# Versions

## 0.2.0

### New:

- `user.addWhitelist`
- `user.delWhitelist`
- `user.addBlacklist`
- `user.delBlacklist`

### Modified:

- `task.list`

### Deprecates:

- `user.listTemplate`

## 0.1.0

- Initial specification

# Table of Contents

- System
  - sys.ping
  - sys.watchdog
  - sys.login
  - sys.reload
  - sys.node.list
  - sys.session.list
  - sys.session.kick
  - sys.session.reload
- Pipes
  - pipe.create
  - pipe.close
  - pipe.write
  - pipe.read
- Sync
  - sync.lock

    - sync.unlock
- Tasks
    - task.push
    - task.pull
    - task.result
    - task.error
    - task.reject
    - task.cancel
    - task.list
- Topics
    - topic.sub
    - topic.unsub
    - topic.pub
- Users
    - user.create
    - user.delete
    - user.list
    - user.setTags
    - user.delTags
    - user.setPass
    - user.addTemplate
    - user.delTemplate
    - user.addWhitelist
    - user.delWhitelist
    - user.addBlacklist
    - user.delBlacklist
    - user.setMaxSessions

# System

## sys.ping

Test the connection or generate some traffic to keep the connection alive.

### Parameter:

- `null`

### Result:

```
"result": "pong"
```

### sys.watchdog

Configure the time the connection will be considered alive without traffic.

#### Parameters:

- `Number` - *Optional* - Sets the number of seconds the watchdog will hold. If not set, the result will show the current value.

#### Result:

```
"result": { "ok": true, "watchdog": 10}
```

### sys.login

Switches the user working with the current connection.

#### Parameters:

- `"method": <string>` - *Optional* - Specifies the login method. If omitted, defaults to “basic”.

If auth method is basic:

- `"user": <string>` - User to login as
- `"pass": <string>` - User’s password

Else, the specified method should document which fields its expecting

#### Result:

```
"result": { "ok": true, "connid": <string>, "user": <string> }
```

### sys.reload

## Parameters:

- `null`

## Result:

```
"result": { "ok": true }
```

## sys.node.list

List the nexus nodes connected to the cluster. Includes some info about connected clients and CPU load for each node.

**Parameters:**

- `"limit": <Number>` - *Optional* - Limit the number of results. Defaults to 100
- `"skip": <Number>` - *Optional* - Skips a number of results. Defaults to 0

**Result:**

```
 "result": [ {"id": <string>, "clients": <Number>, "load": {"load1": <Number>, "load5": <Nur
```

## sys.session.list

List the active sessions on the cluster.

**Parameters:**

- `"prefix": <String>` - Session prefix
- `"limit": <Number>` - *Optional* - Limit the number of results. Defaults to 100
- `"skip": <Number>` - *Optional* - Skips a number of results. Defaults to 0

**Result:**

```
"result": [{"sessions":[{"creationTime":"2016-08-30T12:39:16.39Z","id":"687c3b7baf4b9471","r
```

## sys.session.kick

Terminates any connection which session id matches the prefix

**Parameters:**

- `"connId": <String>` - Connection ID prefix

**Result:**

```
"result": { "kicked": 7 }
```

### sys.session.reload

Reloads user data for any connection which connection id matches the prefix

**Parameters:**

- `"connId": <String>` - Connection ID prefix

**Result:**

```
"result": { "reloaded": 2 }
```

## Pipes

### pipe.create

Creates a new pipe.

**Parameters:**

- `"len": <Number>` - *Optional* - Maximum capacity of the pipe. Defaults to 1000

**Result:**

```
"result": { "pipeid": <string> }
```

### pipe.close

Closes a pipe

**Parameters:**

- `"pipeid": <String>` - PipeID of the pipe to close

**Result:**

```
"result": { "ok": true }
```

## pipe.write

Writes any JSON object into a pipe.

**Parameters:**

- `"pipeid": <String>` - PipeID of the pipe to write to
- `"msg": <Object>` - Data to write to the pipe

**Result:**

```
"result": { "ok": true }
```

## pipe.read

Reads a JSON object from a pipe. Blocks until an element is available on the pipe or exceeds the timeout

**Parameters:**

- `"pipeid": <String>` - PipeID of the pipe to write to
- `"max": <Number>` - Maximum number of elements to read from the pipe
- `"timeout": <Number>` - Maximum number of second to wait for a read to happen. Defaults to blocking forever

**Result:**

```
{ "waiting": <Number>, "drops": <Number>, "msgs": [{ "msg": <Object>, "count": <Number> }, .
```

- `waiting`: Number of messages still in the pipe
- `drops`: Number of messages which could not be read on time, did not fit on the pipe and were lost.
- `msgs`: Array of objects containing the data written to the pipe and a secuential identifier

# Sync

## sync.lock

Grabs a lock, cluster-wide.

### Parameters:

- `"lock": <String>` - Name of the lock to grab

### Result:

```
"result": { "ok": <Bool> }
```

## sync.unlock

Frees a lock, cluster-wide.

### Parameters:

- `"lock": <String>` - Name of the lock to grab

### Result:

```
"result": { "ok": <Bool> }
```

# Tasks

## task.push

Calls a method which will be resolved by the system, and will return a result or an error (examples on the result section)

### Parameters:

- `"method": <String>` - Which method is this task invoking
- `"params": <Object>` - Method parameters
- `"detached": <Bool>` - The task will eventually be processed but we do not care about the result

- `"prio": <Number>` - Sets the priority of this task among other pushes on the same method
- `"ttl": <Number>` - How many times this task can be requeued (by a failed worker/node or a task reject)
- `"timeout": <Number>` - How much time should a task be on any state other than “done” before the task is considered failed.

**Result:**

If “detached” is true, it will immediately receive:

`"result": { "ok": true }`

Otherwise, it will get an answer defined by the worker who pulls the task:

`"result": { "answer": 42 }`

or

`"error": {"code":123,"message":"asdf","data":""}`

## task.pull

Pulls a task from a path to work on

**Parameters:**

- `"prefix": <String>` - Prefix to pull tasks from
- `"timeout": <Number>` - How much time should we wait for a task to get pulled

**Result:**

```
 "result": {"detach":false,"method":"test","params":{},"path":"asdf.","prio":0,"tags":{"@adm
```

## task.result

Mark a task as finished successfully, and set the task result parameter

**Parameters:**

- `"taskid": <String>` - Task being resolved
- `"result": <Object>` - Data delivered to the pusher as “result”

**Result:**

```
"result": { "ok": true }
```

### task.error

Mark a task as finished with an error, and set the error fields

**Parameters:**

- `"taskid": <String>` - Task being resolved with an error
- `"code": <Object>` - *Optional* - Error code
- `"message": <Object>` - *Optional* - Error message
- `"data": <Object>` - *Optional* - Error data

**Result:**

```
"result": { "ok": true }
```

### task.reject

Reject a pulled task. It will be marked as waiting, and available to be pulled again. Decrements the task's TTL

**Parameters:**

- `"taskid": <String>` - Task being rejected

**Result:**

```
"result": { "ok": true }
```

### task.cancel

Cancel a task, which will mark it as cancelled and wake up whoever was waiting for its completion

**Parameters:**

- `"taskid": <String>` - Task being cancelled

**Result:**

```
"result": { "ok": true }
```

### task.list

List tasks happening inside a prefix and its properties

**Parameters:**

- `"prefix": <String>` - Path prefix
- `"limit": <Number>` - *Optional* - Limit the number of results. Defaults to 100
- `"skip": <Number>` - *Optional* - Skips a number of results. Defaults to 0

**Result:**

```
"result":  [{"id":"687c3b7bfbcdae7cb774d215cf923252f3fb","state":"waiting","path":"test.","p
```

## Topics

### topic.sub

Subscribe a pipe to a topic. Everything published on the topic will be written on the pipe

**Parameters:**

- `"pipeid": <String>` - PipeID to subscribe
- `"topic": <String>` - Topic to subscribe the pipe to

**Result:**

```
"result": { "ok": true }
```

### topic.unsub

Unsubscribe a pipe from a topic.

**Parameters:**

- `"pipeid": <String>` - PipeID to subscribe
- `"topic": <String>` - Topic to unsubscribe the pipe from

**Result:**

```
"result": { "ok": true }
```

### topic.pub

Publish data to a topic.

**Parameters:**

- `"topic": <String>` - Topic to send the data to
- `"msg": <Object>` - Data to send

**Result:**

```
"result": { "ok": true }
```

## Users

### user.create

Create a new user which will be able to authenticate by basic auth

**Parameters:**

- `"user": <String>` - Username of the new user
- `"pass": <String>` - Password of the new user

**Result:**

```
"result": { "ok": true }
```

## user.delete

Delete an existent user

**Parameters:**

- `"user": <String>` - Username of the user to delete

**Result:**

```
"result": { "ok": true }
```

## user.list

Lists users that matches the prefix

**Parameters:**

- `"prefix": <String>` - Prefix where the tags will take effect
- `"limit": <Number>` - *Optional* - Limit the number of results. Defaults to 100
- `"skip": <Number>` - *Optional* - Skips a number of results. Defaults to 0

**Result:**

```
"result": [{"blacklist":["172.17.*"],"maxsessions":42,"tags":{test":{"@admin":true}},"templa
```

## user.setTags

Set a tag on an user on a prefix

**Parameters:**

- `"user": <String>` - Username of the user to set tags on
- `"prefix": <String>` - Prefix where the tags will take effect
- `"tags": <Object>` - Tags to be set

**Result:**

```
"result": { "ok": true }
```

### user.delTags

Remove a tag from an user on a prefix

**Parameters:**

- `"user": <String>` - Username of the user to remove tags from
- `"prefix": <String>` - Prefix where the tags will take effect
- `"tags": <Object>` - Tags to be deleted

**Result:**

```
"result": { "ok": true }
```

### user.setPass

Set the user password for basic auth

**Parameters:**

- `"user": <String>` - Username of the user
- `"pass": <String>` - New password

**Result:**

```
"result": { "ok": true }
```

### user.addTemplate

Add a template to an user

**Parameters:**

- `"user": <String>` - Username
- `"template": <String>` - Template to add

**Result:**

```
"result": { "ok": true }
```

## user.delTemplate

Remove a template from an user

**Parameters:**

- `"user": <String>` - Username
- `"template": <String>` - Template to remove

**Result:**

```
"result": { "ok": true }
```

## user.addWhitelist

Add an address to an user whitelist

**Parameters:**

- `"user": <String>` - Username
- `"ip": <String>` - IP address. Accepts regular expressions (192.168.*)

**Result:**

```
"result": { "ok": true }
```

## user.delWhitelist

Remove an address to an user whitelist

**Parameters:**

- `"user": <String>` - Username of the user
- `"ip": <String>` - IP address. Accepts regular expressions (192.168.*)

**Result:**

```
"result": { "ok": true }
```

## user.addBlacklist

Add an address to an user blacklist

**Parameters:**

- `"user": <String>` - Username of the user
- `"ip": <String>` - IP address. Accepts regular expressions (192.168.*)

**Result:**

```
"result": { "ok": true }
```

## user.delBlacklist

Remove an address to an user blacklist

**Parameters:**

- `"user": <String>` - Username of the user
- `"ip": <String>` - IP address. Accepts regular expressions (192.168.*)

**Result:**

```
"result": { "ok": true }
```

## user.setMaxSessions

Set the maximum number of parallel sessions active of an user

**Parameters:**

- `"user": <String>` - Username of the user
- `"maxsessions": <Number>` - Number of maximum sessions

**Result:**

```
"result": { "ok": true }
```