Branch: master ▾ notion-py / notion / smoke_test.py / <> Jump to ▾ Find file Copy path

 jamalex Add title_plaintext property to get/set titles without Markdown a09afd4 on 2 Jan

3 contributors   

227 lines (189 sloc) 7.13 KB Raw Blame History

```
1  from datetime import datetime
2
3  from .client import *
4  from .block import *
5
6
7  def run_live_smoke_test(token_v2, parent_page_url_or_id):
8
9      client = NotionClient(token_v2=token_v2)
10
11     parent_page = client.get_block(parent_page_url_or_id)
12
13     page = parent_page.children.add_new(
14         PageBlock,
15         title="Smoke test at {}".format(datetime.now().strftime("%Y-%m-%d %H:%M:%S")),
16     )
17
18     print("Created base smoke test page at:", page.get_browseable_url())
19
20     col_list = page.children.add_new(ColumnListBlock)
21     col1 = col_list.children.add_new(ColumnBlock)
22     col2 = col_list.children.add_new(ColumnBlock)
23     col1kid = col1.children.add_new(
24         TextBlock, title="Some formatting: *italic*, **bold**, ***both***!"
25     )
26     assert col1kid.title.replace("_", "*") == "Some formatting: *italic*, **bold**, ***b
27     assert col1kid.title_plaintext == "Some formatting: italic, bold, both!"
```

```
    col2.children.add_new(TodoBlock, title="I should be unchecked")
    col2.children.add_new(TodoBlock, title="I should be checked", checked=True)

    page.children.add_new(HeaderBlock, title="The finest music:")
    video = page.children.add_new(VideoBlock, width=100)
    video.set_source_url("https://www.youtube.com/watch?v=oHg5SJYRHA0")

    assert video in page.children
    assert col_list in page.children
    assert video in page.children.filter(VideoBlock)
    assert col_list not in page.children.filter(VideoBlock)

    page.children.add_new(SubheaderBlock, title="A link back to where I came from:")
    alias = page.children.add_alias(parent_page)
    assert alias.is_alias
    assert not page.is_alias
    page.children.add_new(
        QuoteBlock,
        title="Clicking [here]({}) should take you to the same place...".format(
            page.parent.get_browseable_url()
        ),
    )

    # ensure __repr__ methods are not breaking
    repr(page)
    repr(page.children)
    for child in page.children:
        repr(child)

    page.children.add_new(
        SubheaderBlock, title="The order of the following should be alphabetical:"
    )

    B = page.children.add_new(BulletedListBlock, title="B")
    D = page.children.add_new(BulletedListBlock, title="D")
    C2 = page.children.add_new(BulletedListBlock, title="C2")
    C1 = page.children.add_new(BulletedListBlock, title="C1")
    C = page.children.add_new(BulletedListBlock, title="C")
    A = page.children.add_new(BulletedListBlock, title="A")

    D.move_to(C, "after")
    A.move_to(B, "before")
    C2.move_to(C)
    C1.move_to(C, "first-child")

    page.children.add_new(CalloutBlock, title="I am a callout", icon="👆")

    cvb = page.children.add_new(CollectionViewBlock)
```

```python
    cvb.collection = client.get_collection(
        client.create_record("collection", parent=cvb, schema=get_collection_schema())
    )
    cvb.title = "My data!"
    view = cvb.views.add_new(view_type="table")

    special_code = uuid.uuid4().hex[:8]

    row = cvb.collection.add_row()
    assert row.person == []
    row.name = "Just some data"
    row.title = "Can reference 'title' field too! " + special_code
    assert row.name == row.title
    row.check_yo_self = True
    row.estimated_value = None
    row.estimated_value = 42
    row.files = [
        "https://www.birdlife.org/sites/default/files/styles/1600/public/slide.jpg"
    ]
    row.person = client.current_user
    row.tags = None
    row.tags = []
    row.tags = ["A", "C"]
    row.where_to = "https://learningequality.org"
    row.category = "A"
    row.category = ""
    row.category = None
    row.category = "B"

    # Run a filtered/sorted query using the view's default parameters
    result = view.default_query().execute()
    assert row in result

    # query the collection directly
    assert row in cvb.collection.get_rows(search=special_code)
    assert row not in cvb.collection.get_rows(search="otherworldly penguins")

    # search the entire space
    assert row in client.search_blocks(search=special_code)
    assert row not in client.search_blocks(search="otherworldly penguins")

    # Run an "aggregation" query
    aggregate_params = [
        {"property": "estimated_value", "aggregation_type": "sum", "id": "total_value"}
    ]
    result = view.build_query(aggregate=aggregate_params).execute()
    assert result.get_aggregate("total_value") == 42

```

```
    # Run a "filtered" query
    filter_params = [
        {
            "property": "person",
            "comparator": "enum_does_not_contain",
            "value": client.current_user.id,
        }
    ]
    result = view.build_query(filter=filter_params).execute()
    assert row not in result

    # Run a "sorted" query
    sort_params = [{"direction": "descending", "property": "estimated_value"}]
    result = view.build_query(sort=sort_params).execute()
    assert row in result

    print(
        "Check it out and make sure it looks good, then press any key here to delete it.
    )
    input()

    _delete_page_fully(page)


def _delete_page_fully(page):

    id = page.id

    parent_page = page.parent

    assert page.get("alive") == True
    assert page in parent_page.children
    page.remove()
    assert page.get("alive") == False
    assert page not in parent_page.children

    assert (
        page.space_info
    ), "Page {} was fully deleted prematurely, as we can't get space info about it anymo
        id
    )

    page.remove(permanently=True)

    time.sleep(1)

    assert (
        not page.space_info
```

```
    ), "Page {} was not really fully deleted, as we can still get space info about it".f
        id
    )


def get_collection_schema():
    return {
        "%9:q": {"name": "Check Yo'self", "type": "checkbox"},
        "=d{|": {
            "name": "Tags",
            "type": "multi_select",
            "options": [
                {
                    "color": "orange",
                    "id": "79560dab-c776-43d1-9420-27f4011fcaec",
                    "value": "A",
                },
                {
                    "color": "default",
                    "id": "002c7016-ac57-413a-90a6-64afadfb0c44",
                    "value": "B",
                },
                {
                    "color": "blue",
                    "id": "77f431ab-aeb2-48c2-9e40-3a630fb86a5b",
                    "value": "C",
                },
            ],
        },
        "=d{q": {
            "name": "Category",
            "type": "select",
            "options": [
                {
                    "color": "orange",
                    "id": "59560dab-c776-43d1-9420-27f4011fcaec",
                    "value": "A",
                },
                {
                    "color": "default",
                    "id": "502c7016-ac57-413a-90a6-64afadfb0c44",
                    "value": "B",
                },
                {
                    "color": "blue",
                    "id": "57f431ab-aeb2-48c2-9e40-3a630fb86a5b",
                    "value": "C",
                },
```

```
            ],
        },
        "LL[(": {"name": "Person", "type": "person"},
        "4Jv$": {"name": "Estimated value", "type": "number"},
        "OBcJ": {"name": "Where to?", "type": "url"},
        "dV$q": {"name": "Files", "type": "file"},
        "title": {"name": "Name", "type": "title"},
    }
```