HTTP interface to CasperExplorer server

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.

request body	response body

	`{`
	`accounts: [`
	`  string  // zero or more`
	`],`
	`keys: [`
	`  string  // zero or more`
	`],`
	`payment: [`
	`  string  // zero or more`
	`],`
	`session: [`
	`  string  // zero or more`
	`]`
	`}`

post /key

Create new key pair on the server identified by “name”.
Any existing key pair saved under “name” will be overwritten.

request body	response body

`{`
`name: string`
`}`
	`{`
	`status: boolean,`
	`message: string,  // if ! status`
	`publicKey: string`
	`}`

put /key

Store new key pair uploaded from client and identified by “name”.
Any existing key pair saved under “name” will be overwritten.

request body	response body

{
name: string,
keyPair: string // contents of
} // uploaded file
	{
	status: boolean,
	message: string // if ! status
	}

delete /key

Delete key pair identified by “name”.
TBD: What about any corresponding account?

request body	response body

{
name: string
}
	{
	status: boolean,
	message: string // if ! status
	}

post /account

Create account identified by public key of key pair identified by “name” and initialize its balance to “balance”.

request body	response body

{
name: string,
balance: int
}
	{
	status: boolean,
	message: string // if ! status
	}

put /contract

Store new contract in the contract library.

request body	response body

{
name: string,
type: string, // "payment","session"
wasm: string // base64 encoded
}
	{
	status: boolean,
	message: string // if ! status
	}

delete /contract

Delete contract identified by "name" from the contract library.

request body	response body

{
name: string
}
	{
	status: boolean,
	message: string // if ! status
	}

post /contract

Deploy specified contracts to a node.

request body	response body

{
payment: string,
paymentArgs: string,
session: string,
sessionArgs: string
}
	{
	status: boolean,
	message: string // if ! status
	}

get /query

Query state of the DAG.

request body	response body

{
blockHash: string,
keyVariant: string, // enum
keyBytes: string,
path: string
}
	{
	result: string
	}