

Server Script

[✎ Edit](#) ▾

A Server Script lets you dynamically define a Python Script that is executed on the server on a document event or API

1. How to create a Server Script

To create a Server Script

1. If your site is being hosted on erpnext.com, contact support to activate Server Script. In case of self-hosted accounts, set `server_script_enabled` as true in `site_config.json` of your site.
2. To add/edit Server Script, ensure your role is System Manager.
3. Type "New Server Script" in the awesomebar and hit enter to create a new Server Script document.
4. Set the type of server script (Document Event / API).
5. Set the document type and event name, or method name, script and save.

2. Features

2.1 Enabling Server Script

Server script must be enabled via `site_config.json`

```
bench --site site1.local set-config server_script_enabled true
```

2.2 Document Events

For scripts that are to be called via document events, you must set the Reference Document Type and Event Name to define the trigger

- Before Insert
- After Insert
- Before Validate
- Before Save

- After Save
- Before Submit
- After Submit
- Before Cancel
- After Cancel
- Before Delete
- After Delete
- Before Save (Submitted Document)
- After Save (Submitted Document)

2.3 API Scripts

API endpoints can be created on the fly by using the **Script Type** `"API"`. The name of the endpoint depends on field **API Method**. All APIs created using Server Scripts will be automatically prefixed with `/api/method`.

For instance, a script with the **API Method** `"delete-note"` may be accessed via `/api/method/delete-note`. Using Frappe's frontend request library, you could use `frappe.call("delete-note")` in your client scripts.

Guest access may be enabled by checking **Allow Guest** for the created APIs. The response can be set via `frappe.response["message"]` object.

2.3 Security

Frappe Framework uses the RestrictedPython library to restrict access to methods available for server scripts. Only the safe methods, listed below are available in server scripts.

For allowed methods, see [Script API](#)

2.4 Using Server Scripts as libraries

You can use a server script as an internal method by setting `frappe.flags` value in script.

2.5 Comparing changes

You can diff two versions of server scripts using "Compare Versions" button.

Compare Versions ✕

From version *

6472ebc6d6

To version *

ad6b97638a

```

--- 6472ebc6d6 2021-10-23 12:13:59.127885
+++ ad6b97638a 2021-10-23 12:14:03.151547
@@ -1 +1 @@
-doc.docstatus = 2
+doc.docstatus = 42

```

diff

3. Examples

3.1 Change the value of a property before change

Script Type: Before Save

```

if "test" in doc.description:
    doc.status = 'Closed'

```

PY

3.2 Custom validation

Script Type: "Before Save"

```

if "validate" in doc.description:
    raise frappe.ValidationError

```

PY

3.3. Auto Create To Do

Script Type: "After Save"

```

if doc.allocated_to:
    frappe.get_doc(dict(
        doctype = 'ToDo',
        owner = doc.allocated_to,

```

PY

```
description = doc.subject
)).insert()
```

3.4 API

- Script Type: API
- Method Name: `test_method`

```
frappe.response['message'] = "hello"
```

PY

Request: `/api/method/test_method`

3.5 Internal Library

“New in version 13”

Call one (`script_1` script from another `script_2`)

Script 1:

```
frappe.flags.my_key = 'my value'
```

PY

Script 2:

```
my_key = run_script('script_1').get('my_key')
```



PREVIOUS PAGE
System Console

NEXT PAGE
Script API



Last updated 4 months ago

Was this helpful?

