The REST API is now versioned. For more information, see "About API versioning."

Management Console

Use the REST API to manage your GitHub Enterprise Server installation.

About the Management Console

You should explicitly set the port number when making API calls to the Management Console. If TLS is enabled on your enterprise, the port number is `8443`. Otherwise, the port number is `8080`.

If you cannot provide a port number, you'll need to configure your tool to automatically follow redirects.

You may also need to add the `-k` flag when using `curl`, since GitHub Enterprise Server uses a self-signed certificate before you add your own TLS certificate.

Authentication as the root site administrator

You need to pass your root site administrator password as an authentication token to every endpoint in this category except "Create a GitHub license."

Use the `api_key` parameter to send this token with each request. For example:

```
curl -L 'https://HOSTNAME:ADMIN-PORT/setup/api?api_key=YOUR_PASSWORD'
```

You can also use standard HTTP authentication to send this token. For example:

```
curl -L -u "api_key:YOUR_PASSWORD" 'https://HOSTNAME:ADMIN-PORT/setup/api'
```

Authentication as a Management Console user

Management Console user accounts can also authenticate to access this endpoint.

To authenticate with the password for a Management Console user account, use standard HTTP authentication. In the following example, replace YOUR_USER_NAME and YOUR_PASSWORD with the account's user name and password.

```
curl -L -u "YOUR_USER_NAME:YOUR_PASSWORD" 'https://HOSTNAME:ADMIN-PORT/setup/api'
```

Get the configuration status

This endpoint allows you to check the status of the most recent configuration process:

Note that you may need to wait several seconds after you start a process before you can check its status.

The different statuses are:

Status	Description
`PENDING`	The job has not started yet
`CONFIGURING`	The job is running
`DONE`	The job has finished correctly
`FAILED`	The job has finished unexpectedly

HTTP response status codes for "Get the configuration status"

Status code	Description
`200`	OK
`401`	Unauthorized

Code samples for "Get the configuration status"

GET /setup/api/configcheck

cURL

```
curl -L \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/configcheck
```

Response

Example response | Response schema

```
Status: 200
```

```
{ "status": "running", "progress": [ { "status": "DONE", "key": "Appliance core components" }, { "status": "DONE", "key": "GitHub utilities" }, { "status": "DONE", "key": "GitHub applications" }, { "status": "CONFIGURING", "key": "GitHub services" }, { "status": "PENDING", "key": "Reloading appliance services" } ] }
```

Start a configuration process

This endpoint allows you to start a configuration process at any time for your updated settings to take effect:

HTTP response status codes for "Start a configuration process"

Status code	Description
`202`	Accepted
`401`	Unauthorized

Code samples for "Start a configuration process"

POST `/setup/api/configure`

cURL

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/configure
```

Response

```
Status: 202
```

Get the maintenance status

Check your installation's maintenance status:

HTTP response status codes for "Get the maintenance status"

Status code	Description
`200`	OK
`401`	Unauthorized

Code samples for "Get the maintenance status"

GET `/setup/api/maintenance`

cURL

```
curl -L \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/maintenance
```

Response

Example response | Response schema

```
Status: 200
```

```
{ "status": "scheduled", "scheduled_time": "Tuesday, January 22 at 15:34 -0800", "connection_services": [ { "name": "git operations", "number": 0 }, { "name": "mysql queries", "number": 233 }, { "name": "aqueduct jobs", "number": 34 }, { "name": "resque jobs", "number": 54 } ] }
```

Enable or disable maintenance mode

Note: The request body for this operation must be submitted as `application/x-www-form-urlencoded` data. You can submit a parameter value as a string

Note: The request body for this operation must be submitted as `application/x-www-form-urlencoded` data. You can submit a parameter value as a string, or you can use a tool such as `curl` to submit a parameter value as the contents of a text file. For more information, see the [`curl` documentation].

Parameters for "Enable or disable maintenance mode"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Body parameters

`maintenance` string Required

A JSON string with the attributes `enabled` and `when` .
The possible values for `enabled` are `true` and `false` . When it's `false` , the attribute `when` is ignored and the maintenance mode is turned off. `when` defines the time period when the maintenance mode was enabled.
The possible values for `when` are `now` or any date parseable by [mojombo/chronic].

HTTP response status codes for "Enable or disable maintenance mode"

Status code	Description
`200`	OK
`401`	Unauthorized

Code samples for "Enable or disable maintenance mode"

POST /setup/api/maintenance

cURL

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/maintenance \ --data-urlencode "maintenance=
{"enabled":true, "when":"now"}"
```

Response

Example response | Response schema

Status: 200

```
{ "status": "scheduled", "scheduled_time": "Tuesday, January 22 at 15:34 -0800", "connection_services": [ { "name": "git operations", "number": 0 }, { "name": "mysql
queries", "number": 233 }, { "name": "aqueduct jobs", "number": 34 }, { "name": "resque jobs", "number": 54 } ] }
```

Get settings

Gets the settings for your instance. To change settings, see the [Set settings endpoint].

Note: You cannot retrieve the management console password with the Enterprise administration API.

HTTP response status codes for "Get settings"

Status code	Description
`200`	OK
`401`	Unauthorized

Code samples for "Get settings"

GET /setup/api/settings

cURL

```
curl -L \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/settings
```

Response

Example response | Response schema

Status: 200

```
{ "enterprise": { "private_mode": false, "public_pages": false, "subdomain_isolation": true, "signup_enabled": false, "github_hostname": "ghe.local", "identicons_host":
"dotcom", "http_proxy": null, "auth_mode": "default", "expire_sessions": false, "admin_password": null, "configuration_id": 1401777404, "configuration_run_count": 4,
"avatar": { "enabled": false, "uri": "" }, "customer": { "name": "GitHub", "email": "stannis@themannis.biz", "uuid": "af6cac80-e4e1-012e-d822-1231380e52e9",
"secret_key_data": "-----BEGIN PGP PRIVATE KEY BLOCK-----\nVersion: GnuPG v1.4.10 (GNU/Linux)\n\nlQcYBE5TCgsBEACk4yHpUcapplebaumBMXYMiLF+nCQ0lxpx...\n-----END PGP PRIVATE
KEY BLOCK-----\n", "public_key_data": "-----BEGIN PGP PUBLIC KEY BLOCK-----\nVersion: GnuPG v1.4.10 (GNU/Linux)\n\nmI0ETqzZYgEEALSe6snowdenXyqvLfSQ34HWD6C7....\n-----END
PGP PUBLIC KEY BLOCK-----\n" }, "license": { "seats": 0, "evaluation": false, "perpetual": false, "unlimited_seating": true, "support_key": "ssh-rsa AAAAB3N....",
```

Set settings

Applies settings on your instance. For a list of the available settings, see the Get settings endpoint.

Notes:

- The request body for this operation must be submitted as `application/x-www-form-urlencoded` data. You can submit a parameter value as a string, or you can use a tool such as `curl` to submit a parameter value as the contents of a text file. For more information, see the `curl` documentation.
- You cannot set the management console password with the Enterprise administration API. Use the `ghe-set-password` utility to change the management console password. For more information, see "Command-line utilities."

Parameters for "Set settings"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Body parameters

`settings` string Required

A JSON string with the new settings. Note that you only need to pass the specific settings you want to modify. For a list of the available settings, see the Get settings endpoint.

HTTP response status codes for "Set settings"

Status code	Description
`204`	No Content
`401`	Unauthorized

Code samples for "Set settings"

PUT /setup/api/settings

cURL

```
curl -L \ -X PUT \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/settings \ --data-urlencode "settings={
"enterprise": { "public_pages": true }}"
```

Response

```
Status: 204
```

Get all authorized SSH keys

HTTP response status codes for "Get all authorized SSH keys"

Status code	Description
`200`	OK
`401`	Unauthorized

Code samples for "Get all authorized SSH keys"

GET /setup/api/settings/authorized-keys

cURL

```
curl -L \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/settings/authorized-keys
```

Response

Example response | Response schema

```
Status: 200
```

```
[ { "key": "ssh-rsa AAAAB3NzaC1yc2EAAAAB...", "pretty-print": "ssh-rsa 01:14:0f:f2:0f:e2:fe:e8:f4:72:62:af:75:f7:1a:88:3e:04:92:64" }, { "key": "ssh-rsa
AAAAB3NzaC1yc2EAAAAB...", "pretty-print": "ssh-rsa 01:14:0f:f2:0f:e2:fe:e8:f4:72:62:af:75:f7:1a:88:3e:04:92:64" }, { "key": "ssh-rsa AAAAB3NzaC1yc2EAAAAB...", "pretty-
print": "ssh-rsa 01:14:0f:f2:0f:e2:fe:e8:f4:72:62:af:75:f7:1a:88:3e:04:92:64" } ]
```

Add an authorized SSH key

Note: The request body for this operation must be submitted as `application/x-www-form-urlencoded` data. You can submit a parameter value as a string, or you can use a tool such as `curl` to submit a parameter value as the contents of a text file. For more information, see the `curl` documentation.

Parameters for "Add an authorized SSH key"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Body parameters

`authorized_key` string Required

The public SSH key.

HTTP response status codes for "Add an authorized SSH key"

Status code	Description
`201`	Created
`401`	Unauthorized

Code samples for "Add an authorized SSH key"

POST `/setup/api/settings/authorized-keys`

cURL

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/settings/authorized-keys \ --data-urlencode "authorized_key=ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAAgQCssTL/Vtu/ODLTj0VtZoRAbvf7uiv5997GyDq0MoAZUjb5jmA5wYe2/wF6sFuhiZTnZoF1ZtCHunPp0hM/GHrn6VySBhNncx14YO8FPt1CIhEeRMSEjUK9cY3xAbS365oXY8vnUHJsS9+1tr/2bx/+4NJfcUt/Ezf1
```

Response

Example response | Response schema

```
Status: 201
```

```
[ { "key": "ssh-rsa AAAAB3NzaC1yc2EAAAAB...", "pretty-print": "ssh-rsa 01:14:0f:f2:0f:e2:fe:e8:f4:72:62:af:75:f7:1a:88:3e:04:92:64" }, { "key": "ssh-rsa AAAAB3NzaC1yc2EAAAAB...", "pretty-prir
01:14:0f:f2:0f:e2:fe:e8:f4:72:62:af:75:f7:1a:88:3e:04:92:64" }, { "key": "ssh-rsa AAAAB3NzaC1yc2EAAAAB...", "pretty-print": "ssh-rsa 01:14:0f:f2:0f:e2:fe:e8:f4:72:62:af:75:f7:1a:88:3e:04:92:64
```

Remove an authorized SSH key

Note: The request body for this operation must be submitted as `application/x-www-form-urlencoded` data. You can submit a parameter value as a string, or you can use a tool such as `curl` to submit a parameter value as the contents of a text file. For more information, see the `curl` documentation.

Parameters for "Remove an authorized SSH key"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Body parameters

`authorized_key` string Required

The public SSH key.

HTTP response status codes for "Remove an authorized SSH key"

Status code	Description
`200`	OK
`401`	Unauthorized

Code samples for "Remove an authorized SSH key"

DELETE `/setup/api/settings/authorized-keys`

cURL

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/settings/authorized-keys \ --data-urlencode "authorized_key=ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAAgQCssTL/Vtu/ODLTj0VtZoRAbvf7uiv5997GyDq0MoAZUjb5jmA5wYe2/wF6sFuhiZTnZoF1ZtCHunPp0hM/GHrn6VySBhNncx14YO8FPt1CIhEeRMSEjUK9cY3xAbS365oXY8vnUHJsS9+1tr/2bx/+4NJfcUt/Ezf1
```

Response

Example response | Response schema

Status: 200

```
[ { "key": "ssh-rsa AAAAB3NzaC1yc2EAAAAB...", "pretty-print": "ssh-rsa 01:14:0f:f2:0f:e2:fe:e8:f4:72:62:af:75:f7:1a:88:3e:04:92:64" }, { "key": "ssh-rsa AAAAB3NzaC1yc2EAAAAB...", "pretty-prir
01:14:0f:f2:0f:e2:fe:e8:f4:72:62:af:75:f7:1a:88:3e:04:92:64" }, { "key": "ssh-rsa AAAAB3NzaC1yc2EAAAAB...", "pretty-print": "ssh-rsa 01:14:0f:f2:0f:e2:fe:e8:f4:72:62:af:75:f7:1a:88:3e:04:92:64
```

Create a GitHub license

When you boot a GitHub instance for the first time, you can use the following endpoint to upload a license.

Note that you need to `POST` to [`/setup/api/configure`](/setup/api/configure) to start the actual configuration process.

When using this endpoint, your GitHub instance must have a password set. This can be accomplished two ways:

1. If you're working directly with the API before accessing the web interface, you must pass in the password parameter to set your password.
2. If you set up your instance via the web interface before accessing the API, your calls to this endpoint do not need the password parameter.

Note: The request body for this operation must be submitted as `multipart/form-data` data. You can can reference the license file by prefixing the filename with the `@` symbol using `curl`. For more information, see the `curl` documentation.

Parameters for "Create a GitHub license"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Body parameters

`license` string Required

The content of your *.ghl* license file.

`password` string

You **must** provide a password *only if* you are uploading your license for the first time. If you previously set a password through the web interface, you don't need this parameter.

`settings` string

An optional JSON string containing the installation settings. For a list of the available settings, see the Get settings endpoint.

HTTP response status codes for "Create a GitHub license"

Status code	Description
`202`	Accepted
`401`	Unauthorized

Code samples for "Create a GitHub license"

POST `/setup/api/start`

cURL

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/start \ --form "license=@enterprise.ghl" --form
```

```
"password=secret"
```

Response

```
Status: 202
```

Upgrade a license

This API upgrades your license and also triggers the configuration process.

Note: The request body for this operation must be submitted as `multipart/form-data` data. You can can reference the license file by prefixing the filename with the `@` symbol using `curl`. For more information, see the `curl` documentation.

Parameters for "Upgrade a license"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Body parameters

`license` string

The content of your new *.ghl* license file.

HTTP response status codes for "Upgrade a license"

Status code	Description
`202`	Accepted
`401`	Unauthorized

Code samples for "Upgrade a license"

POST /setup/api/upgrade

cURL

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -u "api_key:your-password" \ http(s)://HOSTNAME/setup/api/upgrade \ --form "license=@enterprise.ghl"
```

Response

```
Status: 202
```