frappe / **frappe** Public

Code | Issues 2k | Pull requests 45 | Actions | Projects 2 | Wiki | Security

Commit 2ebd885

Browse files

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

```
Merge pull request #28898 from frappe/version-14-hotfix

chore: release v14
```

version-14 (#28898) · v14.95.0 ··· v14.89.0   2 parents faae98f + 6d89b81 commit 2ebd885

Filter files...

- frappe
  - build.py
  - desk/form
    - load.py
  - model
    - base_document.py
    - db_query.py
    - document.py
    - dynamic_links.py
  - monitor.py
  - tests
    - test_document.py
  - utils
    - background_jobs.py
    - data.py
    - print_format.py
  - website
    - utils.py
  - www
    - printview.py

13 files changed +74 -19 lines changed

Search within code

frappe/build.py +1 -1

```
        @@ -201,7 +201,7 @@ def symlink(target, link_name, overwrite=False):
201  201                  if os.path.isdir(link_name):
202  202                          raise IsADirectoryError(f"Cannot symlink over existing directory: '{link_name}'")
203  203                  try:
204       -                       os.replace(temp_link_name, link_name)
     204  +                       shutil.move(temp_link_name, link_name)
205  205                  except AttributeError:
206  206                          os.renames(temp_link_name, link_name)
207  207          except Exception:
```

frappe/desk/form/load.py +2

```
        @@ -14,6 +14,7 @@
 14   14   from frappe.model.utils.user_settings import get_user_settings
 15   15   from frappe.permissions import get_doc_permissions
 16   16   from frappe.utils.data import cstr
      17 + from frappe.utils.html_utils import clean_email_html
 17   18
 18   19
 19   20   @frappe.whitelist()
        @@ -262,6 +263,7 @@ def _get_communications(doctype, name, start=0, limit=20):
262  263          communications = get_communication_data(doctype, name, start, limit)
263  264          for c in communications:
264  265                  if c.communication_type in ("Communication", "Automated Message"):
     266  +                       clean_email_html(c.content)
265  267                          c.attachments = json.dumps(
266  268                                  frappe.get_all(
267  269                                          "File",
```

frappe/model/base_document.py +4 -2

```
          @@ -17,6 +17,7 @@
 17     17        table_fields,
 18     18    )
 19     19    from frappe.model.docstatus import DocStatus
        20  + from frappe.model.dynamic_links import invalidate_distinct_link_doctypes
 from frappe.model.naming import set_new_name
 from frappe.model.utils.link_count import notify_link_count
 from frappe.modules import load_doctype_module
@@ -748,6 +749,7 @@ def get_msg(df, docname):
                                         doctype = self.get(df.options)
                                         if not doctype:
                                                 frappe.throw(_("{0} must be set first").format(self.meta.get_label(df.options)))
+                                        invalidate_distinct_link_doctypes(df.parent, df.options, doctype)
 
                                 # MySQL is case insensitive. Preserve case of the original docname in the Link Field.
 
@@ -1234,11 +1236,11 @@ def get_value(self, fieldname):
         def cast(self, value, df):
                 return cast_fieldtype(df.fieldtype, value, show_warning=False)
 
-        def _extract_images_from_text_editor(self):
+        def _extract_images_from_editor(self):
                 from frappe.core.doctype.file.utils import extract_images_from_doc
 
                 if self.doctype != "DocType":
-                        for df in self.meta.get("fields", {"fieldtype": ("=", "Text Editor")}):
+                        for df in self.meta.get("fields", {"fieldtype": ("in", ("Text Editor", "HTML Editor"))}):
                                 extract_images_from_doc(self, df.fieldname)
 
 
```

frappe/model/db_query.py +2 -1

```
@@ -28,6 +28,7 @@
         get_time,
         get_timespan_date_range,
         make_filter_tuple,
+        sanitize_column,
 )
 from frappe.utils.data import DateTimeLikeObject, get_datetime, getdate, sbool
 
```

```diff
@@ -601,7 +602,7 @@ def build_filter_conditions(self, filters, conditions: list,
ignore_permissions=

 		for f in filters:
 			if isinstance(f, str):
-				conditions.append(f)
+				conditions.append(sanitize_column(f))
 			else:
 				conditions.append(self.prepare_filter_condition(f))

```

frappe/model/document.py +2 -2

```diff
@@ -548,7 +548,7 @@ def _validate(self):
 		self._fix_rating_value()
 		self._validate_code_fields()
 		self._sync_autoname_field()
-		self._extract_images_from_text_editor()
+		self._extract_images_from_editor()
 		self._sanitize_content()
 		self._save_passwords()
 		self.validate_workflow()
@@ -561,7 +561,7 @@ def _validate(self):
 			d._fix_rating_value()
 			d._validate_code_fields()
 			d._sync_autoname_field()
-			d._extract_images_from_text_editor()
+			d._extract_images_from_editor()
 			d._sanitize_content()
 			d._save_passwords()
 		if self.is_new():
```

frappe/model/dynamic_links.py +39 -3

```diff
@@ -42,9 +42,7 @@ def get_dynamic_link_map(for_delete=False):
 				dynamic_link_map.setdefault(meta.name, []).append(df)
 			else:
 				try:
-					links = frappe.db.sql_list(
-						"""select distinct `{options}` from
`tab{parent}`""".format(**df)
-					)
```

```diff
+                    links = fetch_distinct_link_doctypes(df.parent, df.options)
                     for doctype in links:
                         dynamic_link_map.setdefault(doctype, []).append(df)
                 except frappe.db.TableMissingError:
@@ -61,3 +59,41 @@ def get_dynamic_links():
     for query in dynamic_link_queries:
         df += frappe.db.sql(query, as_dict=True)
     return df
+
+
+def _dynamic_link_map_key(doctype, fieldname):
+    return f"dynamic_link_map::{doctype}::{fieldname}"
+
+
+def fetch_distinct_link_doctypes(doctype: str, fieldname: str):
+    """Return all unique doctypes a dynamic link is linking against.
+    Note:
+    - results are cached and can *possibly be outdated*
+    - cache gets updated when a document with different document link is discovered
+    - raw queries adding dynamic link won't update this cache
+    - cache miss can often be VERY expensive on large table.
+    """
+
+    key = _dynamic_link_map_key(doctype, fieldname)
+    doctypes = frappe.cache().get_value(key)
+
+    if doctypes is None:
+        doctypes = frappe.db.sql(f"""select distinct `{fieldname}` from `tab{doctype}`""", pluck=True)
+        frappe.cache().set_value(key, doctypes, expires_in_sec=12 * 60 * 60)
+
+    return doctypes
+
+
+def invalidate_distinct_link_doctypes(doctype: str, fieldname: str, linked_doctype: str):
+    """If new linked doctype is discovered for a dynamic link then cache is evicted."""
+
+    key = _dynamic_link_map_key(doctype, fieldname)
+    doctypes = frappe.cache().get_value(key)
+
+    if doctypes is None or not isinstance(doctypes, list):
```

```diff
+                return
+
+            if linked_doctype not in doctypes:
+                # Note: Do NOT "update" cache because it can lead to concurrency bugs.
+                frappe.cache().delete_value(key)
+                frappe.db.add_before_commit(lambda: frappe.cache().delete_value(key))
```

frappe/monitor.py +1 -1

```diff
@@ -125,7 +125,7 @@ def flush():
         logs = frappe.cache().lrange(MONITOR_REDIS_KEY, 0, -1)
         if logs:
             logs = list(map(frappe.safe_decode, logs))
-            with open(log_file(), "a", os.O_NONBLOCK) as f:
+            with open(log_file(), "a") as f:
                 f.write("\n".join(logs))
                 f.write("\n")
             # Remove fetched entries from cache
```

frappe/tests/test_document.py +1 -1

```diff
@@ -498,7 +498,7 @@ def test_web_view_link_authentication(self):
 
         # without key
         url_without_key = f"/ToDo/{todo.name}"
-        self.assertEqual(self.get(url_without_key).status, "403 FORBIDDEN")
+        self.assertEqual(self.get(url_without_key).status, "404 NOT FOUND")
 
         # Logged-in user can access the page without key
         self.assertEqual(self.get(url_without_key, "Administrator").status, "200 OK")
```

frappe/utils/background_jobs.py +4

```diff
@@ -198,6 +198,7 @@ def execute_job(site, method, event, job_name, kwargs, user=None, is_async=True,
             # 1213 = deadlock
             # 1205 = lock wait timeout
             # or RetryBackgroundJobError is explicitly raised
+            frappe.job.after_job.reset()
             frappe.destroy()
             time.sleep(retry + 1)
 
```

```
         @@ -219,6 +220,9 @@ def execute_job(site, method, event, job_name, kwargs, user=None, is_async=True,
219  220                  return retval
220  221
221  222          finally:
     223  +               if not hasattr(frappe.local, "site"):
     224  +                       frappe.init(site)
     225  +                       frappe.connect()
222  226                  for after_job_task in frappe.get_hooks("after_job"):
223  227                          frappe.call(after_job_task, method=method_name, kwargs=kwargs, result=retval)
224  228
```

frappe/utils/data.py +5

```
           @@ -1921,6 +1921,11 @@ def _raise_exception():
1921  1921                          elif any(keyword in column_name.split() for keyword in blacklisted_keywords):
1922  1922                                  _raise_exception()
1923  1923
      1924  +                       elif any(
      1925  +                               re.search(rf"\b{keyword}\b", column_name, re.IGNORECASE) for keyword in blacklisted_keywords
      1926  +                       ):
      1927  +                               _raise_exception()
      1928  +
1924  1929          elif regex.match(column_name):
1925  1930                  _raise_exception()
1926  1931
```

frappe/utils/print_format.py +3 -1

```
         @@ -122,7 +122,9 @@ def read_multi_pdf(output: PdfWriter) -> bytes:
122  122
123  123
124  124     @frappe.whitelist(allow_guest=True)
125       -  def download_pdf(doctype, name, format=None, doc=None, no_letterhead=0, language=None, letterhead=None):
     125  +  def download_pdf(
     126  +          doctype: str, name: str, format=None, doc=None, no_letterhead=0, language=None, letterhead=None
     127  +  ):
126  128             doc = doc or frappe.get_doc(doctype, name)
```

```
		validate_print_permission(doc)

```

frappe/website/utils.py +2 -2 🟩🟩🟥🟥⬜

```diff
@@ -531,14 +531,14 @@ def build_response(path, data, http_status_code, headers: dict | None = None):
 	response = Response()
 	response.data = set_content_type(response, data, path)
 	response.status_code = http_status_code
-	response.headers["X-Page-Name"] = cstr(path.encode("ascii", errors="xmlcharrefreplace"))
+	response.headers["X-Page-Name"] = cstr(cstr(path).encode("ascii", errors="xmlcharrefreplace"))
 	response.headers["X-From-Cache"] = frappe.local.response.from_cache or False
 
 	add_preload_for_bundled_assets(response)
 
 	if headers:
 		for key, val in headers.items():
-			response.headers[key] = cstr(val.encode("ascii", errors="xmlcharrefreplace"))
+			response.headers[key] = cstr(cstr(val).encode("ascii", errors="xmlcharrefreplace"))
 
 	return response
 
```

frappe/www/printview.py +8 -5 🟩🟩🟩🟥⬜

```diff
@@ -362,15 +362,18 @@ def get_rendered_raw_commands(doc, name=None, print_format=None, meta=None, lang
 
 
 def validate_print_permission(doc):
+	if frappe.has_website_permission(doc):
+		return
+
 	for ptype in ("read", "print"):
-		if frappe.has_permission(doc.doctype, ptype, doc) or frappe.has_website_permission(doc):
+		if frappe.has_permission(doc.doctype, ptype, doc):
 			return
```

```diff
 
-        key = frappe.form_dict.key
-        if key and isinstance(key, str):
+        if (key := frappe.form_dict.key) and isinstance(key, str):
             validate_key(key, doc)
-        else:
-            raise frappe.PermissionError(_("You do not have permission to view this document"))
+            return
+
+        frappe.throw(_("{0} {1} not found").format(_(doc.doctype), doc.name), frappe.DoesNotExistError)
 
 
 def validate_key(key, doc):
```

Comments 0

Please sign in to comment.