

Métodos (API)

Indice:

- API
 - /stats
 - /ping
 - /api/get
 - /api/set
 - /api/node
 - /api/node/hash
 - /api/node/health
 - /api/node/lock
 - /api/node/unlock
 - /api/nodes
 - /api/nodes/cache
 - /api/nodes/alive
 - /api/nodes/alive/health
 - /api/nodes/fallen
 - /api/nodes/availables
 - /api/nodes/hash
 - /api/nodes/canonical
- Web Socket

/stats

Devuelve un conjunto de información relacionada al estado del servidor.

Respuesta

```
// http://localhost:4000/stats
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "health": {
      "details": {
        "mission_control": true,
        "mission_control_channel": true,
        "publisher": true,
        "scope": {
          "nodes": {
            "alive": [

            ],
            "fallen": [

            ]
          }
        }
      },
      "uptime": {
        "epoch": {
          "check": 1471385972.663555,
          "start": 1471385965.011688,
          "uptime": 7.651866912841797
        },
        "uptime": {
          "check": "2016-08-16T22:19:32.663555",
          "info": {
            "days": 0,
            "hours": 0,
            "minutes": 0,
            "seconds": 7
          },
          "start": "2016-08-16T22:19:25.011688"
        }
      }
    }
  },
}
```

```

        "status": true
    },
    "uptime": {
        "epoch": {
            "check": 1471385972.663574,
            "start": 1471385965.011688,
            "uptime": 7.651885986328125
        },
        "uptime": {
            "check": "2016-08-16T22:19:32.663574",
            "info": {
                "days": 0,
                "hours": 0,
                "minutes": 0,
                "seconds": 7
            },
            "start": "2016-08-16T22:19:25.011688"
        }
    }
}
}
}

```

/ping

Devuelve un mensaje predefinido en la configuración del servidor como respuesta a la petición.

Dentro del archivo `sinthius_octopus/settings.py` , se debe definir una constante `PING_RESPONSE` y asignarle un valor como `string` .

Respuesta

```

// http://localhost:4000/ping
// Content-Type:text/plain; charset=UTF-8

hello kitty, =^.^=

```

/api/get

Devuelve la información y estado de un `nodo` específico.

No soporta lista de nodos.

Argumentos:

- **node** (`string`) = Identificador del nodo
 - `ex: node:192.168.0.12:4001`

Respuesta

```
// http://localhost:4000/api/get?node=node:192.168.0.12:40
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "hash": "fd8c7bf493a1aa388199af8900d4517e8e16981c52a0d"
    "health": {
      "details": {
        "mission_control": true,
        "mission_control_channel": true,
        "publisher": true,
        "scope": {
          "nodes": {
            "alive": [
              "node:192.168.0.12:4000",
              "node:192.168.0.12:4002"
            ],
            "fallen": [

          ]
        }
      },
      "uptime": {
        "epoch": {
          "check": 1471389063.300029,
```

```

        "start": 1471389059.079493,
        "uptime": 4.22053599357605
    },
    "uptime": {
        "check": "2016-08-16T23:11:03.300029",
        "info": {
            "days": 0,
            "hours": 0,
            "minutes": 0,
            "seconds": 4
        },
        "start": "2016-08-16T23:10:59.079493"
    }
}
},
"status": true
},
"ip": "192.168.0.12",
"locked": false,
"mode": 0,
"name": null,
"node": "node:192.168.0.12:4001",
"port": 4001,
"priority": 0,
"updating": null
}
}

```

/api/set

Configura todos los `clientes` asociados a un `nodo`, con la información especificada en el argumento `data`.

La información debe estar formateada en `json` como un `string`. La misma no repercute dentro del `nodo`, este, sólo la propaga al resto de los `clientes` conectados vía `socket`.

Por lo general devolverá una respuesta vacía, en caso de `error` este

se verá reflejado en la respuesta.

Argumentos:

- **data** (string) = Configuración (json)
 - ex: `"{'data1': 'value', 'data2': [1, 2, 3]}"`

Respuesta

```
// http://localhost:4000/api/set?data=%27data%27:%27json-  
// Content-Type: application/json; charset=UTF-8  
  
{}
```

/api/node

Devuelve la información completa de la configuración del nodo .

Respuesta

```
// http://localhost:4000/api/node  
// Content-Type: application/json; charset=UTF-8  
  
{  
  "response": {  
    "hash": "5c0f5e5ae05372e5d3a233f6a4312c80ba247eb37841c",  
    "ip": "192.168.0.12",  
    "locked": false,  
    "mode": 0,  
    "name": null,  
    "node": "node:192.168.0.12:4000",  
    "port": 4000,  
    "priority": 0,  
    "updating": null  
  }  
}
```

```
}
```

/api/node/hash

Devuelve el valor `hash` del `nodo` .

Esto está pensado para una versión posterior que posee una capa de seguridad aplicada. De momento no posee utilidad alguna.

Respuesta

```
// http://localhost:4000/api/node/hash
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "hash": "5c0f5e5ae05372e5d3a233f6a4312c80ba247eb37841c"
  }
}
```

/api/node/health

Esta método es similar al `/stats` , salvo que se agrega la información completa de la `configuración` del `nodo` .

Respuesta

```
// http://localhost:4000/api/node/health
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "hash": "5c0f5e5ae05372e5d3a233f6a4312c80ba247eb37841c"
  }
}
```

```
"health": {
  "details": {
    "mission_control": true,
    "mission_control_channel": true,
    "publisher": true,
    "scope": {
      "nodes": {
        "alive": [
          "node:192.168.0.12:4001",
          "node:192.168.0.12:4002"
        ],
        "fallen": [

        ]
      }
    },
    "uptime": {
      "epoch": {
        "check": 1471391413.796157,
        "start": 1471388042.158525,
        "uptime": 3371.637631893158
      },
      "uptime": {
        "check": "2016-08-16T23:50:13.796157",
        "info": {
          "days": 0,
          "hours": 0,
          "minutes": 56,
          "seconds": 11
        },
        "start": "2016-08-16T22:54:02.158525"
      }
    }
  },
  "status": true
},
"ip": "192.168.0.12",
"locked": false,
"mode": 0,
"name": null,
"node": "node:192.168.0.12:4000",
```



```
    "port": 4000,  
    "priority": 0,  
    "updating": null  
  }  
}
```

/api/node/lock

Método que permite bloquear el `nodo` para que no pueda ser alterado.

Respuesta

```
// http://localhost:4000/api/node/lock  
// Content-Type: application/json; charset=UTF-8  
  
{  
  "response": {  
    "lock": true  
  }  
}
```

/api/node/unlock

Método que permite desbloquear el `nodo` para que pueda ser alterado.

Respuesta

```
// http://localhost:4000/api/node/unlock  
// Content-Type: application/json; charset=UTF-8  
  
{  
  "response": {
```

```
    "unlock": true
  }
}
```

/api/nodes

Devuelve una lista con los identificadores de todos los nodos que se encuentran como `alive` y `fallen`.

Respuesta

```
// http://localhost:4000/api/nodes
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "alive": {
      "nodes": [
        "node:192.168.0.12:4001",
        "node:192.168.0.12:4002"
      ],
      "total": 2
    },
    "fallen": {
      "nodes": [

      ],
      "total": 0
    }
  }
}
```

/api/nodes/cache

Devuelve una lista de todos los nodos, con la información completa

de la configuración de cada uno.

Respuesta

```
// http://localhost:4000/api/nodes/cache
// Content-Type: application/json; charset=UTF-8

{
  "response": [
    {
      "hash": "fd8c7bf493a1aa388199af8900d4517e8e16981c52a",
      "name": null,
      "ip": "192.168.0.12",
      "priority": 0,
      "mode": 0,
      "locked": false,
      "port": 4001,
      "updating": null
    },
    {
      "hash": "bbdb0ab689c05c7ba71423800ec19e6ae7fb0de3929",
      "name": null,
      "ip": "192.168.0.12",
      "priority": 0,
      "mode": 0,
      "locked": false,
      "port": 4002,
      "updating": null
    }
  ]
}
```

/api/nodes/alive

Devuelve una lista con los identificadores de todos los nodos que se encuentran como `alive`.

Respuesta

```
// http://localhost:4000/api/nodes/alive
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "alive": {
      "list": [
        "node:192.168.0.12:4001",
        "node:192.168.0.12:4002"
      ],
      "total": 2
    }
  }
}
```

/api/nodes/alive/health

Devuelve una lista con información y estado de todos los `nodos` que se encuentran como `alive`.

Respuesta

```
// http://localhost:4000/api/nodes/alive/health
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "alive": {
      "nodes": [
        {
          "hash": "fd8c7bf493a1aa388199af8900d4517e8e16981",
          "health": {
            "details": {
              "mission_control": true,
              "mission_control_channel": true,

```

```

    "publisher": true,
    "scope": {
      "nodes": {
        "alive": [
          "node:192.168.0.12:4000",
          "node:192.168.0.12:4002"
        ],
        "fallen": [

        ]
      }
    },
    "uptime": {
      "epoch": {
        "check": 1471391744.67777,
        "start": 1471389059.079493,
        "uptime": 2685.5982768535614
      },
      "uptime": {
        "check": "2016-08-16 23:55:44.677770",
        "info": {
          "days": 0,
          "hours": 0,
          "minutes": 44,
          "seconds": 45
        },
        "start": "2016-08-16 23:10:59.079493"
      }
    },
    "status": true
  },
  "ip": "192.168.0.12",
  "locked": false,
  "mode": 0,
  "name": null,
  "node": "node:192.168.0.12:4001",
  "port": 4001,
  "priority": 0,
  "updating": null
},

```

```

{
  "hash": "bbdb0ab689c05c7ba71423800ec19e6ae7fb0de
  "health": {
    "details": {
      "mission_control": true,
      "mission_control_channel": true,
      "publisher": true,
      "scope": {
        "nodes": {
          "alive": [
            "node:192.168.0.12:4000",
            "node:192.168.0.12:4001"
          ],
          "fallen": [

          ]
        }
      },
      "uptime": {
        "epoch": {
          "check": 1471391744.683141,
          "start": 1471389047.740745,
          "uptime": 2696.942395925522
        },
        "uptime": {
          "check": "2016-08-16 23:55:44.683141",
          "info": {
            "days": 0,
            "hours": 0,
            "minutes": 44,
            "seconds": 56
          },
          "start": "2016-08-16 23:10:47.740745"
        }
      },
      "status": true
    },
    "ip": "192.168.0.12",
    "locked": false,
    "mode": 0,

```

```
        "name": null,
        "node": "node:192.168.0.12:4002",
        "port": 4002,
        "priority": 0,
        "updating": null
    }
],
"total": 2
}
}
```

/api/nodes/fallen

Devuelve una lista con los `identificadores` de todos los `nodos` que se encuentran como `fallen` .

Respuesta

```
// http://localhost:4000/api/nodes/fallen
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "fallen": {
      "nodes": [

      ],
      "total": 0
    }
  }
}
```

/api/nodes/availables

Este método esta pensado para listar todo los nodos disponibles para ser configurados con el método `{node-ip-port}/api/set?data={ 'key': 'value' }`.

Devuelve una lista con la información completa de la configuración de cada nodos disponibles.

Respuesta

```
// http://localhost:4000/api/nodes/availables
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "availables": {
      "nodes": [
        [
          0,
          {
            "hash": "bbdb0ab689c05c7ba71423800ec19e6ae7fb0",
            "ip": "192.168.0.12",
            "locked": false,
            "mode": 0,
            "name": null,
            "port": 4002,
            "priority": 0,
            "updating": null
          }
        ],
        [
          0,
          {
            "hash": "fd8c7bf493a1aa388199af8900d4517e8e169",
            "ip": "192.168.0.12",
            "locked": false,
            "mode": 0,
            "name": null,
            "port": 4001,
            "priority": 0,
```



```

        "updating": null
      }
    ]
  ],
  "total": 2
}
}
}

```

/api/nodes/hash

Devuelve una lista con los hashes de todos los nodos que se encuentran como alive y fallen.

Respuesta

```

// http://localhost:4000/api/nodes/hash
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "alive": {
      "nodes": {
        "node:192.168.0.12:4001": "fd8c7bf493a1aa388199af8"
        "node:192.168.0.12:4002": "bbdb0ab689c05c7ba714238"
      },
      "total": 2
    },
    "fallen": {
      "nodes": {

      },
      "total": 0
    }
  }
}

```

/api/nodes/canonical

Devuelve una lista con los valores `ip` y `port` de todos los `nodos` que se encuentran como `alive` y `fallen`.

Respuesta

```
// http://localhost:4000/api/nodes/canonical
// Content-Type: application/json; charset=UTF-8

{
  "response": {
    "alive": {
      "nodes": {
        "node:192.168.0.12:4001": {
          "ip": "192.168.0.12",
          "port": 4001
        },
        "node:192.168.0.12:4002": {
          "ip": "192.168.0.12",
          "port": 4002
        }
      },
      "total": 2
    },
    "fallen": {
      "nodes": {
      },
      "total": 0
    }
  }
}
```

Web Socket

Por medio de una conexión vía `socket` , el `nodo` podrá notificarle a los `clientes` los cambios que se soliciten mediante el método `/ap/set` .

Ejemplo:

```
var ws = new WebSocket("ws://localhost:4000/ws/observer");

ws.onopen = function() {
    ws.send('{"key": "value"}');
};

ws.onmessage = function (evt) {
    alert(evt.data);
};
```