# #

Gov

This module provides the basic functionalities forGovernance .

# #

Available Commands

Name Description proposal Query details of a single proposalproposals Query proposals with optional filtervote Query details of a single vote votes Query votes on a proposal deposit Query details of a deposit deposits Query deposits on a proposal tally Get the tally of a proposal voteparam Query the parameters (voting params Query the parameters of the governance process proposer Query which address proposed a proposal with a given ID.draft-proposal Draft any type of proposal submit-proposal Submit a proposal along with an initial depositsubmit-legacy-proposal Submit a legacy proposal along with an initial deposit deposit Deposit tokens for an active proposalvote Vote for an active proposal, options: yes/no/no_with_veto/abstain weighted-vote Submit a weighted vote for a given governance proposal

# #

iris query gov proposal

Query details of a proposal.

iris query gov proposal[proposal-id] [flags]

#

Query a proposal

iris query gov proposal< proposal-id>

# #

iris query gov proposals

Query proposals with optional filter.

iris query gov proposals[flags] Flags:

Name, shorthand Type Required Default Description --depositor Address

Filter proposals by depositor address --limit uint

Limit to the latest [number] of proposals. Default to all proposals --status string

Filter proposals by status --voter Address

Filter proposals by voter address

#

Query all proposals

iris query gov proposals

#

Query proposals by conditions

iris query gov proposals--limit

3 --status = Passed--depositor = < iaa.. .>

# #

iris query gov vote

Query details of a single vote.

iris query gov vote[proposal-id] [voter-addr] [flags]

#

Query a vote

iris query gov vote< proposal-id> < iaa.. .>

#

iris query gov votes

Query votes on a proposal.

iris query gov votes[proposal-id] [flags]

#

Query all votes of a proposal

iris query gov votes< proposal-id>

#

iris query gov deposit

Query details for a single proposal deposit on a proposal by its identifier.

iris query gov deposit[proposal-id] [depositer-addr] [flags]

#

Query a deposit of a proposal

iris query gov deposit< proposal-id> < iaa.. .>

#

iris query gov deposits

Query details for all deposits on a proposal.

iris query gov deposits[proposal-id] [flags]

#

Query all deposits of a proposal

iris query gov deposits< proposal-id>

#

iris query gov tally

Query tally of votes on a proposal. You can find the proposal-id by running "iris query gov proposals".

iris query gov tally[proposal-id] [flags]

#

Query the statistics of a proposal

iris query gov tally< proposal-id>

#

iris query gov param

Query the parameters (voting|tallying|deposit) of the governance process.

iris query gov param[param-type] [flags] Example:

iris query gov param voting> iris query gov param tallying> iris query gov param deposit

#

iris query gov params

Query the all the parameters for the governance process.

iris query gov params[flags]

#

iris query gov proposer

Query which address proposed a proposal with a given ID.

iris query gov proposer[proposal-id] [flags]

#

iris tx gov draft-proposal

The draft-proposal command allows users to draft any type of proposal. The command returns a draft_proposal.json, to be used by submit-proposal after being completed. The draft_metadata.json is meant to be uploaded to IPFS.

iris tx gov draft-proposal

#

iris tx gov submit-proposal

The submit-proposal command allows users to submit a governance proposal along with some messages and metadata. Messages, metadata and deposit are defined in a JSON file.

iris tx gov submit-proposal[path/to/proposal.json] [flags] where proposal.json contains:

{ "messages" : [{ "@type" : "/cosmos.bank.v1beta1.MsgSend" , "from_address" : "iaa1..." , // The gov module module address "to_address" : "iaa1..." , "amount" : [{ "denom" : "stake" , "amount" : "10" }] }] , "metadata" : "AQ==" , "deposit" : "10stake" }

#

iris tx gov submit-legacy-proposal

The submit-legacy-proposal command allows users to submit a governance legacy proposal along with an initial deposit. Proposal title, description, type and deposit can be given directly or through a proposal JSON file. Available Commands:community-pool-spend ,param-change ,software-upgrade ,cancel-software-upgrade ,client-create ,client-upgrade ,relayer-register ,set-rules .

#

iris tx gov submit-legacy-proposal community-pool-spend

Submit a community pool spend proposal along with an initial deposit. The proposal details must be supplied via a JSON file.

iris tx gov submit-legacy-proposal community-pool-spend< path/to/proposal.json> --from = < key_or_address> Where proposal.json contains:

{ "title" : "Community Pool Spend" , "description" : "Pay me some Atoms!" , "recipient" :

"iaa1mjk4p68mmulwla3x5uzlgjwsc3zrms448rel3q" , "amount" : "1000uiris" , "deposit" : "1000uiris" }

#

iris tx gov submit-legacy-proposal param-change

Submit a parameter proposal along with an initial deposit. The proposal details must be supplied via a JSON file. For values that contains objects, only non-empty fields will be updated.

IMPORTANT: Currently parameter changes are evaluated but not validated, so it is very important that any "value" change is valid (ie. correct type and within bounds) for its respective parameter, eg. "MaxValidators" should be an integer and not a decimal.

Proper vetting of a parameter change proposal should prevent this from happening (no deposits should occur during the governance process), but it should be noted regardless.

iris tx gov submit-legacy-proposal param-change< path/to/proposal.json> --from = < key_or_address> Where proposal.json contains:

{ "title" : "Staking Param Change" , "description" : "Update max validators" , "changes" : [{ "subspace" : "staking" , "key" : "MaxValidators" , "value" : 105 }] , "deposit" : "1000uiris" }

#

iris tx gov submit-legacy-proposal software-upgrade

Submit a software upgrade along with an initial deposit. Please specify a unique name and height OR time for the upgrade to take effect.

iris tx gov submit-legacy-proposal software-upgrade[name] (--upgrade-height[height] | --upgrade-time[time]) (--upgrade-info[info]) [flags] Flags:

Name, shorthand Type Required Default Description --deposit Coin Yes

Deposit of the proposal --title string Yes

Title of proposal --description string Yes

Description of proposal --upgrade-height int64

The height at which the upgrade must happen (not to be used together with --upgrade-time) --time string

The time at which the upgrade must happen (not to be used together with --upgrade-height) --info string

Optional info for the planned upgrade such as commit hash, etc.

#

iris tx gov submit-legacy-proposal cancel-software-upgrade

Cancel a software upgrade along with an initial deposit.

iris tx gov submit-legacy-proposal cancel-software-upgrade[flags] Flags:

Name, shorthand Type Required Default Description --deposit Coin Yes

Deposit of the proposal --title string Yes

Title of proposal --description string Yes

Description of proposal

#

iris tx gov submit-legacy-proposal client-create

Submit a client create along with an initial deposit.

iris tx gov submit-legacy-proposal client-create[chain-name] [path/to/client_state.json] [path/to/consensus_state.json] [flags]

#

iris tx gov submit-legacy-proposal client-upgrade

Submit a client upgrade along with an initial deposit.

iris tx gov submit-legacy-proposal client-upgrade[chain-name] [path/to/client_state.json] [path/to/consensus_state.json] [flags]

#

iris tx gov submit-legacy-proposal relayer-register

Submit a relayer register along with an initial deposit.

iris tx gov submit-legacy-proposal relayer-register[chain-name] [relayers-address] [flags]

#

iris tx gov submit-legacy-proposal set-rules

Submit a set rules along with an initial deposit.

iris tx gov submit-legacy-proposal set-rules[path/to/routing_rules.json] [flags]

#

iris tx gov deposit

Submit a deposit for an active proposal. You can find the proposal-id by running "iris query gov proposals".

iris tx gov deposit[proposal-id] [deposit] [flags]

#

Deposit for an active proposal

iris tx gov deposit[proposal-id] [deposit]

#

iris tx gov vote

Submit a vote for an active proposal. You can find the proposal-id by running "iris query gov proposals". Vote for an active proposal, options: yes/no/no_with_veto/abstain.

iris tx gov vote[proposal-id] [option] [flags]

#

Vote for an active proposal

iris tx gov vote< proposal-id> < option> --from = < key-name> --fees = 0 .3iris

#

iris tx gov weighted-vote

The weighted-vote command allows users to submit a weighted vote for a given governance proposal.

iris tx gov weighted-vote[proposal-id] [weighted-options] [flags]