

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

## GPG Keys

Use the REST API to manage GPG keys of authenticated users.

### About user GPG key administration

The data returned in the `public_key` response field is not a GPG formatted key. When a user uploads a GPG key, it is parsed and the cryptographic public key is extracted and stored. This cryptographic key is what the endpoints in this category will return. This key is not suitable for direct use in programs such as GPG.

If a request URL does not include a `{username}` parameter then the response will be for the signed-in user (and you must pass [authentication information](#) with your request). Additional private information, such as whether a user has two-factor authentication enabled, is included when authenticated through OAuth with the `user` scope.

### List GPG keys for the authenticated user

Lists the current user's GPG keys. Requires that you are authenticated via Basic Auth or via OAuth with at least `read:pgp_key` [scope](#).

#### Parameters for "List GPG keys for the authenticated user"

##### Headers

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

##### Query parameters

**per\_page** integer  
The number of results per page (max 100).  
Default: `30`

**page** integer  
Page number of the results to fetch.  
Default: `1`

#### HTTP response status codes for "List GPG keys for the authenticated user"

Status code	Description
200	OK
304	Not modified
401	Requires authentication
403	Forbidden
404	Resource not found

#### Code samples for "List GPG keys for the authenticated user"

**GET** /user/gpg\_keys

cURLJavaScriptGitHub CLI

```
curl -L -H "Accept: application/vnd.github+json" \
-H "Authorization: Bearer <YOUR-TOKEN>" \
-H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/user/gpg_keys
```

#### Response

Example responseResponse schema

Status: 200

```
[ { "id": 3, "name": "Octocat's GPG Key", "primary_key_id": 2, "key_id": "3262EFF25BA0D270", "public_key": "xsBNBFayYZ...", "emails": [ { "email": "octocat@users.noreply.github.com", "verified": true } ], "subkeys": [ { "id": 4, "primary_key_id": 3, "key_id": "4A595D4C72EE49C7", "public_key": "zsBNBFayYZ...", "emails": [], "can_sign": false, "can_encrypt_comms": true, "can_encrypt_storage": true, "can_certify": false, "created_at": "2016-03-24T11:31:04-06:00", "expires_at": "2016-03-24T11:31:04-07:00", "revoked": false } ], "can_sign": true, "can_encrypt_comms": false, "can_encrypt_storage": false, "can_certify": true, "created_at": "2016-03-24T11:31:04-06:00", "expires_at": "2016-03-24T11:31:04-07:00", "revoked": false, "raw_key": "string" } ]
```

## Create a GPG key for the authenticated user 🔑

Adds a GPG key to the authenticated user's GitHub account. Requires that you are authenticated via Basic Auth, or OAuth with at least `write:gpg_key scope`.

### Parameters for "Create a GPG key for the authenticated user"

#### Headers

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

#### Body parameters

`name` string  
A descriptive name for the new key.

`armored_public_key` string **Required**  
A GPG key in ASCII-armored format.

### HTTP response status codes for "Create a GPG key for the authenticated user"

Status code	Description
201	Created
304	Not modified
401	Requires authentication
403	Forbidden
404	Resource not found
422	Validation failed, or the endpoint has been spammed.

### Code samples for "Create a GPG key for the authenticated user"

POST

/user/gpg\_keys

cURL

JavaScript

GitHub CLI

```
curl -L \
-X POST \
-H "Accept: application/vnd.github+json" \
-H "Authorization: Bearer <YOUR-TOKEN>" \
-H "X-GitHub-API-Version: 2022-11-28" \
https://api.github.com/user/gpg_keys \
-d '{"name":
```

#### Response

Example response

Response schema

Status: 201

```
{ "id": 3, "name": "Octocat's GPG Key", "primary_key_id": 2, "key_id": "3262EFF25BA0D270", "public key": "xsBNBFayYZ...", "emails": [ { "email": "octocat@users.noreply.github.com", "verified": true } ], "armored_public_key": "-----BEGIN PGP PUBLIC KEY BLOCK-----\n\nmQENBFayYZ8BCAC4hScoJXXpyR+MXGcrBxEIqw3FzCVvkViuyeko+Jp76QJhg8kr\n\nnucRTxbn0Hfda/FmILeA/wxf9ch5/PSrrL26FxEOpHhJolp8fnIDLQeITn94NYdB\n\nnZtnnEKslpPrG97qSUWIchvyqCptv0b8+8fwGx9K/ZwcEEdh1X8+WFR\n\n-----END PGP PUBLIC KEY BLOCK-----\n" }
```

## Get a GPG key for the authenticated user 🔑

View extended details for a single GPG key. Requires that you are authenticated via Basic Auth or via OAuth with at least `read:gpg_key scope`.

### Parameters for "Get a GPG key for the authenticated user"

#### Headers

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

#### Path parameters

`gpg_key_id` integer **Required**  
The unique identifier of the GPG key.

### HTTP response status codes for "Get a GPG key for the authenticated user"

Status code	Description
200	OK

304	Not modified
401	Requires authentication
403	Forbidden
404	Resource not found

Code samples for "Get a GPG key for the authenticated user"

GET

/user/gpg\_keys/{gpg\_key\_id}

cURL

JavaScript

GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ https://api.github.com/user/gpg_keys/GPG_KEY_ID
```

Response

Example response

Response schema

Status: 200

```
{ "id": 3, "name": "Octocat's GPG Key", "primary_key_id": 2, "key_id": "3262EFF25BA0D270", "public_key": "xsBNBFayYZ...", "emails": [ { "email": "octocat@users.noreply.github.com", "verified": true } ], "url": "https://api.github.com/user/gpg_keys/3" }
```

Delete a GPG key for the authenticated user

Removes a GPG key from the authenticated user's GitHub account. Requires that you are authenticated via Basic Auth or via OAuth with at least `admin:gpg_key` [scope](#).

Parameters for "Delete a GPG key for the authenticated user"

Headers

accept

string

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

Path parameters

gpg\_key\_id

integer

Required

The unique identifier of the GPG key.

HTTP response status codes for "Delete a GPG key for the authenticated user"

Status code	Description
204	No Content
304	Not modified
401	Requires authentication
403	Forbidden
404	Resource not found
422	Validation failed, or the endpoint has been spammed.

Code samples for "Delete a GPG key for the authenticated user"

DELETE

/user/gpg\_keys/{gpg\_key\_id}

cURL

JavaScript

GitHub CLI

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ https://api.github.com/user/gpg_keys/GPG_KEY_ID
```

Response

Status: 204

List GPG keys for a user

Lists the GPG keys for a user. This information is accessible by anyone.

Parameters for "List GPG keys for a user"

Headers

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

Path parameters

**username** string **Required**  
The handle for the GitHub user account.

Query parameters

**per\_page** integer  
The number of results per page (max 100).  
Default: `30`

**page** integer  
Page number of the results to fetch.  
Default: `1`

HTTP response status codes for "List GPG keys for a user"

Status code	Description
200	OK

Code samples for "List GPG keys for a user"

GET

/users/{username}/gpg\_keys

cURL

JavaScript

GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/users/USERNAME/gpg_keys
```

Response

Example response

Response schema

Status: 200

```
[ { "id": 3, "name": "Octocat's GPG Key", "primary_key_id": 2, "key_id": "3262EFF25BA0D270", "public_key": "xsBNBFayYZ...", "emails": [ { "email": "octocat@users.noreply.github.com", "verified": true } ], "subkeys": [ { "id": 4, "primary_key_id": 3, "key_id": "4A595D4C72EE49C7", "public_key": "zsBNBFayYZ...", "emails": [], "can_sign": false, "can_encrypt_comms": true, "can_encrypt_storage": true, "can_certify": false, "created_at": "2016-03-24T11:31:04-06:00", "expires_at": "2016-03-24T11:31:04-07:00", "revoked": false } ], "can_sign": true, "can_encrypt_comms": false, "can_encrypt_storage": false, "can_certify": true, "created_at": "2016-03-24T11:31:04-06:00", "expires_at": "2016-03-24T11:31:04-07:00", "revoked": false, "raw_key": "string" } ]
```

Legal