Commit 27f1343

Browse files

akhilnarang authored on Dec 24, 2024 · 1 / 1 · Verified

```
Merge pull request #28897 from frappe/version-15-hotfix
chore: release v15
```

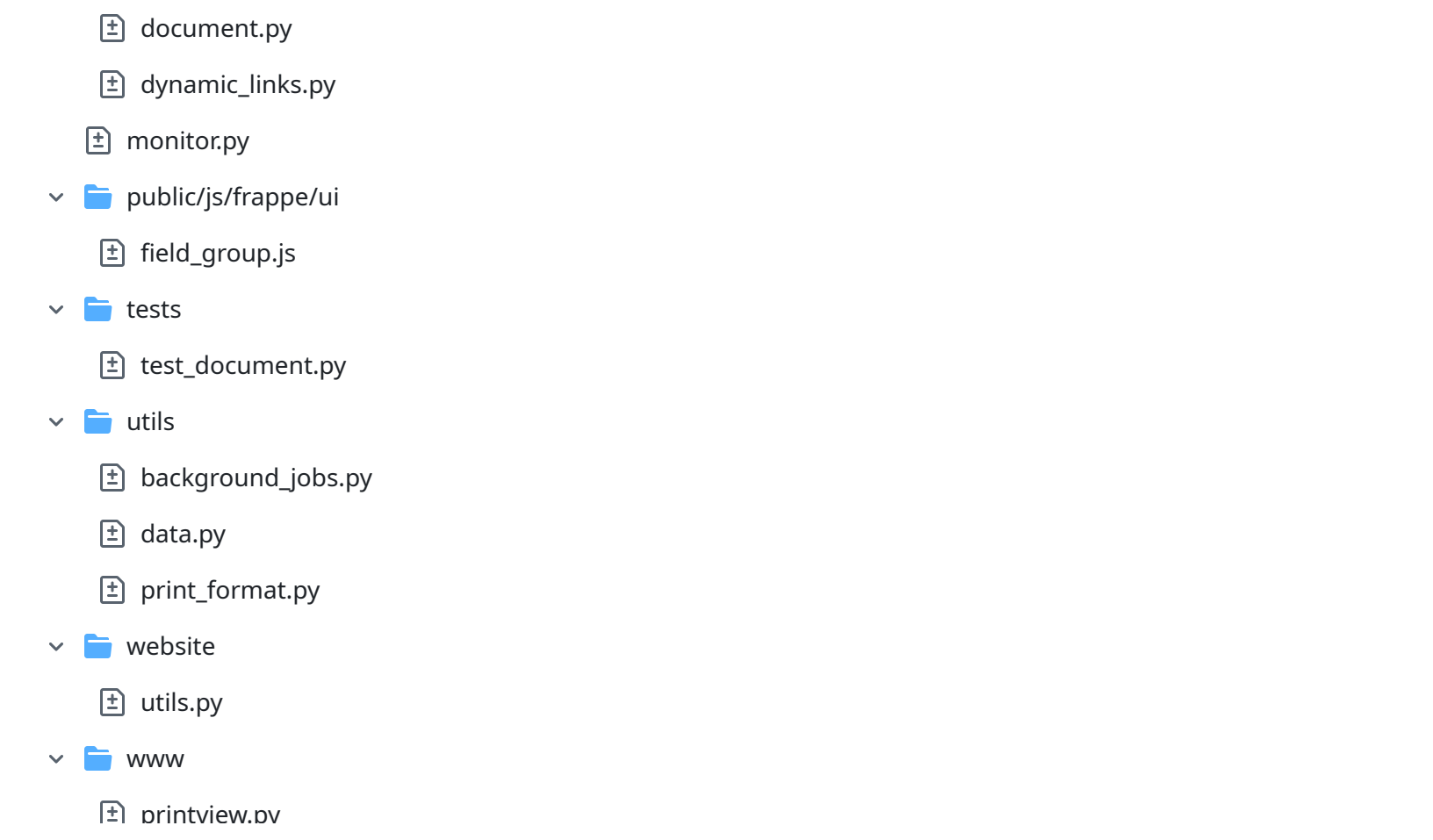
mergify/bp/version-15/pr-31127 (#28897) + version-15 (ikrokdev/frappe-crocrm#3, #28897)

v15.61.0 · ••• v15.51.0

2 parents 93ee9ca + 7117a12 commit 27f1343

Filter files...

- frappe
  - build.py
  - core/doctype/prepared_report
    - prepared_report.json
    - prepared_report.py
  - database
    - db_manager.py
  - desk
    - form
      - load.py
    - page/setup_wizard
      - setup_wizard.py
  - geo
    - country_info.json
  - hooks.py
  - model
    - base_document.py
    - db_query.py



20 files changed +108 -22 lines changed

Search within code

frappe/build.py +1 -1

```
@@ -197,7 +197,7 @@ def symlink(target, link_name, overwrite=False):
197 197                     if os.path.isdir(link_name):
198 198                             raise IsADirectoryError(f"Cannot symlink over existing directory:
            '{link_name}'")
199 199                     try:
200     -                       os.replace(temp_link_name, link_name)
    200 +                       shutil.move(temp_link_name, link_name)
201 201                     except AttributeError:
202 202                             os.renames(temp_link_name, link_name)
203 203             except Exception:
```

frappe/core/doctype/prepared_report/prepared_report.json +9 -1

```
@@ -13,6 +13,7 @@
13 13       "column_break_4",
14 14       "queued_at",
15 15       "report_end_time",
   16 +     "peak_memory_usage",
16 17       "section_break_7",
```

```
17  18      "error_message",
18  19      "filters_sb",
        @@ -101,11 +102,18 @@
101 102      "is_virtual": 1,
102 103      "label": "Queued At",
103 104      "read_only": 1
    105 +   },
    106 +   {
    107 +    "fieldname": "peak_memory_usage",
    108 +    "fieldtype": "Int",
    109 +    "label": "Peak Memory Usage",
    110 +    "print_hide": 1,
    111 +    "read_only": 1
104 112     }
105 113    ],
106 114    "in_create": 1,
107 115    "links": [],
108     -  "modified": "2024-03-07 12:01:58.209879",
    116 +  "modified": "2024-12-20 10:18:19.174608",
109 117    "modified_by": "Administrator",
110 118    "module": "Core",
111 119    "name": "Prepared Report",
```

frappe/core/doctype/prepared_report/prepared_report.py +4

```
        @@ -2,6 +2,7 @@
2   2    # License: MIT. See LICENSE
3   3    import gzip
4   4    import json
    5 + import resource
5   6    from contextlib import suppress
6   7    from typing import Any
7   8
        @@ -33,6 +34,7 @@ class PreparedReport(Document):
33  34                   error_message: DF.Text | None
34  35                   filters: DF.SmallText | None
35  36                   job_id: DF.Data | None
    37 +                 peak_memory_usage: DF.Int
36  38                   queued_at: DF.Datetime | None
37  39                   queued_by: DF.Data | None
38  40                   report_end_time: DF.Datetime | None
```

```
@@ -119,6 +121,8 @@ def generate_report(prepared_report):
                _save_error(instance, error=frappe.get_traceback(with_context=True))

        instance.report_end_time = frappe.utils.now()
+       instance.peak_memory_usage = resource.getrusage(resource.RUSAGE_SELF).ru_maxrss
+       add_data_to_monitor(peak_memory_usage=instance.peak_memory_usage)
        instance.save(ignore_permissions=True)

        frappe.publish_realtime(
```

frappe/database/db_manager.py +3

```
@@ -81,6 +81,9 @@ def restore_database(verbose: bool, target: str, source: str, user: str, passwor
                # Newer versions of MariaDB add in a line that'll break on older versions, so remove it
                command.extend(["sed", r"'/\/\*M\{0,1\}!999999\\- enable the sandbox mode \*\//d'", "|"])

+               # Remove view security definers
+               command.extend(["sed", r"'/\/\*![0-9]* DEFINER=[^ ]* SQL SECURITY DEFINER \*\//d'", "|"])
+
                # Generate the restore command
                bin, args, bin_name = get_command(
                        socket=frappe.conf.db_socket,
```

frappe/desk/form/load.py +2

```
@@ -14,6 +14,7 @@
from frappe.model.utils.user_settings import get_user_settings
from frappe.permissions import get_doc_permissions
from frappe.utils.data import cstr
+ from frappe.utils.html_utils import clean_email_html

if typing.TYPE_CHECKING:
        from frappe.model.document import Document

@@ -258,6 +259,7 @@ def _get_communications(doctype, name, start=0, limit=20):
        communications = get_communication_data(doctype, name, start, limit)
        for c in communications:
                if c.communication_type in ("Communication", "Automated Message"):
```

```
                                clean_email_html(c.content)
                        c.attachments = json.dumps(
                                frappe.get_all(
                                        "File",
```

frappe/desk/page/setup_wizard/setup_wizard.py +14 -1

```
@@ -50,7 +50,7 @@ def setup_complete(args):
        if cint(frappe.db.get_single_value("System Settings", "setup_complete")):
                return {"status": "ok"}

-       args = parse_args(args)
+       args = parse_args(sanitize_input(args))
        stages = get_setup_stages(args)
        is_background_task = frappe.conf.get("trigger_site_setup_in_background")

@@ -253,6 +253,19 @@ def parse_args(args): # nosemgrep
        return args


+ def sanitize_input(args):
+       from frappe.utils import is_html, strip_html_tags
+
+       if isinstance(args, str):
+               args = json.loads(args)
+
+       for key, value in args.items():
+               if is_html(value):
+                       args[key] = strip_html_tags(value)
+
+       return args
+
+
def add_all_roles_to(name):
        user = frappe.get_doc("User", name)
        user.append_roles(*_get_default_roles())
```

frappe/geo/country_info.json +2

```
@@ -2599,6 +2599,8 @@
  "currency_name": "Swedish Krona",
  "currency_symbol": "kr",
```

```
2601  2601      "number_format": "#.###,##",
      2602  +   "date_format": "yyyy-mm-dd",
      2603  +   "time_format": "HH:mm",
2602  2604      "timezones": [
2603  2605       "Europe/Stockholm"
2604  2606      ],
```

frappe/hooks.py +1

```
      @@ -570,4 +570,5 @@
570   570           "insert_queue_for_*",  # Deferred Insert
571   571           "recorder-*",  # Recorder
572   572           "global_search_queue",
      573   +       "monitor-transactions",
573   574   ]
```

frappe/model/base_document.py +4 -2

```
      @@ -18,6 +18,7 @@
18    18            table_fields,
19    19    )
20    20    from frappe.model.docstatus import DocStatus
      21    + from frappe.model.dynamic_links import invalidate_distinct_link_doctypes
21    22    from frappe.model.naming import set_new_name
22    23    from frappe.model.utils.link_count import notify_link_count
23    24    from frappe.modules import load_doctype_module

      @@ -784,6 +785,7 @@ def get_msg(df, docname):
784   785                                       doctype = self.get(df.options)
785   786                                       if not doctype:
786   787                                               frappe.throw(_("{0} must be set first").format(self.meta.get_label(df.options)))
      788   +                                   invalidate_distinct_link_doctypes(df.parent, df.options, doctype)
787   789
788   790                                   # MySQL is case insensitive. Preserve case of the original docname in the Link Field.
789   791

      @@ -1270,11 +1272,11 @@ def get_value(self, fieldname):
1270  1272          def cast(self, value, df):
1271  1273                  return cast_fieldtype(df.fieldtype, value, show_warning=False)
1272  1274
1273        -     def _extract_images_from_text_editor(self):
```

```
      1275  +         def _extract_images_from_editor(self):
1274  1276                    from frappe.core.doctype.file.utils import extract_images_from_doc
1275  1277
1276  1278                    if self.doctype != "DocType":
1277        -                         for df in self.meta.get("fields", {"fieldtype": ("=", "Text
                Editor")}):
      1279  +                         for df in self.meta.get("fields", {"fieldtype": ("in", ("Text
                Editor", "HTML Editor"))}):
1278  1280                                  extract_images_from_doc(self, df.fieldname)
1279  1281
1280  1282
```

frappe/model/db_query.py +2 -1

```
      @@ -28,6 +28,7 @@
28    28            get_time,
29    29            get_timespan_date_range,
30    30            make_filter_tuple,
      31  +         sanitize_column,
31    32    )
32    33    from frappe.utils.data import DateTimeLikeObject, get_datetime, getdate, sbool
33    34
      @@ -596,7 +597,7 @@ def build_filter_conditions(self, filters, conditions: list,
      ignore_permissions=
596   597
597   598                   for f in filters:
598   599                           if isinstance(f, str):
599       -                                 conditions.append(f)
      600 +                                 conditions.append(sanitize_column(f))
600   601                           else:
601   602                                   conditions.append(self.prepare_filter_condition(f))
602   603
```

frappe/model/document.py +2 -2

```
      @@ -591,7 +591,7 @@ def _validate(self):
591   591                   self._fix_rating_value()
592   592                   self._validate_code_fields()
593   593                   self._sync_autoname_field()
594       -                 self._extract_images_from_text_editor()
      594 +                 self._extract_images_from_editor()
595   595                   self._sanitize_content()
596   596                   self._save_passwords()
```

```
597  597                    self.validate_workflow()
     @@ -604,7 +604,7 @@ def _validate(self):
604  604                            d._fix_rating_value()
605  605                            d._validate_code_fields()
606  606                            d._sync_autoname_field()
607       -                        d._extract_images_from_text_editor()
     607  +                        d._extract_images_from_editor()
608  608                            d._sanitize_content()
609  609                            d._save_passwords()
610  610                    if self.is_new():
```

frappe/model/dynamic_links.py +39 -3

```
     @@ -42,9 +42,7 @@ def get_dynamic_link_map(for_delete=False):
42   42                                 dynamic_link_map.setdefault(meta.name, []).append(df)
43   43                         else:
44   44                                 try:
45       -                                 links = frappe.db.sql_list(
46       -                                         """select distinct `{options}` from
         `tab{parent}`""".format(**df)
47       -                                 )
     45  +                                 links = fetch_distinct_link_doctypes(df.parent,
         df.options)
48   46                                 for doctype in links:
49   47                                         dynamic_link_map.setdefault(doctype,
         []).append(df)
50   48                                 except frappe.db.TableMissingError:
     @@ -61,3 +59,41 @@ def get_dynamic_links():
61   59                 for query in dynamic_link_queries:
62   60                         df += frappe.db.sql(query, as_dict=True)
63   61                 return df
     62  +
     63  +
     64  + def _dynamic_link_map_key(doctype, fieldname):
     65  +         return f"dynamic_link_map::{doctype}::{fieldname}"
     66  +
     67  +
     68  + def fetch_distinct_link_doctypes(doctype: str, fieldname: str):
     69  +         """Return all unique doctypes a dynamic link is linking against.
     70  +         Note:
     71  +         - results are cached and can *possibly be outdated*
     72  +         - cache gets updated when a document with different document link is discovered
     73  +         - raw queries adding dynamic link won't update this cache
```

```
74  +         - cache miss can often be VERY expensive on large table.
75  +         """
76  +
77  +         key = _dynamic_link_map_key(doctype, fieldname)
78  +         doctypes = frappe.cache.get_value(key)
79  +
80  +         if doctypes is None:
81  +                 doctypes = frappe.db.sql(f"""select distinct `{fieldname}` from
      `tab{doctype}`""", pluck=True)
82  +                 frappe.cache.set_value(key, doctypes, expires_in_sec=12 * 60 * 60)
83  +
84  +         return doctypes
85  +
86  +
87  + def invalidate_distinct_link_doctypes(doctype: str, fieldname: str, linked_doctype: str):
88  +         """If new linked doctype is discovered for a dynamic link then cache is
      evicted."""
89  +
90  +         key = _dynamic_link_map_key(doctype, fieldname)
91  +         doctypes = frappe.cache.get_value(key)
92  +
93  +         if doctypes is None or not isinstance(doctypes, list):
94  +                 return
95  +
96  +         if linked_doctype not in doctypes:
97  +                 # Note: Do NOT "update" cache because it can lead to concurrency bugs.
98  +                 frappe.cache.delete_value(key)
99  +                 frappe.db.after_commit.add(lambda: frappe.cache.delete_value(key))
```

frappe/monitor.py +1 -1

```
    @@ -128,7 +128,7 @@ def flush():
128 128                  logs = frappe.cache.lrange(MONITOR_REDIS_KEY, 0, -1)
129 129                  if logs:
130 130                          logs = list(map(frappe.safe_decode, logs))
131     -                    with open(log_file(), "a", os.O_NONBLOCK) as f:
    131 +                    with open(log_file(), "a") as f:
132 132                          f.write("\n".join(logs))
133 133                          f.write("\n")
134 134                  # Remove fetched entries from cache
```

frappe/public/js/frappe/ui/field_group.js +1 -1

```
@@ -128,7 +128,7 @@ frappe.ui.FieldGroup = class FieldGroup extends frappe.ui.form.Layout
{
128 128
129 129                   if (invalid.length && check_invalid) {
130 130                           frappe.msgprint({
131     -                                 title: __("Inavlid Values"),
    131 +                                 title: __("Invalid Values"),
132 132                                   message:
133 133                                           __("Following fields have invalid values:") +
134 134                                           "<br><br><ul><li>" +
```

frappe/tests/test_document.py +1 -1

```
@@ -533,7 +533,7 @@ def test_web_view_link_authentication(self):
533 533
534 534                   # without key
535 535                   url_without_key = f"/ToDo/{todo.name}"
536     -                 self.assertEqual(self.get(url_without_key).status, "403 FORBIDDEN")
    536 +                 self.assertEqual(self.get(url_without_key).status, "404 NOT FOUND")
537 537
538 538                   # Logged-in user can access the page without key
539 539                   self.assertEqual(self.get(url_without_key, "Administrator").status, "200
OK")
```

frappe/utils/background_jobs.py +4

```
@@ -231,6 +231,7 @@ def execute_job(site, method, event, job_name, kwargs, user=None,
is_async=True,
231 231                           # 1213 = deadlock
232 232                           # 1205 = lock wait timeout
233 233                           # or RetryBackgroundJobError is explicitly raised
    234 +                         frappe.job.after_job.reset()
234 235                           frappe.destroy()
235 236                           time.sleep(retry + 1)
236 237
@@ -252,6 +253,9 @@ def execute_job(site, method, event, job_name, kwargs, user=None,
is_async=True,
252 253                   return retval
253 254
254 255           finally:
    256 +                 if not hasattr(frappe.local, "site"):
    257 +                         frappe.init(site)
    258 +                         frappe.connect()
```

```
255  259                    for after_job_task in frappe.get_hooks("after_job"):
256  260                            frappe.call(after_job_task, method=method_name, kwargs=kwargs,
     result=retval)
257  261                    frappe.local.job.after_job.run()
```

frappe/utils/data.py +5

```
     @@ -1912,6 +1912,11 @@ def _raise_exception():
1912 1912                            elif any(keyword in column_name.split() for keyword in
      blacklisted_keywords):
1913 1913                                    _raise_exception()
1914 1914
      1915 +                           elif any(
      1916 +                                   re.search(rf"\b{keyword}\b", column_name,
      re.IGNORECASE) for keyword in blacklisted_keywords
      1917 +                           ):
      1918 +                                   _raise_exception()
      1919 +
1915 1920            elif regex.match(column_name):
1916 1921                    _raise_exception()
1917 1922
```

frappe/utils/print_format.py +3 -1

```
     @@ -226,7 +226,9 @@ def read_multi_pdf(output: PdfWriter) -> bytes:
226  226
227  227
228  228    @frappe.whitelist(allow_guest=True)
229       - def download_pdf(doctype, name, format=None, doc=None, no_letterhead=0, language=None,
      letterhead=None):
      229  + def download_pdf(
      230  +         doctype: str, name: str, format=None, doc=None, no_letterhead=0, language=None,
      letterhead=None
      231  + ):
230  232            doc = doc or frappe.get_doc(doctype, name)
231  233            validate_print_permission(doc)
232  234
```

frappe/website/utils.py +2 -2

```
     @@ -545,14 +545,14 @@ def build_response(path, data, http_status_code, headers: dict |
     None = None):
```

```
545 545             response = Response()
546 546             response.data = set_content_type(response, data, path)
547 547             response.status_code = http_status_code
548     -           response.headers["X-Page-Name"] = cstr(path.encode("ascii",
                    errors="xmlcharrefreplace"))
    548 +           response.headers["X-Page-Name"] = cstr(cstr(path).encode("ascii",
                    errors="xmlcharrefreplace"))
549 549             response.headers["X-From-Cache"] = frappe.local.response.from_cache or False
550 550
551 551             add_preload_for_bundled_assets(response)
552 552
553 553             if headers:
554 554                     for key, val in headers.items():
555     -                           response.headers[key] = cstr(val.encode("ascii",
                    errors="xmlcharrefreplace"))
    555 +                           response.headers[key] = cstr(cstr(val).encode("ascii",
                    errors="xmlcharrefreplace"))
556 556
557 557             return response
558 558
```

frappe/www/printview.py +8 -5

```
    @@ -350,15 +350,18 @@ def get_rendered_raw_commands(doc: str, name: str | None = None,
    print_format: s
350 350
351 351
352 352   def validate_print_permission(doc):
    353 +           if frappe.has_website_permission(doc):
    354 +                   return
    355 +
353 356             for ptype in ("read", "print"):
354     -                   if frappe.has_permission(doc.doctype, ptype, doc) or
                    frappe.has_website_permission(doc):
    357 +                   if frappe.has_permission(doc.doctype, ptype, doc):
355 358                             return
356 359
357     -           key = frappe.form_dict.key
358     -           if key and isinstance(key, str):
    360 +           if (key := frappe.form_dict.key) and isinstance(key, str):
359 361                     validate_key(key, doc)
360     -           else:
```

```
361       -                 raise frappe.PermissionError(_("You do not have permission to view this
            document"))
      362 +                 return
      363 +
      364 +         frappe.throw(_("{0} {1} not found").format(_(doc.doctype), doc.name),
            frappe.DoesNotExistError)
362   365
363   366
364   367     def validate_key(key, doc):
```

Comments 0

Please sign in to comment.