

HTTP interface to CasperExplorer server (v8)

Black text should/will appear as is; orange text should/will be substituted with values of the specified type.

get /data

Read current state of the app from server.

query parameters	response body
None	<pre>{ accounts: [{ // zero or more name: string, publicKey: string }], payment: [string // zero or more], session: [string // zero or more], savedDeploys: [{ // zero or more name: string, payment: string, paymentArgs: string, session: string, sessionArgs: string }], savedQueries: [{ // zero or more name: string, block: string, variant: string, key: string, path: string }] }</pre>

Errors

None

post /account

Create key pair on server, create account identified by “name” (and the public key of the newly generated key pair), and initialize its balance to “balance”.

request body	response body
<pre>{ name: string, balance: int }</pre>	<pre>{ status: boolean, message: string, publicKey: string }</pre>

Errors

- Malformed request body
- Error emitted by node in response to account creation deploy

delete /account

Delete account identified by “name”. This only removes the account data from the persistent store of the server – it does not delete the account on the node (which cannot be done yet).

request body	response body
<pre>{ name: string }</pre>	<pre>{ status: boolean, message: string }</pre>

Errors

- Malformed request body
- Account with specified name does not exist

put /contract

Store new contract in the contract library.

request body	response body
<pre>{ name: string, type: string, // "payment","session" wasm: string // base64 encoded }</pre>	<pre>{ status: boolean, message: string }</pre>

Errors

- Malformed request body
- Invalid "type" value

post /contract/save

Save a deploy with specified parameters and name. Any existing saved deploy will be overwritten.

request body	response body
<pre>{ name: string, payment: string, paymentArgs: string, session: string, sessionArgs: string }</pre>	<pre>{ status: boolean, message: string }</pre>

Errors

- Malformed request body

post /contract/deploy

Deploy specified contracts to a node.

request body	response body
<pre>{ account: string, payment: string, paymentArgs: string, session: string, sessionArgs: string, propose: boolean }</pre>	<pre>{ status: boolean, message: string }</pre>

Errors

- ▶ Malformed request body
- ▶ Error emitted by node in response to deploy

delete /contract

Delete contract identified by “name” from the contract library.

request body	response body
<pre>{ name: string, type: string //one of: "payment", "session", "saved" }</pre>	<pre>{ status: boolean, message: string }</pre>

Errors

- ▶ Malformed request body
- ▶ Invalid “type” value
- ▶ Contract with specified type and name does not exist

post /query/save

Save a query with specified parameters and name. Any existing saved query will be overwritten.

request body	response body
<pre>{ name: string, block: string, variant: string, key: string, path: string }</pre>	<pre>{ status: boolean, message: string }</pre>

Errors

- Malformed request body

get /query

Query specific variables on the DAG.

query parameters	response body
<pre>blockHash=string & keyVariant=string & keyBytes=string & path=string</pre>	<pre>{ status: string, message: string, result: string }</pre>

Errors

- Invalid query parameters
- Error emitted by node in response to query

get /dag

Get the topology of the DAG.

query parameters	response body
	<pre>{ status: string, message: string, result: string }</pre>

Errors

- Error emitted by node in response to query