Klaytn OpenSDK Specification

1. Introduction

* Developing a blockchain-based application requires careful consideration of many technical aspects, including the creation of a Remote Procedure Call (RPC) specification and Software Development Kit (SDK). These components play a critical role in ensuring that your application is well-structured, scalable, and interoperable with other systems. This document describes the Klaytn RPC Specification and how to write it by OpenAPI Specification for documents by redocly and the SDK for each language by the code generator.
* OpenAPI known as [OAS 3.0](https://swagger.io/specification/) that is upgraded from Swagger 2.0 is a specification for language-agnostic interface to HTTP API. OpenAPI also provides a tool, OpenAPI Generator, from the HTTP API Specification to the source code. You can find the supported language list here([openAPI generator resources](https://github.com/OpenAPITools/openapi-generator/tree/master/modules/openapi-generator/src/main/resources)). The generator takes a [mustache](https://www.tsmean.com/articles/mustache/the-ultimate-mustache-tutorial/) and a yaml file as input and returns the generated source code as output. In the mustache file, there is a template code with placeholder and the generator uses the data in the yaml file to replace it. The placeholder can be used for generating similar functions iteratively. You can see an example in detail in the Implementation section.

1. Workflow and schedule

* There are 3 steps to achieve for developing the OpenSDK
  + Step0: the entire development process for only a few APIs for feedback
    - **LiteOpenSDK**
  + Step1: writing klaytn RPC specification by OAS 3.0
    - **OpenSDK.yaml**
  + Step2: code generation by OpenAPI generator and writing API test code
    - OpenSDK **source code**
    - **Test code** for each API
  + Step3: mustache file tuning to adjust function input parameters
    - **Changed function parameters** to accept specific variables rather than all json request bodies
  + Final : Summary
* Output in detail
  + Example repository
    - <https://github.com/kjeom/klaytn-open-sdk>
  + OpenSDK.yaml
    - write files in api/components and api/paths/klay directory(possible 1 file)
    - get a **klay.yaml** with $ yarn build at api/redocly directory
    - also run docs server with $yarn start
  + OpenSDK source code and test code
    - get openapi directory with $sdk/client/python/python-generate.sh
    - write test(sdk/client/python/openapi-test/test/klay/**klayApiGetRewardsTest.py**)
  + Modified handlebars template files
    - Handlebars template directory (sdk/client/python/template/…)
    - Modified test code
* Schedule

| PoC | Write OpenSDK Spec | | | | Code Gen. | | Mustache tuning | | | | final |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1w | 2w | 3w | 4w | 5w | 6w | 7w | 8w | 9w | 10w | 11w | 12w |
|  | klay | eth  personal  net  txpool  rpc | debug  governance | Admin  Mainbridge  subbridge |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | klay | Eth  Personal  Net  Txpool  rpc | Debug  governance | Admin  Mainbridge  subbridge |  |

1. RPC Specification

* It covers only **eth/klay/personal/net/rpc/admin/debug/governance/mainbridge/subbridge** namespace
* eth namespace
* reference link (<https://docs.klaytn.foundation/content/dapp/json-rpc/api-references/eth>)

| RPC Name | eth\_accounts | |
| --- | --- | --- |
| Input/Output | // Request   | curl -H "Content-Type: application/json" --data  '{  "jsonrpc":"2.0",  "method":"eth\_accounts",  "id":1,  "params":[]  }' http://localhost:8551 | | --- |   // Result   | { "jsonrpc": "2.0", "id":1, "result": ["0xc94770007dda54cF92009BFF0dE90c06F603a09f"] } | | --- | | |

| RPC Name | eth\_accounts  eth\_blockNumber  eth\_call  eth\_chainId  eth\_coinbase  eth\_createAccessList  eth\_estimateGas  eth\_feeHistory  eth\_fillTransaction  eth\_gasPrice  eth\_getBalance  eth\_getBlockByHash  eth\_getBlockByNumber  eth\_getBlockTransactionCountByHash  eth\_getBlockTransactionCountByNumber  eth\_getCode  eth\_getFilterChanges  eth\_getFilterLogs  eth\_getHeaderByHash  eth\_getHeaderByNumber  eth\_getProof  eth\_getRawTransactionByBlockHashAndIndex  eth\_getRawTransactionByBlockNumberAndIndex  eth\_getRawTransactionByHash  eth\_getStorageAt  eth\_getTransactionByBlockHashAndIndex  eth\_getTransactionByBlockNumberAndIndex  eth\_getTransactionByHash  eth\_getTransactionCount  eth\_getTransactionReceipt  eth\_getUncleByBlockHashAndIndex  eth\_getUncleByBlockNumberAndIndex  eth\_getUncleCountByBlockHash  eth\_getUncleCountByBlockNumber  eth\_getWork  eth\_hashrate  eth\_maxPriorityFeePerGas  eth\_mining  eth\_newBlockFilter  eth\_newFilter  eth\_newPendingTransactionFilter  eth\_pendingTransactions  eth\_protocolVersion  eth\_resend  eth\_sendRawTransaction  eth\_sendTransaction  eth\_sign  eth\_signTransaction  eth\_submitWork  eth\_syncing  eth\_uninstallFilter |
| --- | --- |

* klay namespace
* reference link (<https://docs.klaytn.foundation/content/dapp/json-rpc/api-references/klay>)

| RPC Name | klay\_accountCreated | |
| --- | --- | --- |
| Input/Output | // Request   | curl -H "Content-Type: application/json" --data  '{  "jsonrpc":"2.0",  "method":"klay\_accountCreated",  "id":1,  "params":[  "0xa4f42d4d2a3a13874406435500950c9bf2d783db",  "latest"  ]  }' https://api.baobab.klaytn.net:8651 | | --- |   // Result   | { "jsonrpc":"2.0", "id":1, "result":true } | | --- | | |

| RPC Name | klay\_accountCreated  klay\_accounts  klay\_blockNumber  klay\_call  klay\_chainConfig  klay\_chainConfigAt  klay\_chainId / klay\_chainID  klay\_clientVersion  klay\_clientVersion  klay\_createAccessList  klay\_decodeAccountKey  klay\_encodeAccountKey  klay\_estimateComputationCost  klay\_estimateGas  klay\_feeHistory  klay\_gasPrice  klay\_gasPriceAt  klay\_getAccount  klay\_getAccountKey  klay\_getBalance  klay\_getBlockByHash  klay\_getBlockByNumber  klay\_getBlockReceipts  klay\_getBlockTransactionCountByHash  klay\_getBlockTransactionCountByNumber  klay\_getBlockWithConsensusInfoByHash  klay\_getBlockWithConsensusInfoByNumber  klay\_getBlockWithConsensusInfoByNumberRange  klay\_getCode  klay\_getCommittee  klay\_getCommitteeSize  klay\_getCouncil  klay\_getCouncilSize  klay\_getCypressCredit  klay\_getDecodedAnchoringTransactionByHash  klay\_getFilterChanges  klay\_getFilterLogs  klay\_getHeaderByHash  klay\_getHeaderByNumber  klay\_getRawTransactionByBlockHashAndIndex  klay\_getRawTransactionByBlockNumberAndIndex  klay\_getRawTransactionByHash  klay\_getRewards  klay\_getStakingInfo  klay\_getStorageAt  klay\_getTransactionByBlockHashAndIndex  klay\_getTransactionByBlockNumberAndIndex  klay\_getTransactionByHash  klay\_getTransactionBySenderTxHash  klay\_getTransactionCount  klay\_getTransactionReceipt  klay\_getTransactionReceiptBySenderTxHash  klay\_govParamsAt  klay\_isContractAccount  klay\_isParallelDBWrite  klay\_isSenderTxHashIndexingEnabled  klay\_lowerBoundGasPrice  klay\_maxPriorityFeePerGas  klay\_newBlockFilter  klay\_newFilter  klay\_newPendingTransactionFilter  klay\_nodeAddress  klay\_pendingTransactions  klay\_protocolVersion  klay\_resend  klay\_rewardbase  klay\_sendRawTransaction  klay\_sendTransaction  klay\_sendTransactionAsFeePayer  klay\_sha3  klay\_sign  klay\_signTransaction  klay\_signTransactionAsFeePayer  klay\_submitTransaction  klay\_syncing  klay\_uninstallFilter  klay\_upperBoundGasPrice |
| --- | --- |

* net namespace
* reference link (<https://docs.klaytn.foundation/content/dapp/json-rpc/api-references/network>)

| RPC Name | net\_peerCountByType |
| --- | --- |
| Input/Output | // Request curl -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"net\_peerCountByType","params":[],"id":74}' https://api.baobab.klaytn.net:8651 // Result { "id":74, "jsonrpc": "2.0", "result": {"en":3,"pn":2,"total":5} } |

| RPC Name | net\_peerCountByType  net\_version  net\_networkID  net\_listening  net\_peerCount |
| --- | --- |

* txpool namespace
* reference link(<https://docs.klaytn.foundation/content/dapp/json-rpc/api-references/txpool>)

| RPC Name | txpool\_content |
| --- | --- |
| Input/Output | // Request  curl -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"txpool\_content","id":1}' https://api.baobab.klaytn.net:8651 // Result  {"jsonrpc":"2.0","id":1,"result":{"pending":{},"queued":{}}} |

| RPC Name | txpool\_content  txpool\_inspect  txpool\_status |
| --- | --- |

* personal namespace
* reference link (<https://docs.klaytn.foundation/content/dapp/json-rpc/api-references/personal>)

| RPC Name | personal\_deriveAccount |
| --- | --- |
| Input/Output | // Request curl -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"personal\_deriveAccount","params":[url, path, pin],"id":1}' https://api.baobab.klaytn.net:8651 // Result {"jsonrpc":"2.0","id":1,"result":"0xed1b12248aee85a32aead06c7789d3fcdcd4dae6"} |

| RPC Name | personal\_deriveAccount  personal\_ecRecover  personal\_ecRecover  personal\_importRawKey  personal\_importRawKey  personal\_listAccounts  personal\_listWallets  personal\_lockAccount  personal\_newAccount  personal\_openWallet  personal\_replaceRawKey  personal\_sendAccountUpdate  personal\_sendTransaction  personal\_sendValueTransfer  personal\_sign  personal\_sign  personal\_signTransaction  personal\_unlockAccount |
| --- | --- |

* governance namespace
* console api specification
* reference link (<https://docs.klaytn.foundation/content/dapp/json-rpc/api-references/governance>)

| RPC Name | governance\_vote |
| --- | --- |
| Input/Output | // call  governance.vote ("governance.governancemode", "ballot")  // result  "Your vote was successfully placed." |

| RPC Name | governance\_vote  governance\_itemsAt  governance\_itemCacheFromDb  governance\_getStakingInfo  governance\_chainConfigAt  governance\_showTally  governance\_totalVotingPower  governance\_myVotes  governance\_myVotingPower  governance\_chainConfig  governance\_nodeAddress  governance\_pendingChanges  governance\_votes  governance\_idxCache  governance\_idxCacheFromDb |
| --- | --- |

* debug namespace
* reference link (<https://docs.klaytn.foundation/content/dapp/json-rpc/api-references/debug>)

| RPC Name | debug\_dumpBlock |
| --- | --- |
| Input/Output | // Request  $ curl -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"debug\_dumpBlock","params":["0x80"],"id":1}' https://api.baobab.klaytn.net:8651 // Result  {"jsonrpc":"2.0","id":1,"result":{"root":"70383c826d1161ec2f12d799023317d8da7775dd47b8502d2d7ef646d094d3a5","accounts":{"0000000000000000000000000000000000000035":{"balance":"12800000000000000000","nonce":0,"root":"56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421","codeHash":"62b00472fac99d94ccc52f5addac43d54c129cd2c6d2357c9557abea67efdec5","code":"6080…bcd73e40029","storage":{}},...(skipped)...}}} |

| RPC Name | debug\_dumpBlock  debug\_dumpStateTrie  debug\_getBlockRlp  debug\_getModifiedAccountsByNumber  debug\_getModifiedAccountsByHash  debug\_getModifiedStorageNodesByNumber  debug\_getBadBlocks  debug\_printBlock  debug\_setHead  debug\_startWarmUp  debug\_startContractWarmUp  debug\_stopWarmUp  debug\_startCollectingTrieStats  debug\_chaindbProperty  debug\_chaindbCompact  debug\_metrics  debug\_verbosity  debug\_verbosityByName  debug\_verbosityByID  debug\_vmodule  debug\_backtraceAt  debug\_stacks  debug\_freeOSMemory  debug\_setGCPercent  debug\_memStats  debug\_gcStats  debug\_startPProf  debug\_stopPProf  debug\_isPProfRunning  debug\_cpuProfile  debug\_startCPUProfile  debug\_stopCPUProfile  debug\_goTrace  debug\_startGoTrace  debug\_stopGoTrace  debug\_blockProfile  debug\_setBlockProfileRate  debug\_writeBlockProfile  debug\_mutexProfile  debug\_setMutexProfileRate  debug\_writeMutexProfile  debug\_writeMemProfile  debug\_traceBlock  debug\_traceBlockFromFile  debug\_traceBadBlock  debug\_standardTraceBadBlockToFile  debug\_standardTraceBlockToFile  debug\_traceBlockByNumber  debug\_traceBlockByNumberRange  debug\_traceBlockByHash  debug\_traceTransaction  debug\_preimage  debug\_storageRangeAt  debug\_setVMLogTarget |
| --- | --- |

* admin namespace
* reference link (<https://docs.klaytn.foundation/content/dapp/json-rpc/api-references/admin>)

| RPC Name | admin\_addPeer |
| --- | --- |
| Input/Output | // Request  $ curl -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"admin\_startHTTP","id":1, "params":["127.0.0.1", 8551, "", "klay"]}' https://api.baobab.klaytn.net:8651 // Result  {"jsonrpc":"2.0","id":1,"error":{"code":-32000,"message":"HTTP RPC already running on 127.0.0.1:8551"}} |

| RPC Name | admin\_addPeer  admin\_removePeer  admin\_exportChain  admin\_importChain  admin\_importChainFromString  admin\_sleepBlocks  admin\_startHTTP  admin\_stopHTTP  admin\_startRPC  admin\_stopRPC  admin\_startWS  admin\_stopWS  admin\_startStateMigration  admin\_stopStateMigration  admin\_saveTrieNodeCacheToDisk  admin\_setMaxSubscriptionPerWSConn  admin\_startSpamThrottler  admin\_stopSpamThrottler  admin\_setSpamThrottlerWhiteList  admin\_getSpamThrottlerWhiteList  admin\_getSpamThrottlerThrottleList  admin\_getSpamThrottlerCandidateList  admin\_nodeInfo  admin\_peers  admin\_datadir  admin\_stateMigrationStatus  admin\_spamThrottlerConfig |
| --- | --- |

* mainbridge/subbridge namespace
* reference link
  + <https://ko.docs.klaytn.foundation/content/dapp/json-rpc/servicechain/mainbridge>
  + <https://ko.docs.klaytn.foundation/content/dapp/json-rpc/servicechain/subbridge>

| RPC Name | mainbridge\_getChildChainIndexingEnabled |
| --- | --- |
| Input/Output | // Request  $ curl -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"mainbridge\_convertChildChainBlockHashToParentChainTxHash","params":["0xeadc6a3a29a20c13824b5df1ba05cca1ed248d046382a4f2792aac8a6e0d1880"],"id":1}' https://api.baobab.klaytn.net:8651  // Result  {"jsonrpc":"2.0","id":1,"result":"0x9a68591c0faa138707a90a7506840c562328aeb7621ac0561467c371b0322d51"} |

| RPC Name | mainbridge\_getChildChainIndexingEnabled  mainbridge\_convertChildChainBlockHashToParentChainTxHash  mainbridge\_nodeInfo  mainbridge\_peers  subbridge\_addPeer  subbridge\_removePeer  subbridge\_getReceiptFromParentChain  subbridge\_getAnchoringTxHashByBlockNumber  subbridge\_registerOperator  subbridge\_getRegisteredOperators  subbridge\_getValueTransferOperatorThreshold  subbridge\_setValueTransferOperatorThreshold  subbridge\_deployBridge  subbridge\_subscribeBridge  subbridge\_unsubscribeBridge  subbridge\_kASAnchor  subbridge\_anchoring  subbridge\_registerBridge  subbridge\_deregisterBridge  subbridge\_registerToken  subbridge\_deregisterToken  subbridge\_convertRequestTxHashToHandleTxHash  subbridge\_getBridgeInformation  subbridge\_getParentTransactionReceipt  subbridge\_setKLAYFee  subbridge\_setERC20Fee  subbridge\_setFeeReceiver  subbridge\_getKLAYFee  subbridge\_getERC20Fee  subbridge\_getFeeReceiver  subbridge\_lockParentOperator  subbridge\_lockChildOperator  subbridge\_unlockParentOperator  subbridge\_unlockChildOperator  subbridge\_setParentOperatorFeePayer  subbridge\_setChildOperatorFeePayer  subbridge\_getBridgePairByAlias  subbridge\_changeBridgeAlias  subbridge\_setParentBridgeOperatorGasLimit  subbridge\_setChildBridgeOperatorGasLimit  subbridge\_getParentBridgeContractBalance  subbridge\_getChildBridgeContractBalance  subbridge\_requestParentSync  subbridge\_nodeInfo  subbridge\_peers  subbridge\_getParentOperatorAddr  subbridge\_getChildOperatorAddr  subbridge\_getOperators  subbridge\_getAnchoringPeriod  subbridge\_getSentChainTxsLimit  subbridge\_getParentOperatorNonce  subbridge\_getChildOperatorNonce  subbridge\_getParentOperatorBalance  subbridge\_getChildOperatorBalance  subbridge\_listBridge  subbridge\_txPendingCount  subbridge\_txPending  subbridge\_getLatestAnchoredBlockNumber  subbridge\_getParentOperatorFeePayer  subbridge\_getChildOperatorFeePayer  subbridge\_getParentBridgeOperatorGasLimit  subbridge\_getChildBridgeOperatorGasLimit  subbridge\_getParentGasPrice  subbridge\_getParentKIP71Config |
| --- | --- |

* rpc namespace

| RPC Name | rpc\_modules |
| --- | --- |
| Input/Output | // Request curl -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"rpc\_modules","id":1}' https://api.baobab.klaytn.net:8651 // Result {  "Jsonrpc":"2.0",  "Id":1,  "Result":{  "Debug":"1.0","eth":"1.0","klay":"1.0","net":"1.0","rpc":"1.0","web3":"1.0"}  } |

* transaction error code
* reference link (<https://docs.klaytn.foundation/content/dapp/json-rpc/transaction-error-codes>)

1. Architecture

* RPC Specification is written as a yaml file according to OAS 3.0.
* It can be converted into an html documentation that can show the specification pretty by redocly.
* It is used as input to the code generator to make the source code. For example, It can replace the placeholder in the mustache file, predefined source code template. You can download the code generator jar file for each language you want to make, and the customization through the class inheritance is possible.



1. Implementation

* OpenSDK covers only **eth/klay/personal/net/rpc** namespace functions
* Example Writing Klaytn RPC Specification in python ([klay\_getRewards](https://docs.klaytn.foundation/content/dapp/json-rpc/api-references/klay/block#klay_getrewards) case)
  + An API Specification is written by yaml format in OAS 3.0 syntax
  + Following is the main keywords
    - paths: available paths and operations in url
    - post: http method
    - requestBody: request body in http
    - responses: response in http
    - x-codeSamples: linked to example code in documentation
    - allOf: merge all the sub-objects
    - tags: you can use it for grouping in code and docs
  + Objects in other documents can be used as references instead. *$ref* 
    - In this example, *JsonRpcRequest* and JsonRpcResponse are used as $ref object that is used in other places frequently
    - *RewardsReq* and *RewardResp* objects in the *RewardsSpec* document are also used as a $ref
    - Example code, KlayApiGetRewardsTest.py, is also used as a $ref

| openapi: "3.0.2" info:  title: Klay Open API  version: "1.10.0"  license:  name: Apache 2.0  url: https://www.apache.org/licenses/LICENSE-2.0.html servers:  - url: http://localhost:8551 paths:  /klay/getRewards:  post:  tags:  - klay  operationId: getRewards  summary: "[getRewards] Find distributed block rewards"  description: |  Returns distributed block rewards information at the latest block number   \*\*JSONRPC:\*\* `klay\_getRewards`    requestBody:  content:  application/json:  schema:   allOf:  - $ref: "../../components/requests/JsonRpcRequest.yaml#/components/schemas/JsonRpcRequest"  - $ref: "../../components/schemas/RewardsSpec.yaml#/components/schemas/RewardsReq"  responses:  default:  description: |  block rewards information <br/>  Link <br/>  <SchemaDefinition schemaRef="#/components/schemas/RewardSpec" />  content:  application/json:  schema:  allOf:  - $ref: "../../components/responses/JsonRpcResponse.yaml#/components/schemas/JsonRpcResponse"  - $ref: "../../components/schemas/RewardsSpec.yaml#/components/schemas/RewardsResp"  x-codeSamples:  - lang: "curl"  label: "Curl"  source:  $ref: "../../code-samples/klay/getReward.sh"  - lang: "python"  label: "Python"  source:   $ref: "../../../sdk/client/python/openapi-test/test/klay/KlayApiGetRewardsTest.py" |
| --- |

../../components/requests/JsonRpcRequest.yaml

| components:  schemas:  JsonRpcRequest:  type: object  required:  - method  - id  - jsonrpc  properties:  method:  type: string  description: Method name  id:  type: integer  default: 1  format: int64  description: Request ID, Random or Auto incrementation  jsonrpc:  type: string  default: "2.0"  description: JSON-RPC Version (2.0)  params:  type: array  items: {} |
| --- |

../../components/responses/JsonRpcResponse.yaml

| .. skip .. |
| --- |

../../components/schemas/RewardsSpec.yaml

| components:  schemas:  RewardsReq:  type: object  properties:  method:  type: string  default: 'klay\_getRewards'  params:  type: array  items:  default: "latest"  oneOf:  - $ref: "#/components/schemas/BlockNumber"  - $ref: "#/components/schemas/BlockTag"  BlockNumber:  title: blockNumber  type: integer  format: int64  description: the block number  BlockTag:  title: blockTag  type: string  description: the block tag [latest, pending, eariest]    RewardsResp:  type: object  properties:  result:  $ref: "#/components/schemas/Result"  Result:  type: object  required:  - burntFee  - kgf  - kir  - minted  - proposer  - rewards  - stakers  - totalFee  properties:  burntFee:  type: integer  format: int64  description: the amount burnt  kgf:  type: integer  format: int64  description: the amount allocated to KGF  kir:  type: integer  format: int64  description: the amount allocated to KIR  minted:  type: number  format: double  description: the amount newly minted  example: 9600000000000000000  proposer:  type: number  format: double  description: the amount allocated to the block proposer  example: 9600000000000000000  rewards:  type: object  description: mapping from reward recipient to amounts  additionalProperties:  type: number  format: double  example:  0xb408fc9b2512284823264aceeb50d2970533b4ff: 9600000000000000000  stakers:  type: integer  format: int64  description: total amount allocated to stakers  totalFee:  type: integer  format: int64  description: total tx fee spent |
| --- |

KlayApiGetRewardsTest.py

| # coding: utf-8 import unittest from caver\_python.configuration import Configuration from caver\_python.api\_client import ApiClient from caver\_python.apis.tags.klay\_api import KlayApi  class TestKlayGetRewards(unittest.TestCase):  \_configuration = Configuration(host="http://localhost:8551")  klay = KlayApi(ApiClient(configuration=\_configuration))    def test\_post(self):  get\_rewards\_request = {  "method": "klay\_getRewards",  "id": 1,  "jsonrpc": "2.0",  "params": [  0  ]  }  result = self.klay.get\_rewards(body=get\_rewards\_request)  if \_\_name\_\_ == '\_\_main\_\_':  unittest.main() |
| --- |

* SDK Code generation
  + <https://openapi-generator.tech/docs/installation>
  + Download default python code generator jar package

| wget https://repo1.maven.org/maven2/org/openapitools/openapi-generator-cli/6.3.0/openapi-generator-cli-6.3.0.jar -O openapi-generator-cli.jar |
| --- |

* Following script generates the SDK code using default mustache files in the github repository. If you want to use a modified or added mustache file, just add a template folder that stores it and uncomment the templateDir in python-config.yaml.

| #!/usr/bin/env bash  CURRENT\_FILE\_DIR=$(cd "$( dirname "${BASH\_SOURCE[0]}" )" && pwd) PROJECT\_DIR=$(cd "$CURRENT\_FILE\_DIR" && cd ../../.. && pwd ) cd "${CURRENT\_FILE\_DIR}" rm -rf "${CURRENT\_FILE\_DIR}/openapi" mkdir "${CURRENT\_FILE\_DIR}/openapi" cp .openapi-generator-ignore "${CURRENT\_FILE\_DIR}/openapi"  "${PROJECT\_DIR}"/bin/caver-openapi-generator-cli generate -c "${CURRENT\_FILE\_DIR}/python-config.yaml"  cd "${CURRENT\_FILE\_DIR}/openapi" |
| --- |

python-config.yaml (code generator config file)

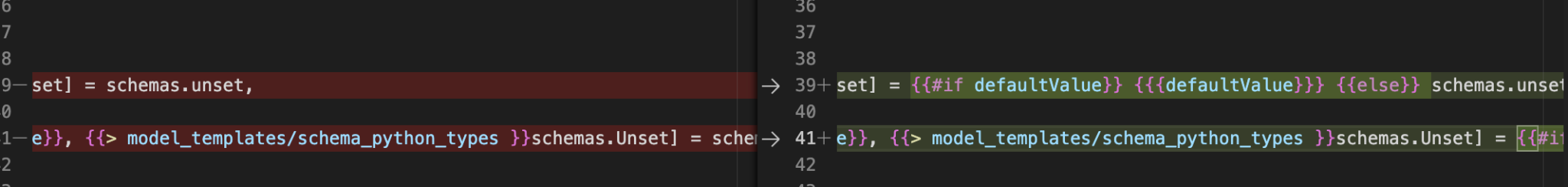
| generatorName: caver-python outputDir: ./openapi inputSpec: ../../../site/klaytn-openapi.yaml projectName: caver-python packageName: caver\_python # templateDir: ./template sortParamsByRequiredFlag: false globalProperties:  apiDocs: false  modelDocs: false  modelTests: false  apiTests: true inlineSchemaNameMappings:  accountCreated\_default\_response: AccountCreatedResponse  getRewards\_default\_response: GetRewardsResponse |
| --- |

* Code template tuning with mustache for getting only variable parameter

1. Modifying [schema\_composed\_or\_anytype.handlebars](https://github.com/OpenAPITools/openapi-generator/blob/master/modules/openapi-generator/src/main/resources/python/model_templates/schema_composed_or_anytype.handlebars) and [new.handlebars](https://github.com/OpenAPITools/openapi-generator/blob/master/modules/openapi-generator/src/main/resources/python/model_templates/new.handlebars) to set default value in specification because python generator doesn’t care about it. just need to get the file only you want to modify. code generator will use the online repo for the other files. In this case It is only handled to modify “allOf” compound type.

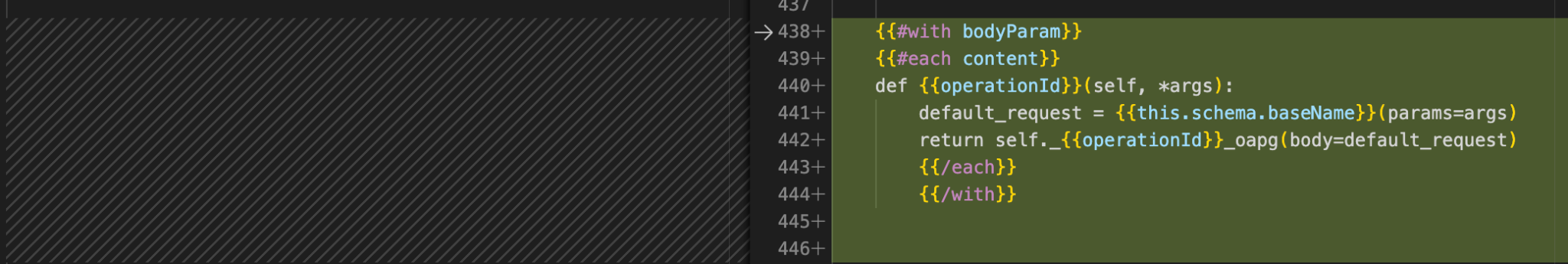


model\_templates/schema\_composed\_or\_anytype.handlebars



model\_templates/new.handlebars

1. Modifying [endpoint.handlebars](https://github.com/OpenAPITools/openapi-generator/blob/master/modules/openapi-generator/src/main/resources/python/endpoint.handlebars) to generate an overloaded function that can accept the only variable parameters. In klay\_getRewards case, it is blockNumberOrTag parameter.



endpoint.handlebars

1. Modifying the test code

| # coding: utf-8 import unittest from caver\_python.configuration import Configuration from caver\_python.api\_client import ApiClient from caver\_python.apis.tags.klay\_api import KlayApi  class TestKlayGetRewards(unittest.TestCase):  host="http://localhost:8551"  klay = KlayApi(ApiClient(configuration=Configuration(host=host)))    def test\_post(self):  result = self.klay.get\_rewards("latest")  if \_\_name\_\_ == '\_\_main\_\_':  unittest.main() |
| --- |

Ground Rules for writing RPC Spec. and Test code.

* You should specify either [**parameters**](https://github.com/kjeom/klaytn-open-sdk/blob/main/api/paths/klay/rewards.yaml#L23) and [**params**](https://github.com/kjeom/klaytn-open-sdk/blob/main/api/components/schemas/RewardsSpec.yaml#L13).
  + There should be **parameters** if you want to get **a specific type**.
  + According to my analysis, you cannot get a specific type from params specification.
  + The parameters **will be deleted** after generating source code for making **right documentation**
* The **directory** is a **namespace** and it contains OAS 3.0 yaml files.
* **One file** includes only **one API** spec.
  + If an object is repeated multiple times, you can split the object into another file in the **common directory** and reference it by *$ref*
* Just use **OpenSDK** as the name, not caver like example.
* It’s good to write the **test codes** in **another folder**, not auto-generated test folder.

Additional request

* You need to **refer** to the **web3j** request/response pattern for java OpenSDK.
  + Unlike javascript and python (ethers.js, web3.js and web3.py), java requires the wrapper package to integrate with web3j.
  + I think it would be better to implement manually than to use openapi generator.
  + Web3j RPC request pattern
  + You can reference [repo](https://github.com/kjeom/klaytn-open-sdk/tree/main/sdk/client/java/template) and run [java-generate.sh](https://github.com/kjeom/klaytn-open-sdk/blob/main/sdk/client/java/java-generate.sh)

| EthGetTransactionReceipt transactionReceipt =  web3j.ethGetTransactionReceipt(transactionHash).send(); |
| --- |

* Web3j request
  + web3.[ethGetTransactionReceipt](https://github.com/web3j/web3j/blob/master/core/src/main/java/org/web3j/protocol/core/JsonRpc2_0Web3j.java#L442) return Request instance
  + The key is that the instance has send() method

| public Request<?, EthGetTransactionReceipt> ethGetTransactionReceipt(String transactionHash) {  return new Request<>(  "eth\_getTransactionReceipt",  Arrays.asList(transactionHash),  web3jService,  EthGetTransactionReceipt.class);  } |
| --- |

* Web3j response
  + Every response instance like [EthGetTransactionRecept](https://github.com/web3j/web3j/blob/master/core/src/main/java/org/web3j/protocol/core/methods/response/EthGetTransactionReceipt.java) inherit **Response**

| public class EthGetTransactionReceipt extends Response<TransactionReceipt> {  public Optional<TransactionReceipt> getTransactionReceipt() {  return Optional.ofNullable(getResult());  }  public static class ResponseDeserialiser extends JsonDeserializer<TransactionReceipt> {  private ObjectReader objectReader = ObjectMapperFactory.getObjectReader();  @Override  public TransactionReceipt deserialize(  JsonParser jsonParser, DeserializationContext deserializationContext)  throws IOException {  if (jsonParser.getCurrentToken() != JsonToken.VALUE\_NULL) {  return objectReader.readValue(jsonParser, TransactionReceipt.class);  } else {  return null; // null is wrapped by Optional in above getter  }  }  }  } |
| --- |