Weighted Vote Checkpoint

- **Introduced in:** v1.5.0
- **Contract name:** WeightedVoteCheckpoint.sol
- **Type:** Checkpoint Module

How it works

The Weighted Vote Checkpoint module allows the token owner to creates ballot and carry out voting activities. Vote weight is based on the token amount the voter holds at the time of voting.

Key functionalities (as defined in the Smart Contract)

Create ballot & create custom ballot

- createCustomBallot() takes in _startTime, _endTime and _checkPointId to create a ballot based on the total token supply at the _checkPointId
- createBallot() takes only _duration and uses Create Custom Ballot to create a new ballot with start time set to now and create a new check point on the spot.

Active Stats of a Ballot

- setActiveStatsBallot() can be used to set a ballot active stats, require param _isActive different from current stats
- Ballot stats can only be changed if the ballot is not ended yet.
- If a ballot is deemed as invalid, it will not allow voting anymore and the results should be treated as invalid.

Cast Vote

- Takes in a bool value of _vote and _ballotId, it will calculate the vote weight based on the msg.sender's token amount.
- Require ballot to be active at time of voting

Get ballot result

- function getResults(ballotId) can be used to retreive voting results, it will return all weighted and accumulated yes, no and Abstain votes based on voters' token amount, as well as a remaining time of the current ballot.
- If the ballot has already ended, the remaining time will be 0, which will give the final results of the ballot. Otherwise the return results might not be final.