

Ballot

Addresses

ballot.sol	0x0Dfd2d99fbfC4E4C31A020F5c53Ab8Dc835b56D8
chairperson	0x3a872f8FED4421E7d5BE5c98Ab5Ea0e0245169A0
address1	0x518F1D434f0B3F7881fdc314cC8c2d0b282b0395
address2	0x51BfB0d94b98428e1bb87c4EDAFbD93366D2Ab71
address3	0x9d6F5e75e9B104b4922257E43bBf78310b3d8C51
address4	0x0aB352d8Fe6544EABa643ac27457841403D3e244

Scenario 1: Basic Voting Rights and Voting

1. [As chairperson, give voting rights to Address1](#)
2. [As Address1, vote for Proposal A](#)
3. Check vote count for Proposal A (should be 1)

Read

proposals

uint256

0

*

Result:

array

[0]: 0x6100

[1]: 1

Read

4. Try to vote again with Address1 (should fail)

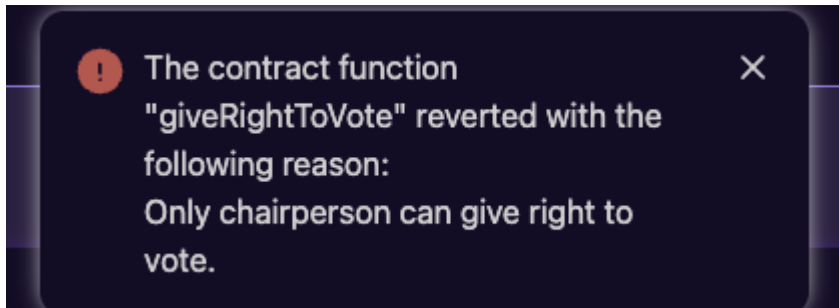
!

The contract function "vote" reverted
with the following reason:
Already voted.

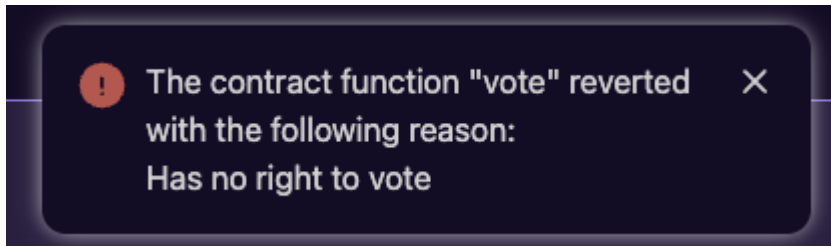
×

Scenario 2: Unauthorized Actions

1. As Address1 (not chairperson), try to give voting rights to Address2 (should fail)

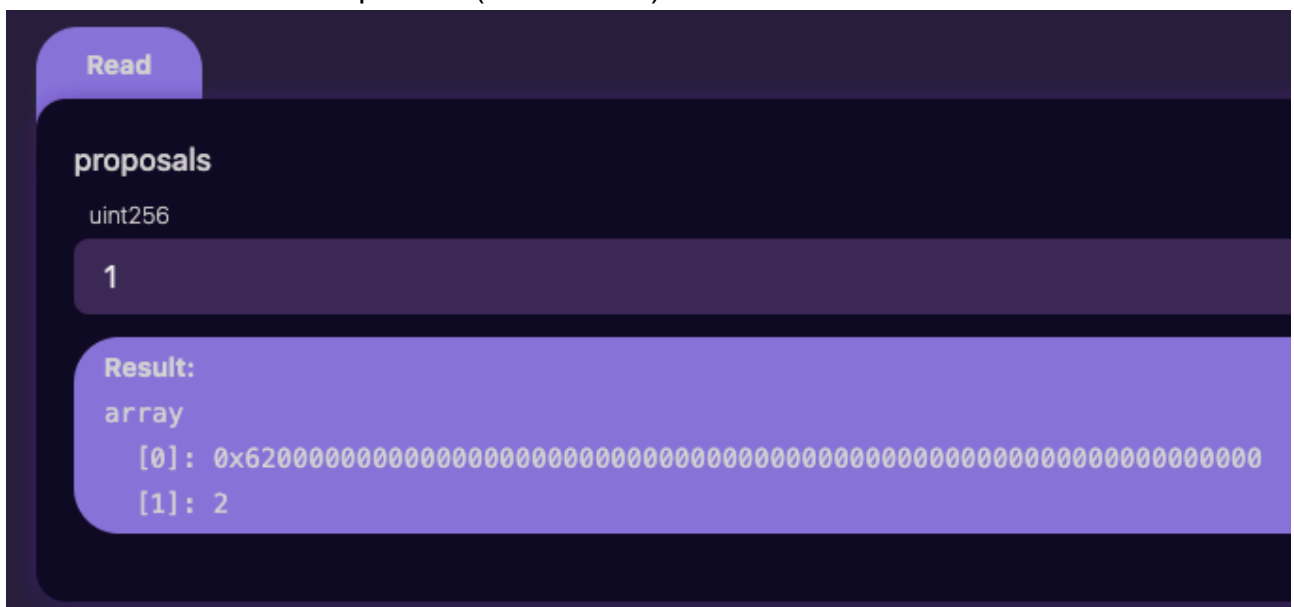


2. As Address2 (without voting rights), try to vote (should fail)



Scenario 3: Delegation

1. As chairperson, give voting rights to [Address2](#) and [Address3](#)
2. [As Address2, delegate vote to Address3](#)
3. [As Address3, vote for Proposal B](#)
4. Check vote count for Proposal B (should be 2)



Scenario 5: Winning Proposal

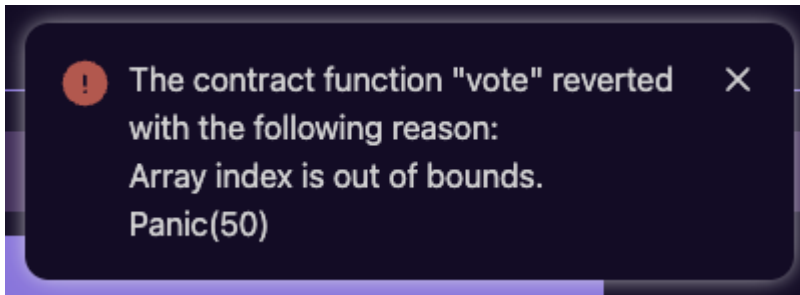
1. Set up a scenario where Proposal B has the most votes
2. Call winningProposal() and verify it returns the index of Proposal B

3. Call `winnerName()` and verify it returns "Proposal B"

```
winnerName  ↻  
0x6200000000000000000000000000000000000000000000000000000000000000  
000  
winningProposal  ↻  
1
```

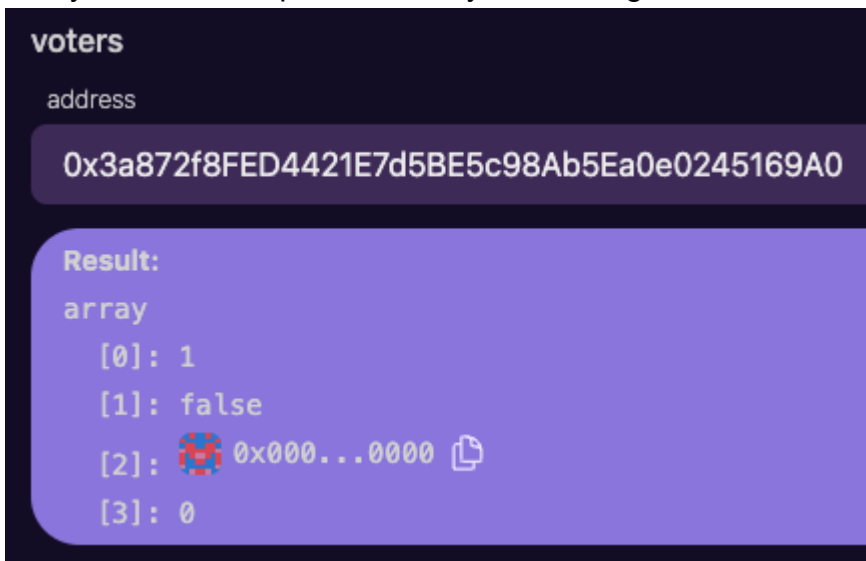
Scenario 6: Out of Bounds Voting

1. Try to vote for a proposal index that doesn't exist (should fail)



Scenario 7: Chairperson Special Case

1. Verify that the chairperson initially has a weight of 1



2. Have the chairperson vote without explicitly calling `giveRightToVote` for themselves

3. Verify that the vote is counted correctly

proposals

uint256

0

Result:

array

[0]: 0x6100

[1]: 2