

Links

Swagger: <https://staking-testnet.cratd2csmartchain.io/api/v1/swagger/>

Django admin: <https://staking-testnet.cratd2csmartchain.io/django-admin/> (login:, password:)

Gitlab:

https://gitlab.rocknblock.io/rock-n-block/projects/crat/cratd2c_stakingmanager_backend

Accounts

Endpoint: `api/v1/accounts/wallet/get_message/`

Methods: GET

Description: Generates message, that the user should sign via wallet extension for authorization

Responses:

- 200 OK
Response body:

```
{  
  "message": "Sign this message to connect your wallet"  
}
```

Endpoint: `api/v1/accounts/wallet/connect/`

Methods: POST

Description: View that requires user address, message and message signed with a private key

Request Body:

```
{  
  "address": "0x1234567890abcdef...",  
  "message": "Sign this message to connect your wallet.",  
  "signed_msg": "0xabcdef..."  
}
```

Name	Type	Required	Description
address	string	Yes	Address of the user
message	string	Yes	The message that needs to be signed

signed_msg	string	Yes	The signature of the message signed with the private key
------------	--------	-----	--

Responses:

- 200 OK
Response Body:

```
{
  "token": "abcd1234efgh5678ijkl"
}
```
- 400 Bad Request
Response Body:

```
{
  "address": ["Address validation error", "This field is required."],
  "message": ["This field is required."],
  "signed_msg": ["Please provide a correct signature and address.", "This field is required."]
}
```

Notes:

- Ensure the user's address is in a valid format.
- The message should be consistent and must be signed with the private key corresponding to the provided address.

Statistic

Endpoint: api/v1/staticstic/interval

Methods: GET

Description: Return interval between current checkpoint block and previous checkpoint block.

Responses:

- 200 OK
Response Body:

```
{
  "3020"
}
```

Caching: The response is cached for 1200 seconds

Endpoint: api/v1/staticstic/delegators

Methods: `GET`

Description: Return total Delegators for all active validators.

Responses:

- 200 OK
Response Body:

```
{  
  "200"  
}
```

Caching: The response is cached for 60 seconds

Validators

Endpoint: `api/v1/validators/`

Description: Get all validators or create a validator

Methods:

- GET – Returns the user's paginated list of validators (HEALTHY, STOPPED).

Query Parameters:

Parameter	Type	Required	Description
sort	string	No	Field to sort the results
page	number	No	The page number to retrieve in the paginated results
page_size	number	No	Number of results per page
address	string	No	Exact match for the validator's address
name	string	No	Partial match for the validator's name
ordering	string	No	Ordering field

Responses:

- 200 OK
Response Body:

```
{  
  "count": 1,  
  "next": "https://domain.io/api/v1/validators/?page=2",  
  "previous": null,  
  "results": [  
    {  
      "id": 1,  
      "checkpoints": 2186.0,  
      "performance_index": 100.0,  
    }  
  ]  
}
```

```

    "status": "HEALTHY",
    "name": "TOP 4th Validator",
    "logo": "https://domain.io/media/validator_logo/1.png",
    "stake_amount": "6913181728971078510422019",
    "commission": "1000",
    "start_block": "1666750",
    "created_at": "2024-08-19T12:47:10.163193Z",
    "updated_at": "2024-10-04T11:19:00.088218Z",
    "address": "0xfdE2F9854BAc6602A9C95b98b23835371e8B84FB",
    "description": "",
    "website": "",
    "twitter": "",
    "telegram": "",
    "penalty": "0"
  },
]
}

```

- POST – Creates a new validator.

Request Body:

```

{
  "name": "Validator Name",
  "logo": "...",
  "address": "0x1234567890abcdef...",
  "commission": 10,
  "description": "Description of the validator.",
  "twitter": "https://twitter.com/validator",
  "telegram": "https://t.me/validator",
  "website": "https://validator.com"
}

```

Name	Type	Required	Description
name	string	Yes	The name of the validator
logo	string	Yes	Base64-encoded image representing the validator
address	string	Yes	Address of the validator
commission	number	No	The commission of the validator
description	string	No	Description of the validator
twitter	string	No	URL to the validator's Twitter

telegram	string	No	URL to the validator's Telegram
website	string	No	URL to the validator's website

Responses:

- 201 Created

Response Body:

```
{
  "id": 1,
  "checkpoints": 0,
  "performance_index": 100.0,
  "status": "HEALTHY",
  "name": "Validator Name",
  "logo": "https://domain.io/media/validator_logo/1.png",
  "stake_amount": "6913181728971078510422019",
  "commission": "0.1",
  "start_block": "1666750",
  "created_at": "2024-08-19T12:47:10.163193Z",
  "updated_at": "2024-10-04T11:19:00.088218Z",
  "address": "0x1234567890abcdef...",
  "description": "Description of the validator.",
  "website": "https://validator.com",
  "twitter": "https://twitter.com/validator",
  "telegram": "https://t.me/validator",
  "penalty": "0"
}
```

- 400 Bad Request

Response Body:

```
{
  "name": ["This field is required."],
  "logo": ["This field is required."],
  "address": ["Invalid address", "This field is required.", "Balance too low"],
}
```

Notes:

- The logo field should contain a valid Base64-encoded image string.
- If a validator with the same address already exists and its status is ARCHIVED, it will be deleted before creating the new validator.
- Additional validation checks for address balance.

Endpoint: `api/v1/validators/update/<str:address>`

Description: Validator update

Methods: PUT, PATCH

Request Body:

```
{
  "name": "Updated Validator Name",
  "logo": "...",
  "description": "Updated description of the validator.",
  "twitter": "https://twitter.com/updated_validator",
  "telegram": "https://t.me/updated_validator",
  "website": "https://updated_validator.com"
}
```

Name	Type	Required	Description
name	string	Yes	The name of the validator
logo	string	Yes	Base64-encoded image representing the validator
description	string	No	Description of the validator
twitter	string	No	URL to the validator's Twitter
telegram	string	No	URL to the validator's Telegram
website	string	No	URL to the validator's website

Responses:

- 200 OK

```
{
  "name": "Updated Validator Name",
  "logo": "...",
  "description": "Updated description of the validator.",
  "twitter": "https://twitter.com/updated_validator",
  "telegram": "https://t.me/updated_validator",
  "website": "https://updated_validator.com"
}
```
- 404 Not Found
- 400 400 Bad Request

Notes:

- The address parameter in the URL must correspond to a valid address for an existing validator.

Endpoint: `api/v1/validators/calldata`

Methods: GET

Description: Get transaction payload.

Query Parameters:

Name	Type	Required	Description
address	string	Yes	Address of the validator
commission	number	Yes	The commission of the validator
amount	number	Yes	The deposit amount

Responses:

- 200 OK

Response body:

```
{
  "from": "0x123456....",
  "gasPrice": 10000000000,
  "chainId": 65349,
  "nonce": 158345,
  "gas": 1000000,
  "value": 1000,
  "to": "0x4015c8A91849f09d1105De279B46dc358a9fF3Da",
  "data": "0x9cb56b66000..."
}
```

- 400 Bad Request

Endpoint: api/v1/validators/transaction

Methods: GET

Description: Update validator's status based on data from the blockchain.

Query Parameters:

Name	Type	Required	Description
tx_hash	string	Yes	Transaction hash
address	string	Yes	Address of the validator

Responses:

- 200 OK
- 400 Bad Request

Endpoint: api/v1/validators/delegator

Methods: POST

Description: Get validators depends on body.

Query Parameters:

```
{
  "addresses": "0x1234567890...,0x1234567890abcd..."
}
```

Query Parameters:

Name	Type	Required	Description
addresses	string	Yes	Comma separated addresses
page	number	No	The page number to retrieve in the paginated results
page_size	number	No	Number of results per page

Responses:

- 200 OK

Response Body:

```
{
  "count": 1,
  "next": "https://domain.io/api/v1/validators/?page=2",
  "previous": null,
  "results": [
    {
      "id": 1,
      "checkpoints": 2186.0,
      "performance_index": 100.0,
      "status": "HEALTHY",
      "name": "TOP 4th Validator",
      "logo": "https://domain.io/media/validator_logo/1.png",
      "stake_amount": "6913181728971078510422019",
      "commission": "1000",
      "start_block": "1666750",
      "created_at": "2024-08-19T12:47:10.163193Z",
      "updated_at": "2024-10-04T11:19:00.088218Z",
      "address": "0xfdE2F9854BAc6602A9C95b98b23835371e8B84FB",
      "description": "",
      "website": "",
      "twitter": "",
      "telegram": "",
      "penalty": "0"
    }
  ]
}
```

- 400 Bad Request

Endpoint: api/v1/validators/penalty

Methods: POST

Description: Increase validator's penalty.

Request Body:

```
{  
  "address": "0x1234567890..."  
}
```

Name	Type	Required	Description
address	string	Yes	Address of the validator

Responses:

- 200 OK
- 400 Bad Request