



Testing Document

Evotekaro

Types of testing used:

Data Flow testing

Black box testing:

- API Testing

Methods used to streamline number of test cases:

Boundary condition testing

Software used:

FastAPI documentation (Swagger UI)

API access for type of user:

Normal Users

Able to access:

- Home/ get/
- Authentication/ login
- Election/election (all)
- Election/election {id} (view)
- Vote/ votes (create)
- Vote/ votes {id} (result)
- Candidates/ candidates (show candidates)
- Candidates/ candidates {id} (get candidates)

Are not able to access:

- Users/user(show user)
- Users/user(create user)
- Users/user{id} (get user)
- Users/user{id} (update)
- User/user{id} (Destroy)
- Election/election (create)
- Election/election {id} (update)
- Election/election {id} (destroy)
- Candidates/ candidates (create candidates)
- Candidates/ candidates {id} (update)

- Candidates/ candidates {id} (destroy)

Admin Users

Able to access:

- Home/ get/
- Authentication/ login
- Users/user(show user)
- Users/user(create user)
- Users/user{id} (get user)
- Users/user{id} (update)
- User/user{id} (Destroy)
- Election/election (all)
- Election/election (create)
- Election/election {id} (view)
- Election/election {id} (update)
- Election/election {id} (destroy)
- Vote/ votes (create)
- Vote/ votes {id} (result)
- Candidates/ candidates (show candidates)
- Candidates/ candidates (create candidates)
- Candidates/ candidates {id} (get candidates)
- Candidates/ candidates {id} (update)
- Candidates/ candidates {id} (destroy)

API Responses:

APIs which does not accept any parameters:

- Home/ get/
- Users/user(show user)
- Election/election (all)
- Candidates/ candidates (show candidates)

APIs which accepts specific parameters:

- Authentication/ login
- Users/user(create user)
- Users/user{id} (get user)
- Users/user{id} (update)
- User/user{id} (Destroy)
- Election/election (create)
- Election/election {id} (view)
- Election/election {id} (update)

- Election/election {id} (destroy)
- Vote/ votes (create)
- Vote/ votes {id} (result)
- Candidates/ candidates (create candidates)
- Candidates/ candidates {id} (get candidates)
- Candidates/ candidates {id} (update)
- Candidates/ candidates {id} (destroy)

Various response codes:

(200, 201, 202) Successful Response

(400) Error: Bad Request

(403) Error: Forbidden

(404) Error: Not Found

(406) Error: Not Acceptable

(422) Unprocessable Entity - **This only happens when the request body is not in the correct format.**

Testing Results:

- Authentication

Authentication/ login:- (404) Error: Not Found {**Invalid Credentials, Incorrect password**}

- Users

Users/user (create user):- (422) Unprocessable Entity

Users/user{id} (get user):- (404) Error: Not found{**User with id not found**}

User/user{id} (Destroy):- (406) Not Acceptable{**Keep the admin user in your app**}; (404) Error Not Found {**User with id not found**}

- Election

Election/election {id} (view):- (404) Error: Not Found{**Election with the id is not available**}

Election/election {id} (Destroy):- (200) Successful Response{**Deleted**}; (404) Error Not Found {**Election with id not found**}

- Votes

Vote/ votes (create):- (404) Error: Not Found {**Candidate not found, User not found**}; (400) Error: Bad Request {**User has already voted for this election**}

Vote/ votes {id} (result):- (404) Error Not Found {**Election with id not found**}; (403) Error: Forbidden (**Election results are not available yet**)

- Candidates

Candidates/ candidates {id} (get candidates):- (404) Error: Not Found{**Candidate with the id is not available**}

Candidates/ candidates {id} (destroy):- (404) Error Not Found {**Candidate with id not found**}

APIs which are being patched:

Users/user{id} (update) [(202) Successful Response]

Election/election (create) [(201)Successful Response;(422) Unprocessable Entity;]

Election/election {id} (update) [(202)Successful Response {Updated}; (422) Unprocessable Entity]

Candidates/ candidates (create candidates) [(200)Successful Response; (422) Unprocessable Entity]

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8000/' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8000/
```

Server response

Code	Details
200	<p>Response body</p> <pre>"Evotekare - Secure Voting System go to '/docs' for api documentation"</pre> <p>Response headers</p> <pre>content-length: 72 content-type: application/json date: Mon, 19 Jun 2023 06:54:33 GMT server: uvicorn</pre>

Responses

Code	Description	Links
200	Successful Response	No links

This is an example of a successful response given by the parameter-less API, Home/ get/

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8000/user/' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8000/user/
```

Server response

Code	Details
401 <i>Undocumented</i>	<p>Error: Unauthorized</p> <p>Response body</p> <pre>{ "detail": "Not authenticated" }</pre>

Trying to access APIs without logging in.

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8000/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -d 'grant_type=USERNAME&password=abc&scope=&client_id=&client_secret='
```

Request URL

http://localhost:8000/login

Server response

Code	Details
404	Error: Not Found

Response body

```
{
  "detail": "Invalid Credentials"
}
```

Response headers

```
access-control-allow-credentials: true
content-length: 32
content-type: application/json
date: Mon, 19 Jun 2023 07:00:56 GMT
server: uvicorn
```

This is the response given by the API, Authentication/ login when invalid username is given.

Server response

Code	Details
404	Error: Not Found

Response body

```
{
  "detail": "Incorrect password"
}
```

Response headers

```
access-control-allow-credentials: true
content-length: 31
content-type: application/json
date: Mon, 19 Jun 2023 07:03:38 GMT
server: uvicorn
```

Same API, but different response for an incorrect password.

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8000/candidates/15' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIjYm51c2hAahlpdGsuY29tIiwidXN1ck1kIjoxLjC3p0FkbWl1Ijp0cnV1LCJ1YXRjaCI6IjY1bnJhbmNoIjo1Ym51IiwiaWF0Ij01NjYyMjQ5MTQ5fQ.'
```

Request URL

http://localhost:8000/candidates/15

Server response

Code	Details
200	

Response body

```
{
  "id": 15,
  "name": "GANESH",
  "electionId": 12,
  "manifesto": "first"
}
```

Successfully getting the candidate details from Candidates/ candidates {id} Get Candidates API.

The screenshot displays a REST client interface with the following sections:

- Responses:** A large empty area at the top.
- Curl:** A dark-themed text area containing a cURL command:

```
curl -X 'POST' \  
'http://localhost:8000/votes/' \  
-H 'accept: application/json' \  
-H 'authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJnYW51c2haaWpdGsuY29tIiwidXN1c2lkIjoxLjC3pc0FkbWluIjp0cnV1LCJiYXRjaCI6MSw1YnJhbWNoIjoIYmNSIiwiekhVhciI6IjAyMSw1ZXhwIjojNjg3MTcwMTQ5fQ.' \  
-H 'Content-Type: application/json' \  
-d '{  
  "id": 3,  
  "userId": 1,  
  "electionId": 1,  
  "candidateId": 26  
}'
```
- Request URL:** A dark bar containing the URL `http://localhost:8000/votes/`.
- Server response:** A section showing the response details:
 - Code:** 404
 - Details:** Error: Not Found
- Response body:** A dark bar containing the JSON response:

```
{  
  "detail": "Candidate not found"  
}
```

At the bottom right, there are two buttons: a trash icon and a button labeled "Download".

Curl

```
curl -X 'POST' \
'http://localhost:8000/votes/' \
-H 'accept: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpvcXVCJ9.eyJzdWIiOiJlbnV5MkZhaApldGsuY29tZWxldXNlc1klIjoxLjCjpc0FkbWluZjp0cnV1LCJ1eXRjaCI6MSwiYnJhbnoMcjoieWYmNSIweHvHciOiEwImJAYmswiZXhwIjoxNjc3MTcwMTQ5fQ.' \
-H 'Content-Type: application/json' \
-d '{
  "id": 3,
  "userId": 1,
  "electionId": 1,
  "candidateId": 26
}
```

Request URL

<http://localhost:8000/votes/>

Server response

Code	Details
------	---------

404
Undocumented Error: Not Found

Response body

```
{
  "detail": "Candidate not found"
}
```


Download

Trying to vote for a candidate who isn't present.

The screenshot displays a REST client interface with the following sections:

- GET /candidates/ Show Candidates**: The method and endpoint are selected.
- Parameters**: A section with the text "No parameters".
- Execute**: A blue button to execute the request.
- Responses**: A section containing:
 - Curl**: A terminal window showing the curl command:

```
curl -X 'GET' \  
'http://localhost:8000/candidates/' \  
-H 'accept: application/json' \  
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXV'
```
 - Request URL**: A terminal window showing the URL:

```
http://localhost:8000/candidates/
```
 - Server response**: A table with two columns: **Code** and **Details**.

Code	Details
401	Error: Unauthorized
 - Response body**: A terminal window showing the JSON response:

```
{  
  "detail": "Could not validate admin access"  
}
```
 - Response headers**: A section that is currently empty.

Parameters

No parameters

Execute

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8000/candidates/' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC'
```

Request URL

```
http://localhost:8000/candidates/
```

Server response

Code	Details
------	---------

401
Undocumented Error: Unauthorized

Response body

```
"detail": "Could not validate admin access"
```

Response headers

[illegible]

integer
(*path*)

2

Execute

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8000/votes/2' \
-H 'accept: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ1YXVQIGpXRRRl' \
```

Request URL

`http://localhost:8000/votes/2`

Server response

Code	Details
------	---------

401
Undocumented Error: Unauthorized

Response body

```
"detail": "Could not validate admin access"
```


The user was unable to view candidate details.