

# Census Control RESTful API: In Requests

*Note: Computation requests are may take several minutes, so they just return an `acknowledged` status with a string token for future reference, they are processed in a “first come first served” queue. Future error or success reports may follow the `acknowledged` response in the form of a web hook report, which is an HTTP request from Census Control to an external web server.*

*Note: If Census Control can't send messages to the web hook every out request will just log `WARNING: Unreachable HTTP hook server.`*

## POST /hook

Registers the server information of aa Web Hook, used by the `HTTPHook` module to send errors and success reports (HTTP messages) to a web server. (See HTTP API: Error and Success Reports)

### Request data

`host` : The web server host name.

`port` : The web server port.

```
{
  host: "10.4.22.123",
  port: 1000
}
```

### Response data

`status` : Value should be `success` , which means the hook server information is now registered, if json was invalid value should be `badrequest` .

Response status code `200` :

```
{
  status: "success"
}
```

Response status code `400` :

```
{
  status: "bad request",
  errors: [
    "'host' field missing.",
    "'port' field missing."
  ]
}
```

Response status code 500 :

```
{
  status: "unreachable host"
}
```

## GET /hook

Returns the server information of the web hook server that is registered.

### Request data

*No data needed.*

### Response data

host : The web server host name.

port : The web server port.

Response status code 200 :

```
{
  host: "10.4.22.123",
  port: 1000
}
```

## POST /compute

Enqueues a graph computation to be executed with a database graph.

### Request data

algorithm : The name of the algorithm that will be computed, must match the current graph format.

graph : An object with all the Neo4j database information.

`tag` : The Neo4j tag that will be used to import the nodes.  
`host` : The Neo4j hostname.  
`port` : The Neo4j port.  
`user` *Optional*: The desired user for the Census Engine instance to use.  
`password` *Optional*: The password of the Census Engine user.

```
{
  algorithm: "Closeness"
  graph: {
    tag: "Person",
    host: "10.3.10.123",
    port: 7474,
    user: "root",
    password: "root"
  }
}
```

## Response data

`status` : Value should be `acknowledged` , which only means the request was enqueued. Future out requests may report errors.  
`token` : A unique string id for the request, this is used to identify future reports.

Response status code 200:

```
{
  status: "acknowledged",
  token: "AJD23JSA941",
}
```

Response status code 400 :

```
{
  status: "bad request",
  errors: [
    "No such algorithm 'InexistentAlgo'",
    "'algorithm' field missing.",
    "'tag' field missing.",
    "'host' field missing.",
    "'port' field missing."
  ]
}
```

# Census Control RESTful API: Error and Success Reports

After registering a web hook, reports will be sent to the following paths:

## POST /censusengine/report

### Computation finished:

```
{
  token: "asd124-asdf-12351af-214",
  status: "success"
}
```

## POST /censusengine/error

### Unreachable Neo4j server:

```
{
  token: "asd124-asdf-12351af-214",
  status: "error",
  error: "unreachable-neo4j"
}
```

### Invalid Neo4j format:

```
{
  token: "asd124-asdf-12351af-214",
  status: "error",
  error: "invalid-neo4j-format"
}
```