Filecoin.sol

This page covers the built-in actors Filecoin.sol API.

For conceptual information on built-in actors, including their purposes, how they work and more, see the conceptual guide .

Prerequisites

Before you can call a built-in actor using the API, you must import filecoin.solidity using one of the available methods.

Call a built-in actor

For available actors and methods see Available actors and methods .

Once you've either imported particular contracts manually or simply installed filecoin-solidity using npm , create a callable method to access the built-in actor methods the way you normally would in a Solidity smart contract. Working examples of smart contracts that call built-in actor methods are available below.

- Account
- DataCap
- Miner
- Storage market
- Storage power
- Verified registry
-

For conceptual information on built-in actors, including their purposes, how they work and available types, see the conceptual guide .

Call the account actor

The following example imports the Account actor library and creates a callable method for each of the available actor methods . For the full code, see the GitHub repository .

```

Copy pragmasolidity^0.8.17;

import"../types/AccountTypes.sol"; import"../types/CommonTypes.sol"; import"../AccountAPI.sol"; import"../Utils.sol";

contractAccountApiTest{ functionauthenticate_message(CommonTypes.FilActorIdtarget,AccountTypes.AuthenticateMessageParamsmemoryparams)public{
AccountAPI.authenticateMessage(target,params); }

functionuniversal_receiver_hook(CommonTypes.FilActorIdtarget,CommonTypes.UniversalReceiverParamsmemoryparams)public{
Utils.universalReceiverHook(target,params); } }

```

Call the DataCap actor

The following example imports the DataCap actor library and creates a callable method for each of the available actor methods . For the full code, see the GitHub repository .

```

Copy pragmasolidity^0.8.17;

import"../types/DataCapTypes.sol"; import"../types/CommonTypes.sol"; import"../cbor/BigIntCbor.sol"; import"../DataCapAPI.sol"; import"../Utils.sol";

contractDataCapApiTest{ functionname()publicreturns(stringmemory) { returnDataCapAPI.name(); }

functionsymbol()publicreturns(stringmemory) { returnDataCapAPI.symbol(); }

functiontotal_supply()publicreturns(CommonTypes.BigIntmemory) { returnDataCapAPI.totalSupply(); }

functionbalance(CommonTypes.FilAddressmemoryaddr)publicreturns(CommonTypes.BigIntmemory) { returnDataCapAPI.balance(addr); }

functionallowance(DataCapTypes.GetAllowanceParamsmemoryparams)publicreturns(CommonTypes.BigIntmemory) { returnDataCapAPI.allowance(params); }

functiontransfer(DataCapTypes.TransferParamsmemoryparams)publicreturns(DataCapTypes.TransferReturnmemory) { returnDataCapAPI.transfer(params); }

functiontransfer_from(DataCapTypes.TransferFromParamsmemoryparams)publicreturns(DataCapTypes.TransferFromReturnmemory) {
returnDataCapAPI.transferFrom(params); }

functionincrease_allowance(DataCapTypes.IncreaseAllowanceParamsmemoryparams)publicreturns(CommonTypes.BigIntmemory) {
returnDataCapAPI.increaseAllowance(params); }

functiondecrease_allowance(DataCapTypes.DecreaseAllowanceParamsmemoryparams)publicreturns(CommonTypes.BigIntmemory) {
returnDataCapAPI.decreaseAllowance(params); }

functionrevoke_allowance(CommonTypes.FilAddressmemoryoperator)publicreturns(CommonTypes.BigIntmemory) {
returnDataCapAPI.revokeAllowance(operator); }

functionburn(CommonTypes.BigIntmemoryamount)publicreturns(CommonTypes.BigIntmemory) { returnDataCapAPI.burn(amount); }

functionburn_from(DataCapTypes.BurnFromParamsmemoryparams)publicreturns(DataCapTypes.BurnFromReturnmemory) {
returnDataCapAPI.burnFrom(params); }

functionhandle_filecoin_method(uint64method,uint64codec,bytescalldataparams)publicpure{ Utils.handleFilecoinMethod(method,codec,params); } }

```

Call the storage market actor

The following example imports the Storage market actor library and creates a callable method for each of the available actor methods . For the full code, see the

GitHub repository .

```

Copy pragmasolidity^0.8.17;

import"../MarketAPI.sol"; import"../types/MarketTypes.sol";

contractMarketApiTest{ functionadd_balance(CommonTypes.FilAddressmemoryproviderOrClient,uint256value)publicpayable{
MarketAPI.addBalance(providerOrClient,value); }

functionwithdraw_balance(MarketTypes.WithdrawBalanceParamsmemoryparams)publicreturns(CommonTypes.BigIntmemory) {
returnMarketAPI.withdrawBalance(params); }

functionget_balance(CommonTypes.FilAddressmemoryaddr)publicreturns(MarketTypes.GetBalanceReturnmemory) { returnMarketAPI.getBalance(addr); }

functionget_deal_data_commitment(uint64dealID)publicreturns(MarketTypes.GetDealDataCommitmentReturnmemory) {
returnMarketAPI.getDealDataCommitment(dealID); }

functionget_deal_client(uint64dealID)publicreturns(uint64) { returnMarketAPI.getDealClient(dealID); }

functionget_deal_provider(uint64dealID)publicreturns(uint64) { returnMarketAPI.getDealProvider(dealID); }

functionget_deal_label(uint64dealID)publicreturns(stringmemory) { returnMarketAPI.getDealLabel(dealID); }

functionget_deal_term(uint64dealID)publicreturns(MarketTypes.GetDealTermReturnmemory) { returnMarketAPI.getDealTerm(dealID); }

functionget_deal_total_price(uint64dealID)publicreturns(CommonTypes.BigIntmemory) { returnMarketAPI.getDealTotalPrice(dealID); }

functionget_deal_client_collateral(uint64dealID)publicreturns(CommonTypes.BigIntmemory) { returnMarketAPI.getDealClientCollateral(dealID); }

functionget_deal_provider_collateral(uint64dealID)publicreturns(CommonTypes.BigIntmemory) { returnMarketAPI.getDealProviderCollateral(dealID); }

functionget_deal_verified(uint64dealID)publicreturns(bool) { returnMarketAPI.getDealVerified(dealID); }

functionget_deal_activation(uint64dealID)publicreturns(MarketTypes.GetDealActivationReturnmemory) { returnMarketAPI.getDealActivation(dealID); }

functionpublish_storage_deals(MarketTypes.PublishStorageDealsParamsmemoryparams)publicreturns(MarketTypes.PublishStorageDealsReturnmemory) {
returnMarketAPI.publishStorageDeals(params); } }

```

Call the miner actor

The following example imports the Account actor library and creates a callable method for each of theavailable actor methods . For the full code, seethe GitHub repository .

```

Copy pragmasolidity^0.8.17;

import"../MinerAPI.sol"; import"../types/MinerTypes.sol";

contractMinerApiTest{ functionget_owner(CommonTypes.FilActorIdtarget)publicreturns(MinerTypes.GetOwnerReturnmemory) { returnMinerAPI.getOwner(target);
}

functionchange_owner_address(CommonTypes.FilActorIdtarget,CommonTypes.FilAddressmemoryaddr)public{ MinerAPI.changeOwnerAddress(target,addr); }

functionis_controlling_address(CommonTypes.FilActorIdtarget,CommonTypes.FilAddressmemoryaddr)publicreturns(bool) {
returnMinerAPI.isControllingAddress(target,addr); }

functionget_sector_size(CommonTypes.FilActorIdtarget)publicreturns(uint64) { returnMinerAPI.getSectorSize(target); }

functionget_available_balance(CommonTypes.FilActorIdtarget)publicreturns(CommonTypes.BigIntmemory) { returnMinerAPI.getAvailableBalance(target); }

functionget_vesting_funds(CommonTypes.FilActorIdtarget)publicreturns(MinerTypes.GetVestingFundsReturnmemory) { returnMinerAPI.getVestingFunds(target);
}

functionchange_beneficiary(CommonTypes.FilActorIdtarget,MinerTypes.ChangeBeneficiaryParamsmemoryparams)public{
returnMinerAPI.changeBeneficiary(target,params); }

functionget_beneficiary(CommonTypes.FilActorIdtarget)publicreturns(MinerTypes.GetBeneficiaryReturnmemory) { returnMinerAPI.getBeneficiary(target); }

functionchange_worker_address(CommonTypes.FilActorIdtarget,MinerTypes.ChangeWorkerAddressParamsmemoryparams)public{
MinerAPI.changeWorkerAddress(target,params); }

functionchange_peer_id(CommonTypes.FilActorIdtarget,CommonTypes.FilAddressmemorynewId)public{ MinerAPI.changePeerId(target,newId); }

functionchange_multiaddresses(CommonTypes.FilActorIdtarget,MinerTypes.ChangeMultiaddrsParamsmemoryparams)public{
MinerAPI.changeMultiaddresses(target,params); }

functionrepay_debt(CommonTypes.FilActorIdtarget)public{ MinerAPI.repayDebt(target); }

functionconfirm_change_worker_address(CommonTypes.FilActorIdtarget)public{ MinerAPI.confirmChangeWorkerAddress(target); }

functionget_peer_id(CommonTypes.FilActorIdtarget)publicreturns(CommonTypes.FilAddressmemory) { returnMinerAPI.getPeerId(target); }

functionget_multiaddresses(CommonTypes.FilActorIdtarget)publicreturns(MinerTypes.GetMultiaddrsReturnmemory) { returnMinerAPI.getMultiaddresses(target); }

functionwithdraw_balance(CommonTypes.FilActorIdtarget,CommonTypes.BigIntmemoryamount)publicreturns(CommonTypes.BigIntmemory) {
returnMinerAPI.withdrawBalance(target,amount); } }

```

Call the storage power actor

The following example imports the Storage power actor library and creates a callable method for each of the available actor methods . For the full code, see the GitHub repository .

```

Copy pragmasolidity^0.8.17;

import"../types/PowerTypes.sol"; import"../types/CommonTypes.sol"; import"../PowerAPI.sol";

contractPowerApiTest{
functioncreate_miner(PowerTypes.CreateMinerParamsmemoryparams,uint256value)publicpayablereturns(PowerTypes.CreateMinerReturnmemory) {
returnPowerAPI.createMiner(params,value); }

functionminer_count()publicreturns(uint64) { returnPowerAPI.minerCount(); }

functionminer_consensus_count()publicreturns(int64) { returnPowerAPI.minerConsensusCount(); }

functionnetwork_raw_power()publicreturns(CommonTypes.BigIntmemory) { returnPowerAPI.networkRawPower(); }

functionminer_raw_power(uint64minerID)publicreturns(PowerTypes.MinerRawPowerReturnmemory) { returnPowerAPI.minerRawPower(minerID); } }

```

Call the verified registry actor

The following example imports the verified registry actor library and creates a callable method for each of the available actor methods . For the full code, see the GitHub repository .

```

Copy pragmasolidity^0.8.17;

import"../types/VerifRegTypes.sol"; import"../types/CommonTypes.sol"; import"../VerifRegAPI.sol";

contractVerifRegApiTest{ functionget_claims(VerifRegTypes.GetClaimsParamsmemoryparams)publicreturns(VerifRegTypes.GetClaimsReturnmemory) {
returnVerifRegAPI.getClaims(params); }

functionadd_verified_client(VerifRegTypes.AddVerifiedClientParamsmemoryparams)public{ VerifRegAPI.addVerifiedClient(params); }

functionremove_expired_allocations( VerifRegTypes.RemoveExpiredAllocationsParamsmemoryparams
)publicreturns(VerifRegTypes.RemoveExpiredAllocationsReturnmemory) { returnVerifRegAPI.removeExpiredAllocations(params); }

functionextend_claim_terms(VerifRegTypes.ExtendClaimTermsParamsmemoryparams)publicreturns(CommonTypes.BatchReturnmemory) {
returnVerifRegAPI.extendClaimTerms(params); }

functionremove_expired_claims(VerifRegTypes.RemoveExpiredClaimsParamsmemoryparams)publicreturns(VerifRegTypes.RemoveExpiredClaimsReturnmemory)
{ returnVerifRegAPI.removeExpiredClaims(params); } }

```

Previous Protocol API Next JSON-RPC

Last updated25 days ago