.425+00:00",
"status": 400,
"error": "Bad Request",
"message": "TestUserName 2",
"path": "/member/joinHOA"

```

Member already in HOA

```

.... "memberId": "TestUserName",
.... "hoaId": 1,
.... "address": {
.... |    .... "country": "Germany",
.... |    .... "city": "Hamburg",
.... |    .... "street": "Randomstreet",
.... |    .... "houseNumber": "2442",
.... |    .... "postalCode": "2222AB"
.... }

```

```

"timestamp": "2022-12-23T08:48:44.593+00:00",
"status": 400,
"error": "Bad Request",
"message": "TestUserName 1",
"path": "/member/joinHOA"

```

Database overview:

SELECT * FROM MEMBERSHIPS;						
MEMBERSHIPID	ADDRESS	DURATION	HOAID	IS_BOARD	MEMBERID	START_TIME
1	Germany,Hamburg,Randomstreeter,244,2222AB	null	1	FALSE	TestUserName	2022-12-23 09:39:56.978201

(1 row, 2 ms)

Leaving a HOA

POST localhost:8083/member/leaveHOA

Cannot leave an HOA he is not a part of

```
{
  ... "memberId": "TestUserName",
  ... "hoaId": 2
}
```

```
1 {
2   "timestamp": "2022-12-23T08:51:36.134+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Hoa or member are not found.",
6   "path": "/member/leaveHOA"
7 }
```

Expected call

```
{
  ... "memberId": "TestUserName",
  ... "hoaId": 1
}
```

```
{
  "membershipId": 1,
  "memberId": "TestUserName",
  "hoaId": 1,
  "address": {
    "country": "Germany",
    "city": "Hamburg",
    "street": "Randomstreeter",
    "houseNumber": "244",
    "postalCode": "2222AB"
  },
  "startTime": "2022-12-23T09:39:56.978201",
  "duration": "00:12:30",
  "inBoard": false
}
```

Memberships are persisted in DB, only changed when one leaves

```
SELECT * FROM MEMBERSHIPS;
```

MEMBERSHIPID	ADDRESS	DURATION	HOAID	IS_BOARD	MEMBERID	START_TIME
1	Germany,Hamburg,Randomstreeter,244,2222AB	00:12:30	1	FALSE	TestUserName	2022-12-23 09:39:56.978201

(1 row, 1 ms)

Duration changes, and membership is saved. Hence active vs inactive memberships are differentiated by the duration being null or not

Not idempotent - cannot leave a second time in a row

```
... "memberId": "TestUserName",
... "hoaId": 1

{
  "timestamp": "2022-12-23T08:54:25.540+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "Hoa or member are not found.",
  "path": "/member/leaveHOA"
```

If a path is specified with an integer, one can also start a board election automatically x days after hoa was created

POST localhost:8084/hoa/create/1

```
... "country": "Germany",
... "city": "Hamburg",
... "name": "RandomAssoc"
```

(not continuous session, ignore id incoherence)

200 OK

```
{
  "id": 2,
  "country": "Germany",
  "city": "Hamburg",
  "name": "RandomAssoc",
  "reports": {},
  "notifications": {}
}
```

SELECT * FROM ELECTIONS;

ELECTION_TYPE	ELECTION_ID	DESCRIPTION	HOA_ID	NAME	SCHEDULED_FOR	STATUS	VOTE_COUNT	AMOUNT_OF_WINNERS	CANDIDATES	VOTES	WINNING_CHOICE
0	1	This is the auto-generated annual board election	2	Annual board election	2022-12-24 10:03:18	scheduled	0	5			null

1 row, 4 ms)

Retrieving the public board for a specific HOA.

http://localhost:8084/activity/publicBoard/1/1

GET http://localhost:8084/activity/publicBoard/1/1

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Type Bearer Token Token

The authorization header will be automatically generated when

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "activityName": "jogging",
4   "activityDescription": "just jogging",
5   "hoaId": 1,
6   "activityTime": "2023-12-05T20:42:45.162685",
7   "activityDuration": "02:00:00",
8   "participants": [],
9   "expired": false
10 }
11
12
```

Join an activity.

http://localhost:8084/activity/join/42/2

PUT http://localhost:8084/activity/join/42/2

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

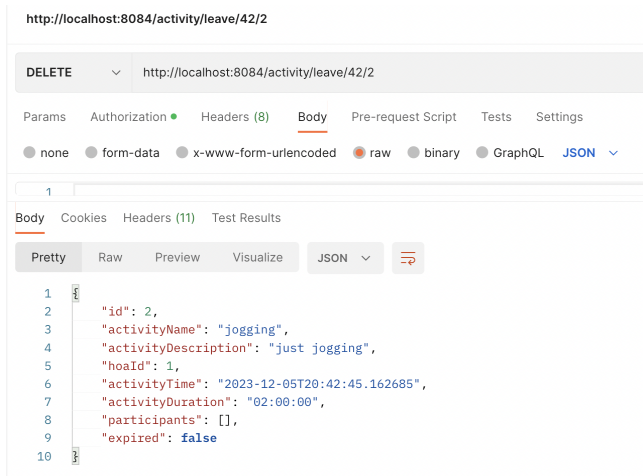
1

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

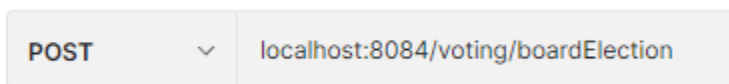
```
1 {
2   "id": 2,
3   "activityName": "jogging",
4   "activityDescription": "just jogging",
5   "hoaId": 1,
6   "activityTime": "2023-12-05T20:42:45.162685",
7   "activityDuration": "02:00:00",
8   "participants": [
9     42
10 ],
11   "expired": false
12 }
```

Leave an activity.

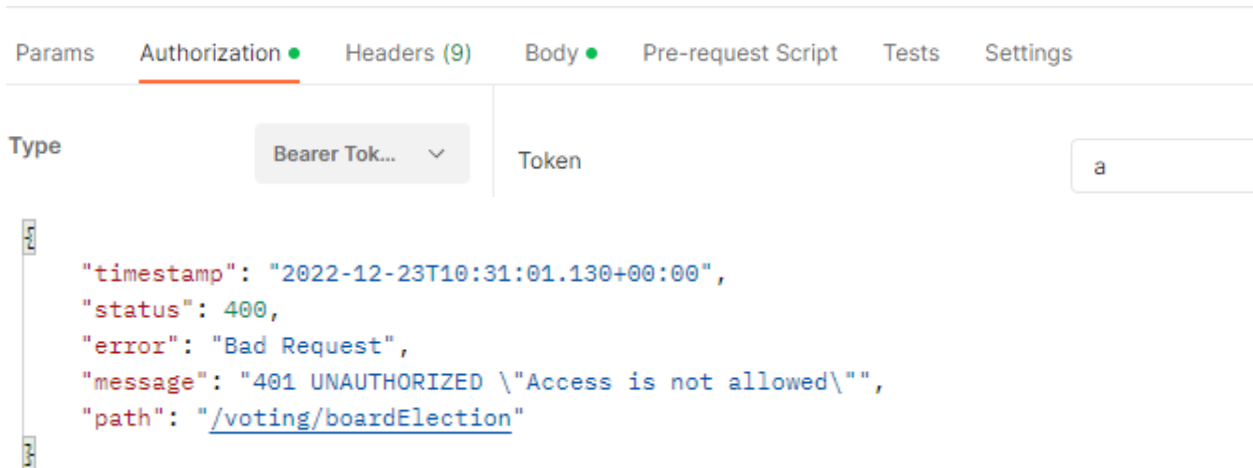


Starting a board election

If not automatic, one can start it manually



Invalid auth token (not in Hoa, will show this once at least)



Invalid date (in the past)

```

{
  "hoaId": 1,
  "name": "TestBoardElection",
  "amountOfWinners": 2,
  "candidates": [],
  "description": "This is a test board election",
  "scheduledFor": {
    "seconds": 10,
    "minutes": 10,
    "hours": 10,
    "day": 22,
    "month": 12,
    "year": 2022
  }
}

```

```

"timestamp": "2022-12-23T10:23:22.045+00:00",
"status": 400,
"error": "Bad Request",
"message": "HTTP 400 Bad Request",
"path": "/voting/boardElection"

```

Hoa id invalid (non-existent, or no access)

```

{
  "hoaId": 10,
  "name": "TestBoardElection",
  "amountOfWinners": 2,
  "candidates": [],
  "description": "This is a test board election",
  "scheduledFor": {
    "seconds": 10,
    "minutes": 24,
    "hours": 10,
    "day": 20,
    "month": 12,
    "year": 2022
  }
}

```

```
"timestamp": "2022-12-23T10:47:05.235+00:00",
"status": 400,
"error": "Bad Request",
"message": "401 UNAUTHORIZED \\"Access is not allowed\\",
"path": "/voting/boardElection"
```

Not all candidates provided are in the HOA

```
1  {
2    "hoaId": 1,
3    "name": "TestBoardElection",
4    "amountOfWinners": 2,
5    "candidates": ["blah"],
6    "description": "This is a test board election",
7    "scheduledFor": {
8      "seconds": 10,
9      "minutes": 25,
10     "hours": 10,
11     "day": 25,
12     "month": 12,
13     "year": 2022
14   }
}
```

```
"timestamp": "2022-12-23T10:47:13.550+00:00",
"status": 400,
"error": "Bad Request",
"message": "400 BAD_REQUEST \\"Not all candidates are in Hoa\\",
"path": "/voting/boardElection"
```

Valid api call

```
{
  "electionId": 8,
  "hoaId": 1,
  "name": "TestBoardElection",
  "description": "This is a test board election",
  "voteCount": 0,
  "status": "scheduled",
  "scheduledFor": "2022-12-23T11:48:10",
  "amountOfWinners": 2,
  "candidates": [
    "TestUserName"
  ],
  "votes": {}
}
```

Joining a board election

Valid join:

POST localhost:8084/voting/joinElection/1

Body Cookies Headers (10) Test Results

1 true

Status: 200 OK

No idempotence/double join (try to join same election twice):

Body Cookies Headers (10) Test Results

1 false

Status: 200 OK Time: 286 ms Size: 32B

Invalid hoaid (unauthorized or non-existent):

POST localhost:8084/voting/joinElection/696969

{
 "timestamp": "2022-12-23T12:22:08.573+00:00",
 "status": 400,
 "error": "Bad Request",
 "message": "The HOA has no running election.",
 "path": "/voting/joinElection/696969"
}

NOTE: ONE OF THE REQUIREMENTS FOR JOINING AN ELECTION AS A CANDIDATE IS TO BE A MEMBER OF THE HOA FOR A PROLONGED AMOUNT OF TIME, AS IN YEARS' WORTH, HENCE THE JOINING CANNOT FUNCTION IN REAL-TIME TESTING WITHOUT COMMENTING OUT THE HANDLER.

Voting on a board election

localhost:8084/voting/vote

Invalid vote options:

It was tough to make the messages be transported properly, so the error message is not helpful here ;-;

- Invalid choice (empty):

```
{
  "electionId": 37,
  "memberId": "TestUserName",
  "choice": ""
}
```

```
{
  "timestamp": "2022-12-23T12:02:21.474+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "400 BAD_REQUEST \\\"HTTP 400 Bad Request\\\"; nested exception is javax.ws.rs.BadRequestException: HTTP 400 Bad Request",
  "path": "/voting/vote"
}
```

- Invalid choice (non-existent candidate):

```
1 {
2   "electionId": 1,
3   "memberId": "TestUserName",
4   "choice": "jimmy"
5 }
```

```
{
  "timestamp": "2022-12-23T12:05:24.248+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "400 BAD_REQUEST \\\"HTTP 400 Bad Request\\\"; nested exception is javax.ws.rs.BadRequestException: HTTP 400 Bad Request",
  "path": "/voting/vote"
}
```

- Voting on yourself:


```

1  {
2    "electionId": 37,
3    "memberId": "TestUserName",
4    "choice": "TestUserName"
5  }

```

```

{
  "timestamp": "2022-12-23T12:03:20.428+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "400 BAD_REQUEST \"HTTP 400 Bad Request\"; nested exception is javax.ws.rs.BadRequestException: HTTP 400 Bad Request",
  "path": "/voting/vote"
}

```

- Invalid electionid (unaccessible or non-existent):

```

1  {
2    "electionId": 1,
3    "memberId": "TestUserName",
4    "choice": "TestUserName"
5  }

```

```

{
  "timestamp": "2022-12-23T12:03:54.431+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "400 BAD_REQUEST \"HTTP 400 Bad Request\"; nested exception is javax.ws.rs.BadRequestException: HTTP 400 Bad Request",
  "path": "/voting/vote"
}

```

- Election hasn't started yet:

```

{
  "electionId": 2,
  "memberId": "TestUserName2",
  "choice": "TestUserName"
}

```

```

{
  "timestamp": "2022-12-23T12:37:17.089+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "400 BAD_REQUEST \"HTTP 400 Bad Request\"; nested exception is javax.ws.rs.BadRequestException: HTTP 400 Bad Request",
  "path": "/voting/vote"
}

```

Valid vote:

```

1  {
2    ... "electionId": 1,
3    ... "memberId": "TestUserName2",
4    ... "choice": "TestUserName"
5  }

```

Database overview:

ELECTION_ID	DESCRIPTION	HOA_ID	NAME	SCHEDULED_FOR	STATUS	VOTE_COUNT	A
1	This is a test board election	1	TestBoardElection	2022-12-23 13:30:00	ongoing	1	2
2	This is a test board election	2	TestBoardElection	2022-12-24 13:30:00	scheduled	0	2

CANDIDATES	VOTES	WINNING_CHOICE
TestUserName	TestUserName2=TestUserName	null
		null

Concluding an election

Returns winners

OST localhost:8084/voting/conclude/1

Status: 200 OK

```

[
  "TestUserName"
]

```

Creates a new membership where the winners are now board members

4	Germany,Hamburg,Randomstreet,2442,2222AB	null	1	TRUE	TestUserName	2022-12-23 14:45:38.659532
---	--	------	---	------	--------------	----------------------------

(4 rows, 11 ms)

And starts a new election that is scheduled for 1 year after the last election's scheduled date (if its a board election)

Notifies members of a rule change otherwise

0	4	This is the auto-generated annual board election	1	Annual board election	2023-12-23 13:30:00	scheduled	0	2
---	---	--	---	-----------------------	---------------------	-----------	---	---

(2 rows, 0 ms)

