(FTC) Online Voting – Part II. Single-Node e-Voting Server

Local Server API

VerifyAuthToken (auth_token) return (True / False)

Some RPCs like **CreateElection**, **CastVote** need to check auth token's validity first, this local API can do so.

There are 3 situations :

- When given **auth_token** is not in the **auth_tokens** dictionary, return **False**
- When given **auth_token** is expired, return **False**
- When given **auth_token** is valid, server will update its expired time and return **True**

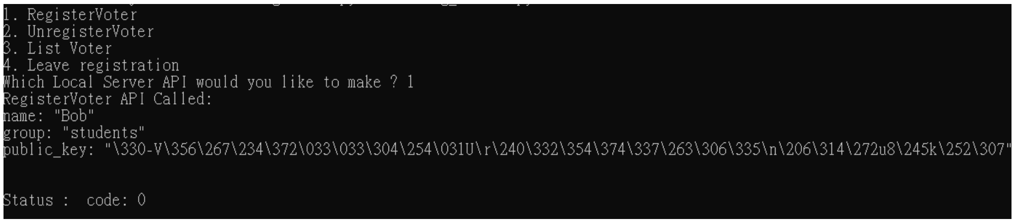
RegisterVoter(Voter) returns (Status)

We can register a new Voter by calling the RegisterVoter API.

- The server side can use voter.txt to register voters.
- voter.txt will contain the voter's name, group, and base64 encoded of a seed.
- The seed will use to generate a public key.
- Return Value
    - Status.code=0 : Successful registration
    - Status.code=1 : Voter with the same name already exists
    - Status.code=2 : Undefined errorFirst, choose 1 to register all voters in voter.txt.

Case of Status.code=0

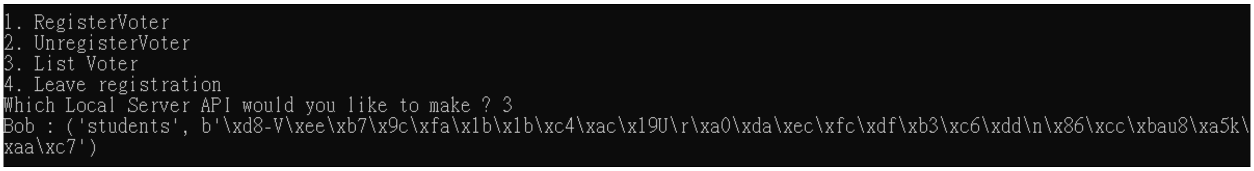
First, we can register voters by choosing 1.



```
1. RegisterVoter
2. UnregisterVoter
3. List Voter
4. Leave registration
Which Local Server API would you like to make ? 1
RegisterVoter API Called:
name: "Bob"
group: "students"
public_key: "\330-V\356\267\234\372\033\033\304\254\031U\r\240\332\354\374\337\263\306\335\n\206\314\272u8\245k\252\307"


Status :  code: 0
```

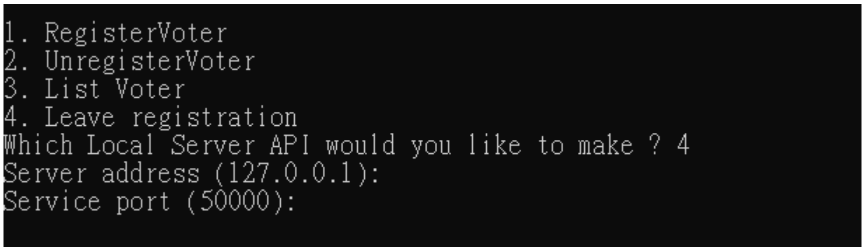
Then we can see all registered voters by choosing 3.



```
1. RegisterVoter
2. UnregisterVoter
3. List Voter
4. Leave registration
Which Local Server API would you like to make ? 3
Bob : ('students', b'\xd8-V\xee\xb7\x9c\xfa\x1b\x1b\xc4\xac\x19U\r\xa0\xda\xec\xfc\xdf\xb3\xc6\xdd\n\x86\xcc\xbau8\xa5k\
xaa\xc7')
```

Second, after registering all voters, you can leave registration by choosing 4.

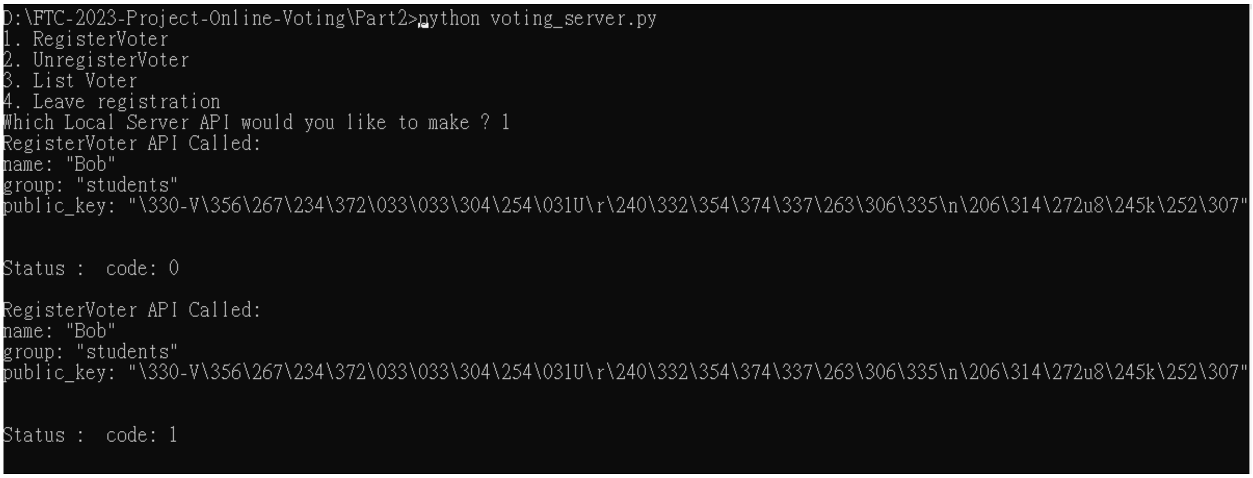
After inputting the server address/port, the server will be running.



```
1. RegisterVoter
2. UnregisterVoter
3. List Voter
4. Leave registration
Which Local Server API would you like to make ? 4
Server address (127.0.0.1):
Service port (50000):
```

Case of Status.code=1

If we register an existing voter name, then return status code 1.



```
D:\FTC-2023-Project-Online-Voting\Part2>python voting_server.py
1. RegisterVoter
2. UnregisterVoter
3. List Voter
4. Leave registration
Which Local Server API would you like to make ? 1
RegisterVoter API Called:
name: "Bob"
group: "students"
public_key: "\330-V\356\267\234\372\033\033\304\254\031U\r\240\332\354\374\337\263\306\335\n\206\314\272u8\245k\252\307"


Status :  code: 0

RegisterVoter API Called:
name: "Bob"
group: "students"
public_key: "\330-V\356\267\234\372\033\033\304\254\031U\r\240\332\354\374\337\263\306\335\n\206\314\272u8\245k\252\307"


Status :  code: 1
```

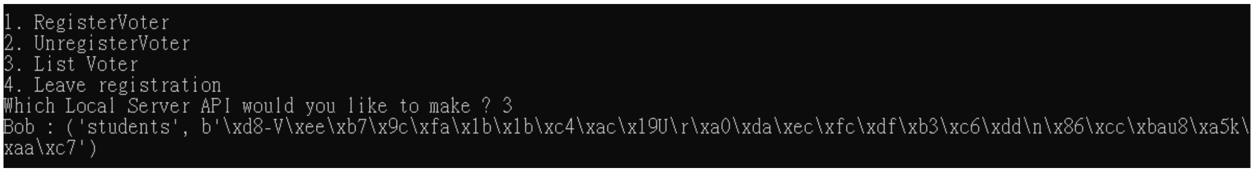
UnregisterVoter(VoterName) returns (Status)

This API is used for unregistering a voter with the name VoterName.

- The server side can unregister a certain voter's name.
- Return Value
  - Status.code=0 : Successful unregistration
  - Status.code=1 : No voter with the name exists on the server
  - Status.code=2 : Undefined error

Case of Status.code=0

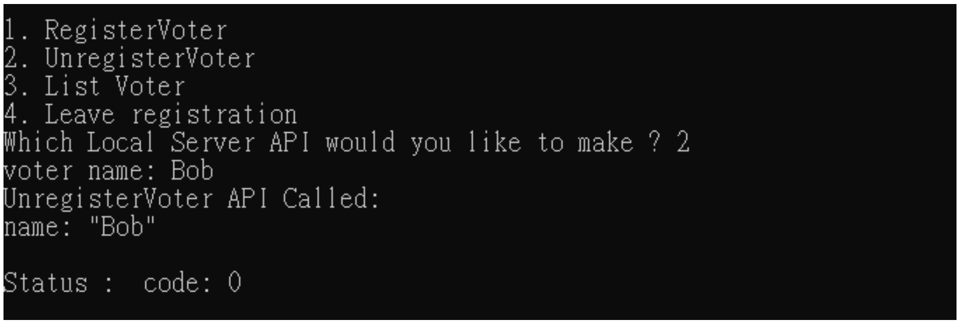
Our registered voters' list currently shows the following.



```
1. RegisterVoter
2. UnregisterVoter
3. List Voter
4. Leave registration
Which Local Server API would you like to make ? 3
Bob : ('students', b'\xd8-V\xee\xb7\x9c\xfa\x1b\x1b\xc4\xac\x19U\r\xa0\xda\xec\xfc\xdf\xb3\xc6\xdd\n\x86\xcc\xbau8\xa5k\
xaa\xc7')
```

First, choose 2 to unregister a certain voter's name.

Then input the registered voter’s name.

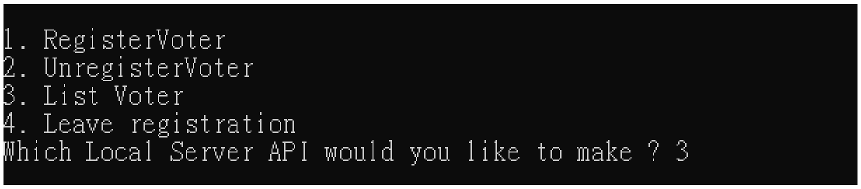


```
1. RegisterVoter
2. UnregisterVoter
3. List Voter
4. Leave registration
Which Local Server API would you like to make ? 2
voter name: Bob
UnregisterVoter API Called:
name: "Bob"

Status :  code: 0
```

Second, you can check the registered voters’ list by choosing 3.

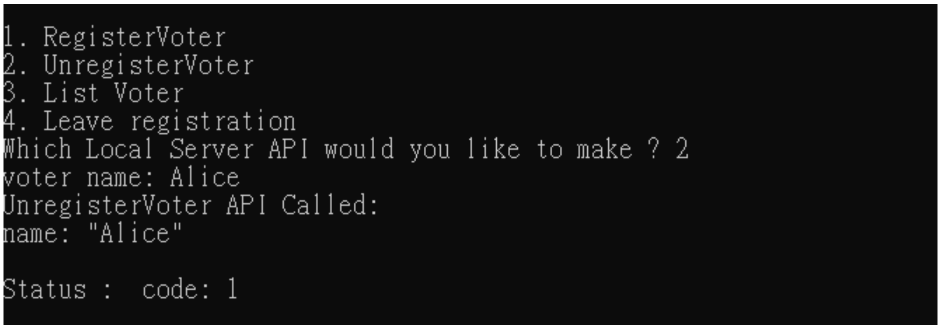
You will see the voter is already unregistered.



```
1. RegisterVoter
2. UnregisterVoter
3. List Voter
4. Leave registration
Which Local Server API would you like to make ? 3
```

Case of Status.code=1

If we unregister a voter that does not exist, then return status code 1.



```
1. RegisterVoter
2. UnregisterVoter
3. List Voter
4. Leave registration
Which Local Server API would you like to make ? 2
voter name: Alice
UnregisterVoter API Called:
name: "Alice"

Status :  code: 1
```

RPC APIs

PreAuth (`VoterName`) returns (`Challenge`)

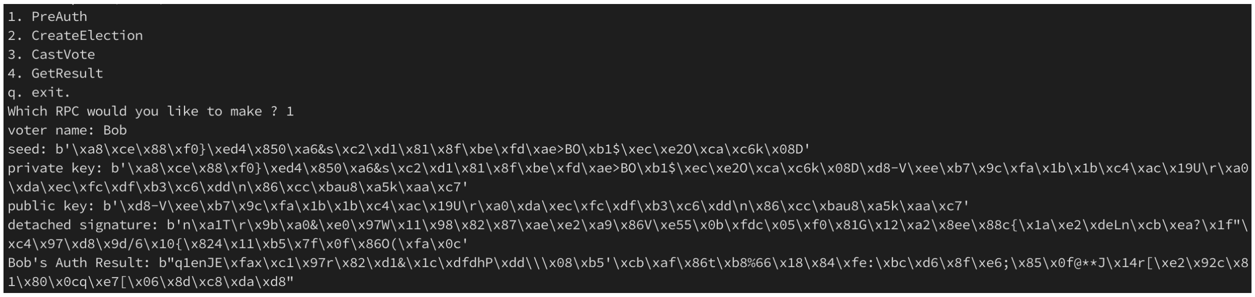
We currently use the user **Bob** for testing, whose ed25519 public key and private key are fixed.

- seed : `b'\xa8\xce\x88\xf0}\xed4\x850\xa6&s\xc2\xd1\x81\x8f\xbe\xfd\xae>BO\xb1$\xec\xe2O\xca\xc6k\x08D’`
- public key : `b'\xd8-V\xee\xb7\x9c\xfa\x1b\x1b\xc4\xac\x19U\r\xa0\xda\xec\xfc\xdf\xb3\xc6\xdd\n\x86\xcc\xbau8\xa5k\xaa\xc7'`
  `detached signature:`
  `b'n\xa1T\r\x9b\xa0&\xe0\x97W\x11\x98\x82\x87\xae\xe2\xa9\x86V\xe55\x0b\xfdc\x05\xf0\x81G\x12\xa2\x8ee\x88c{\x1a\xe2\xdeLn\xcb\xea?`
  `\x1f"\xc4\x97\xd8\x9d/6\x10{\x824\x11\xb5\x7f\x0f\x86O(\xfa\x0c'`

- private key : `b'\xa8\xce\x88\xf0}\xed4\x850\xa6&s\xc2\xd1\x81\x8f\xbe\xfd\xae>BO\xb1$\xec\xe2O\xca\xc6k\x08D\xd8-V\xee\xb7\x9c\xfa\x1b\x1b\xc4\xac\x19U\r\xa0\xda\xec\xfc\xdf\xb3\xc6\xdd\n\x86\xcc\xbau8\xa5k\xaa\xc7’`

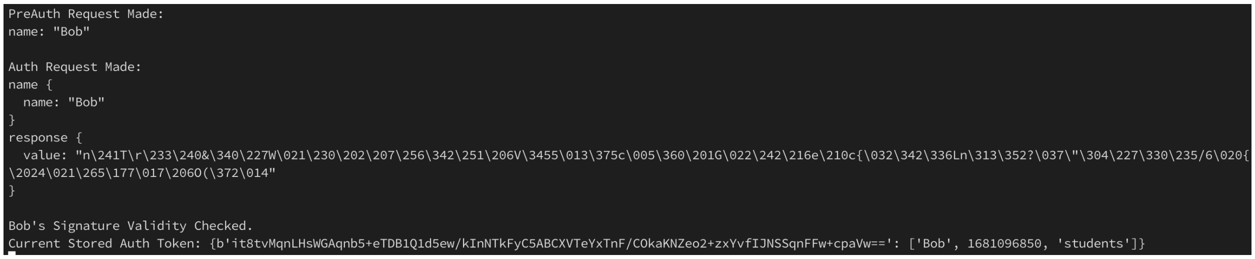
After calling RPC **PreAuth**, server will verify client’s signed signature. If the validity is checked, server will return an auth token to client.

- client side :



```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 1
voter name: Bob
seed: b'\xa8\xce\x88\xf0}\xed4\x850\xa6&s\xc2\xd1\x81\x8f\xbe\xfd\xae>BO\xb1$\xec\xe2O\xca\xc6k\x08D'
private key: b'\xa8\xce\x88\xf0}\xed4\x850\xa6&s\xc2\xd1\x81\x8f\xbe\xfd\xae>BO\xb1$\xec\xe2O\xca\xc6k\x08D\xd8-V\xee\xb7\x9c\xfa\x1b\x1b\xc4\xac\x19U\r\xa0
\xda\xec\xfc\xdf\xb3\xc6\xdd\n\x86\xcc\xbau8\xa5k\xaa\xc7'
public key: b'\xd8-V\xee\xb7\x9c\xfa\x1b\x1b\xc4\xac\x19U\r\xa0\xda\xec\xfc\xdf\xb3\xc6\xdd\n\x86\xcc\xbau8\xa5k\xaa\xc7'
detached signature: b'n\xa1T\r\x9b\xa0&\xe0\x97W\x11\x98\x82\x87\xae\xe2\xa9\x86V\xe55\x0b\xfdc\x05\xf0\x81G\x12\xa2\x8ee\x88c{\x1a\xe2\xdeLn\xcb\xea?\x1f"\
xc4\x97\xd8\x9d/6\x10{\x824\x11\xb5\x7f\x0f\x86O(\xfa\x0c'
Bob's Auth Result: b"q1enJE\xfax\xc1\x97r\x82\xd1&\x1c\xdfdhP\xdd\\\x08\xb5'\xcb\xaf\x86t\xb8%66\x18\x84\xfe:\xbc\xd6\x8f\xe6;\x85\x0f@**J\x14r[\xe2\x92c\x8
1\x80\x0cq\xe7[\x06\x8d\xc8\xda\xd8"
```

- server side :



```
PreAuth Request Made:
name: "Bob"

Auth Request Made:
name {
  name: "Bob"
}
response {
  value: "n\241T\r\233\240&\340\227W\021\230\202\207\256\342\251\206V\3455\013\375c\005\360\201G\022\242\216e\210c{\032\342\336Ln\313\352?\037\"\304\227\330\235/6\020{
\2024\021\265\177\017\206O(\372\014"
}

Bob's Signature Validity Checked.
Current Stored Auth Token: {b'it8tvMqnLHsWGAqnb5+eTDB1Q1d5ew/kInNTkFyC5ABCXVTeYxTnF/COkaKNZeo2+zxYvfIJNSSqnFFw+cpaVw==': ['Bob', 1681096850, 'students']}
```

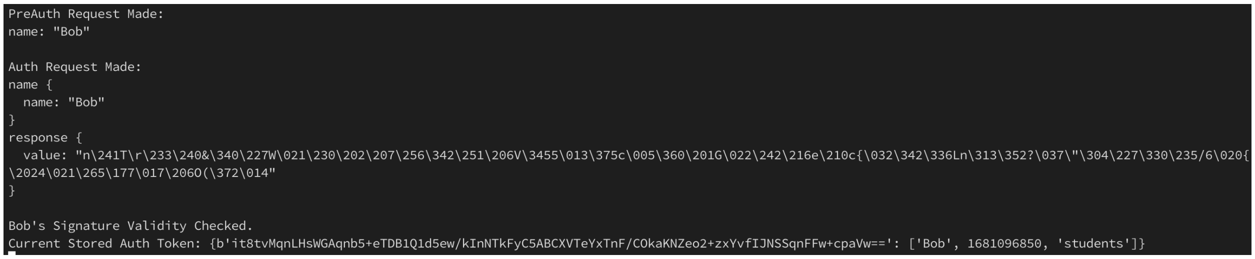
Auth (`AuthRequest`) returns (`AuthToken`)

Right after client received the auth token, client will **automatically** make RPC **Auth** to verify the auth token.

When **Auth** process done, server will store client’s auth token information, such as voter’s name, expired time of auth token, voter’s group, into **auth_tokens** dictionary

- key : client’s auth token
- value : (`Voter.name`, expired time, `Voter.group`)

- server side :



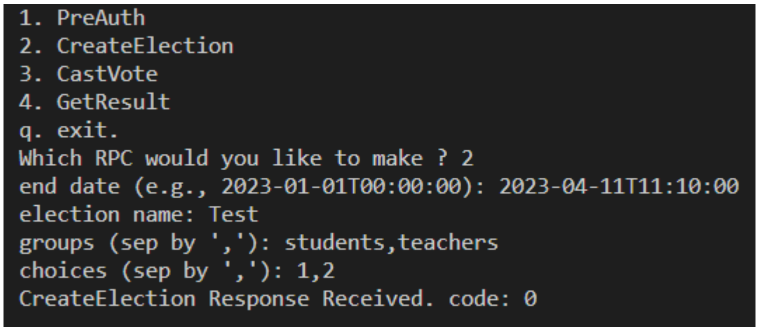
```
PreAuth Request Made:
name: "Bob"

Auth Request Made:
name {
  name: "Bob"
}
response {
  value: "n\241T\r\233\240&\340\227W\021\230\202\207\256\342\251\206V\3455\013\375c\005\360\201G\022\242\216e\210c{\032\342\336Ln\313\352?\037\"\304\227\330\235/6\020{
\2024\021\265\177\017\206O(\372\014"
}

Bob's Signature Validity Checked.
Current Stored Auth Token: {b'it8tvMqnLHsWGAqnb5+eTDB1Q1d5ew/kInNTkFyC5ABCXVTeYxTnF/COkaKNZeo2+zxYvfIJNSSqnFFw+cpaVw==': ['Bob', 1681096850, 'students']}
```

CreateElection (`Election`) returns (`Status`)

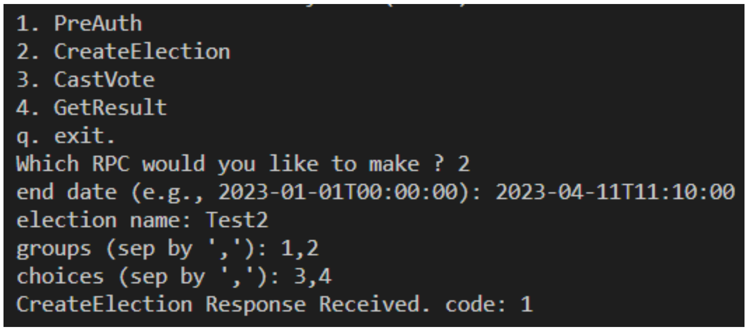
After authicating, one can create an election and determine the attriubutes of the election, such as election name, groups, choices, and end date.



```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 2
end date (e.g., 2023-01-01T00:00:00): 2023-04-11T11:10:00
election name: Test
groups (sep by ','): students,teachers
choices (sep by ','): 1,2
CreateElection Response Received. code: 0
```

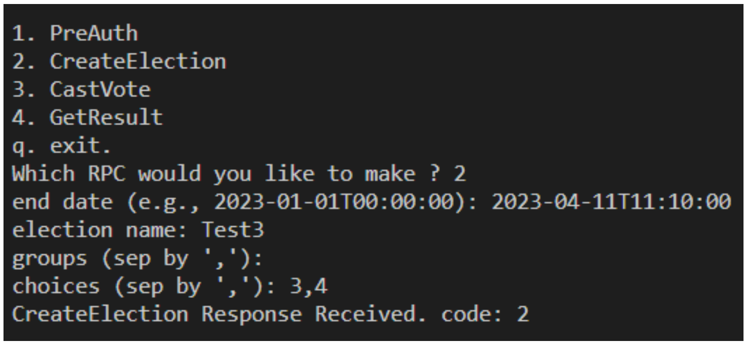
There are serveral error status:

- invalid authentication token



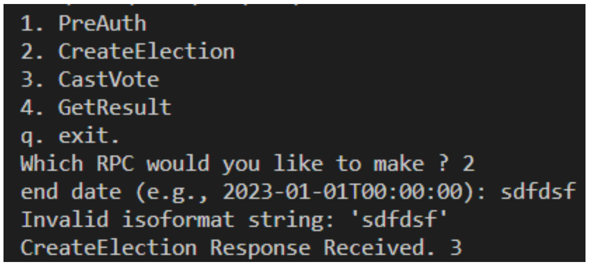
```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 2
end date (e.g., 2023-01-01T00:00:00): 2023-04-11T11:10:00
election name: Test2
groups (sep by ','): 1,2
choices (sep by ','): 3,4
CreateElection Response Received. code: 1
```

- missing groups or choices



```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 2
end date (e.g., 2023-01-01T00:00:00): 2023-04-11T11:10:00
election name: Test3
groups (sep by ','):
choices (sep by ','): 3,4
CreateElection Response Received. code: 2
```

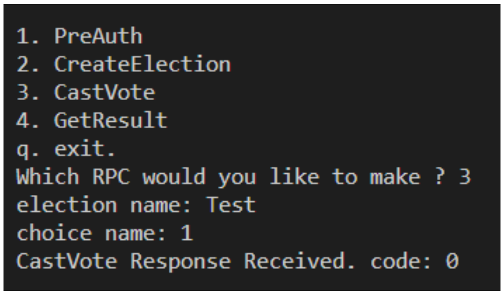
- unknown error



```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 2
end date (e.g., 2023-01-01T00:00:00): sdfdsf
Invalid isoformat string: 'sdfdsf'
CreateElection Response Received. 3
```

CastVote (`Vote`) returns (`Status`)

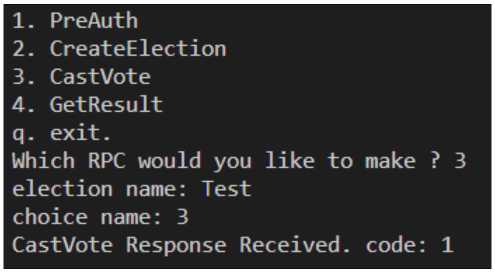
While there is an ongoing election, the authenciated voter can cast the vote to the wanted choice.



```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 3
election name: Test
choice name: 1
CastVote Response Received. code: 0
```

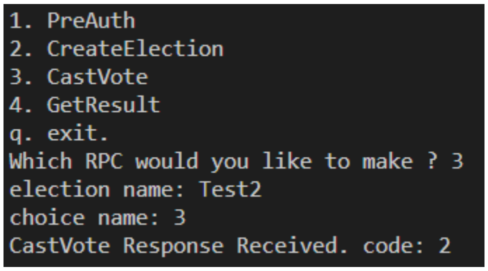
There are serveral error status:

- invalid authentication token



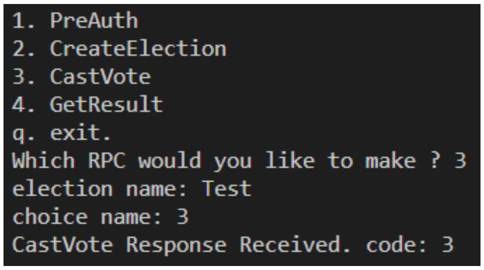
```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 3
election name: Test
choice name: 3
CastVote Response Received. code: 1
```

- invalid election name



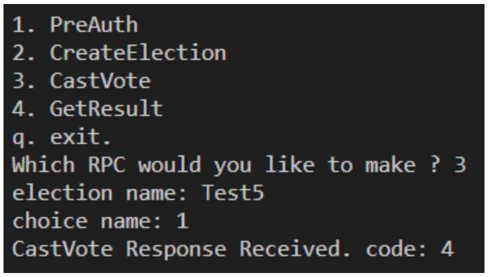
```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 3
election name: Test2
choice name: 3
CastVote Response Received. code: 2
```

- the voter’s group is not allowed



```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 3
election name: Test
choice name: 3
CastVote Response Received. code: 3
```

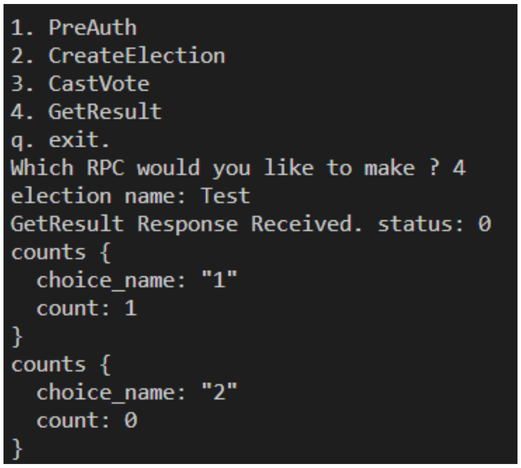
- previous vote has been cast



```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 3
election name: Test5
choice name: 1
CastVote Response Received. code: 4
```

GetResult(`ElectionName`) returns (`ElectionResult`)

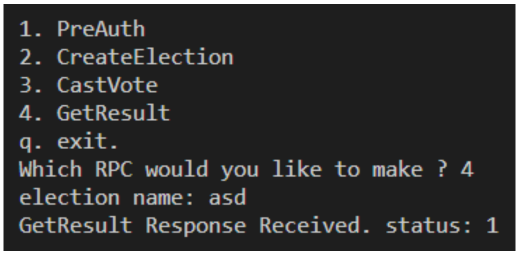
When the end date of the election arrived, one can query the result of the election. Then, the result will print on client's window.



```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 4
election name: Test
GetResult Response Received. status: 0
counts {
  choice_name: "1"
  count: 1
}
counts {
  choice_name: "2"
  count: 0
}
```

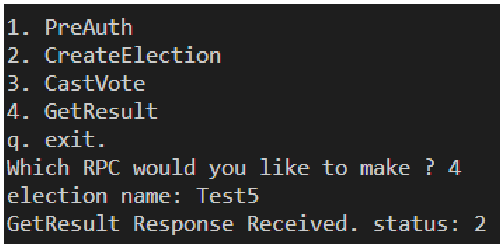
There are serveral status:

- non-existent election



```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 4
election name: asd
GetResult Response Received. status: 1
```

- the election is still ongoing



```
1. PreAuth
2. CreateElection
3. CastVote
4. GetResult
q. exit.
Which RPC would you like to make ? 4
election name: Test5
GetResult Response Received. status: 2
```