# **1. Voting Builder**

## **101: check VotingParameters structure**

## **102: check VotingParameters default values**

## **103: getVoting() returns 0-address before deploy**

## **104: toggleIsVotersLimited() works properly (on/off)**

## **105: toggleIsVotersLimited() false disables isFunded, isVotePowered, isPrivate**

## **106: toggleIsVoterDisclosed() works properly (on/off)**

## **107: toggleIsVotePowered() works properly (on/off)**

## **108: toggleIsVotePowered() false clears voterPowers[]**

## **109: toggleIsVotePowered() true requires isVotersLimited true**

## **110: toggleIsVotePowered() true requires isPrivate true**

## **111: toggleIsVotePowered() true requires equal length powers-votersPrivate**

## **112: toggleIsVotePowered() true requires power >=1**

## **113: toggleIsResultVetoed() works properly (on/off)**

## **114: toggleIsResultPartial() true if % = 51-100**

## **115: toggleIsResultPartial() false if % = 0-50**

## **116: toggleIsResultPartial() requires % = 0-100**

## **117: toggleIsRepeated() works properly (on/off)**

## **118: toggleIsRepeated() true disables isFunded**

## **119: toggleIsRepeated() true requires isVotersLimited true**

## **120: toggleIsRepeated() true requires votersLimit <= 5000**

## **121: toggleIsPrivate() true requires voters <= 5000**

## **122: toggleIsPrivate() true requires voters <=** **votersLimit if isVotersLimited true**

## **123: toggleIsPrivate() works properly (on/off)**

## **124: toggleIsPrivate() false disables isFunded, isVotePowered**

## **125: toggleIsPrivate() false clears votersPrivate[]**

## **126: toggleIsMultiChoiced() requires choices <= 20**

## **127: toggleIsMultiChoiced() false and sets yes/no if choices <= 1**

## **128: toggleIsMultiChoiced() true and sets input choices if choices 2-20**

## **129: toggleIsFunded() true requires isVotersLimited true**

## **130: toggleIsFunded() true requires isPrivate true**

## **131: toggleIsFunded() true requires usersAmount >= votersLimit**

## **132: toggleIsFunded() true requires usersAmount = votersPrivate**

## **133: toggleIsFunded() works properly (on/off)**

## **134: createVoting() requires choices > 1**

## **135: createVoting() requires msg.value > 0 if isFunded true**

## **136: createVoting() requires votersPrivate > 0 if isFunded true**

## **137: createVoting() funds all votersPrivate if isFunded true**

## **138: createVoting() requires msg.value = 0 if isFunded false**

## **139: createVoting() deploys voting properly**

# **2. Voting**

## **201: check VotingParameters structure is equal to VotingBuilder**

## **202: votingParams accepts params from VotingBuilder properly**

## **203: check VotingResult structure**

## **204: check VotingResult default values**

## **205: votingResult accepts choices from votingParams**

## **206: check Voter structure**

## **207: check Voter default values**

## **208: voters accepts votersPrivate from votingParams**

## **209: voters accepts voterPowers from votingParams if isVotePowered true**

## **210: getVoter() requires isVoterDisclosed true**

## **211: isActive false by default**

## **212: vote() requires onlyActive**

## **213: vote() restricts double-voting**

## **214: vote() requires the selected choice to exist**

## **215: vote() checks the votersLimit reached if isVotersLimited true**

## **216: vote() requires voter to be white-listed if isPrivate true**

## **217: vote() works properly (vote, voted, totalVotes, votesReceived)**

## **218: vote() counts power if isVotePowered true**

## **219: start() requires isActive false**

## **220: start() clears voted, votesReceived, totalVotes, winningChoice, winningVotes if isRepeated true**

## **221: start() requires totalVotes = 0 (not repeated)**

## **222: start() enables voting properly**

## **223: finishVoting() requires onlyActive**

## **224: finishVoting() returns no winner if no votes made**

## **225: finishVoting() returns no winner if two winners**

## **226: finishVoting() returns no winner if result was vetoed**

## **227: finishVoting() returns no winner if no choice reached needed % of votes**

## **228: finishVoting() returns winning votes, choice properly if isResultVetoed true**

## **229: finishVoting() returns winning votes, choice properly if isResultPartial true**

## **230: finishVoting() returns winning votes, choice properly for two choices**

## **231: finishVoting() returns winning votes, choice properly for multiple choices**

## **232: finishVoting() disables voting**

# **3. Role Model in Voting Builder**

## **301: getVotingParams() requires onlyAdmin**

## **302: getVoting() requires onlyAdmin**

## **303: owner was set properly**

## **304: owner is not operator by default**

## **305: owner is not admin by default**

## **306: owner cannot transfer Ownership to himself**

## **307: transferOwnership() requires onlyOwner**

## **308: transferOwnership() works properly**

## **309: toggleIsVotersLimited() requires onlyAdmin**

## **310: toggleIsVoterDisclosed() requires onlyAdmin**

## **311: toggleIsVotePowered() requires onlyAdmin**

## **312: toggleIsResultVetoed() requires onlyAdmin**

## **313: toggleIsResultPartial() requires onlyAdmin**

## **314: toggleIsRepeated() requires onlyAdmin**

## **315: toggleIsPrivate() requires onlyAdmin**

## **316: toggleIsMultiChoiced() requires onlyAdmin**

## **317: toggleIsFunded() requires onlyAdmin**

## **318: grantRoleAdmin() requires onlyOwner**

## **319: grantRoleAdmin() requires address is not admin**

## **320: grantRoleOperator() requires onlyOwner**

## **321: grantRoleOperator() requires address is not operator**

## **322: revokeRoleAdmin() requires onlyOwner**

## **323: revokeRoleAdmin() requires address is admin**

## **324: revokeRoleOperator() requires onlyOwner**

## **325: revokeRoleOperator() requires address is operator**

## **326: createVoting() requires onlyAdmin**

# **4. Role Model in Voting**

## **401: getVotingParams() requires onlyAdmin**

## **402: getVoter() requires onlyOperator**

## **403: owner was set properly**

## **404: getIsActive() requires onlyOperator**

## **405: owner is not admin by default**

## **406: owner is not operator by default**

## **407: owner cannot transfer Ownership to himself**

## **408: transferOwnership() requires onlyOwner**

## **409: transferOwnership() works properly**

## **410: start() requires onlyOperator**

## **411: grantRoleAdmin() requires onlyOwner**

## **412: grantRoleAdmin() requires address is not admin**

## **413: grantRoleOperator() requires onlyOwner**

## **414: grantRoleOperator() requires address is not operator**

## **415: revokeRoleAdmin() requires onlyOwner**

## **416: revokeRoleAdmin() requires address is admin**

## **417: revokeRoleOperator() requires onlyOwner**

## **418: revokeRoleOperator() requires address is operator**

## **419: finishVoting() requires onlyOperator**