


Join GitHub today

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)

Dismiss



Returning Invalid syntax for type json on arrays #442

New issue

Closed jrf0110 opened this issue Sep 16, 2013 · 19 comments

jrf0110 commented Sep 16, 2013

When your column is type json and you try and insert a JSON array, you get the following error:

```
{ [error: invalid input syntax for type json]
  length: 154,
  name: 'error',
  severity: 'ERROR',
  code: '22P02',
  detail: 'Expected string or "]", but found "1".',
  hint: undefined,
  position: undefined,
  internalPosition: undefined,
  internalQuery: undefined,
  where: 'JSON data, line 1: {1...',
  file: 'json.c',
  line: '665',
  routine: 'report_parse_error' }
```

For example, this works in PSQL:

```
insert into my_table ( data ) values ( '[1,2,3]' );
```

But this returns an error in node-pg:

```
client.query('insert into my_table ( data ) values ( $1 )', [ [1,2,3] ], function( error ){
  /* ERROR */
});
```

Assignees

No one assigned


Labels

bug

Milestone

No milestone

11 participants



booo commented Sep 16, 2013 Collaborator

We convert javascript arrays into postgres arrays. If you stringify your input array it should work:

```
client.query('INSERT INTO my_table (data) VALUES ($1)', [ JSON.stringify([1,2,3]) ], handler
```

We don't now the column type in advance so we can't do a auto conversions of the array.

7

jrf0110 commented Sep 16, 2013

@booo Yeah, I was thinking that may be the case. Well, in that case, non-issue!

jrf0110 closed this Sep 16, 2013

tgriesser referenced this issue in [bookshelf/bookshelf](#) Oct 31, 2013

JSON Arrays #94 Closed

badave commented Oct 31, 2013 Contributor

@booo @brianc This behavior is a bug. Converting data from `{}` to `{}` should not be determined by pg.

The only solution is to use `JSON.stringify` before running the insert query? What if you're running an ORM? What if you are unaware of this limitation?

Just because pg doesn't know the column type in advance does not mean you should convert to the syntax of column type "array". It might be helpful, but it's also a pretty big assumption and makes it really hard to use an array in a json field, despite it being supported and normal behavior by the database.

The better behavior would be to leave converting data types to the client. Both possibilities would be easy. As is, it's really hard to get the expected behavior using what's available.

spollack commented Oct 31, 2013 Contributor

see also: #374

jrf0110 commented Oct 31, 2013

@badave Doesn't your proposed behavior require the same amount of work to insert JSON? You're gonna be stringifying the JSON anyway.

badave commented Oct 31, 2013 Contributor

Proposed behavior passing `{}` or `[]` into a column without the library being opinionated about converting it into a postgres array type. The larger problem isn't the `JSON.stringify`, it's the behavior of pg. As a matter of fact, the point should be that I shouldn't have to do a `JSON.stringify` because the library already handles it with JSON objects without stringifying.

This shouldn't be a mandatory conversion. It's that simple. The database error is "invalid input syntax for type json", an insufficient error message as a user to conclude its caused by the lib transforming the data. It's transforming valid JSON into invalid JSON.

However, if I were using an array type and inserting an array into postgres with the incorrect format, I would understand the invalid input syntax for type array because it isn't JSON format, it requires an array format. At that point I would look for a solution that converted my data into the correct form to insert into the pg.

And I could understand it better if the array column type was any good in postgres, but it's complete crap. I'd prefer to use a json array because it is more recognizable.

jrf0110 commented Oct 31, 2013

Yeah, I see what you're saying. It's a bit funky that node-pg assumes to stringify Objects but not Arrays. I think making it more consistent, or rather, no incoming magic (I like the magic coming out though!) would be for the best. It seems a *little* arbitrary, especially now that JSON has been added to the data-type mix, that JS arrays are assumed to be PG Arrays, when they could be type json or even hstore.

My preference would be to make no assumptions and leave it up to the library consumer. I'm not sure if that means node-pg should `JSON.stringify` Arrays and Objects by default, as that would be trading one magic for another. But it certainly does seem like a sane default.

tgriesser referenced this issue in [tgriesser/knex](#) Jun 12, 2014

json array becomes key of json object (and fails to parse) #315 Closed

rhyss-vdw referenced this issue in [bookshelf/bookshelf](#) Aug 8, 2015

Json Array #864 Closed

domosapien referenced this issue Sep 15, 2015

Returning Invalid syntax for type jsonb on arrays #857 Closed

KyleLilly referenced this issue in [waterlinejs/postgresql-adapter](#) Dec 16, 2015

Unable to pass JSON Array as parameter #5 Closed

eriknyk commented May 12, 2016 · edited

Hi, this issue should be reopened, since in postgres shell is possible to execute:

```
INSERT INTO public.my_table ("userId", "personId", books) VALUES (1, 1, '[1,2,3]');
```

Best Regards.

eriknyk referenced this issue May 12, 2016

Error while trying to insert an json array #1016 Closed

JamesDixon commented Jul 25, 2016

I'm still running into this error even after stringifying the input array. Here's my JSON object:

```
{
  serviceId: 2, petId: 5, checked: true,
  serviceId: 3, petId: null, checked: false
}
```

After stringifying:

```
"[{\"serviceId\":2,\"petId\":5,\"checked\":true},{\"serviceId\":3,\"petId\":null,\"checked\":false}]"
```

I insert and receive the following error:

```
{\"msec\":54.03832702207565,\"error\":\"update \\\"appointment\\\" set \\\"ended_at\\\" = $1, \\\"report_card\\\" = $2, \\\"staff_notes\\\" = $3, \\\"status\\\" = $4, \\\"updated_at\\\" = $5 where \\\"id\\\" = $6 - invalid input syntax for type json\",\"data\":{\"message\":\"update \\\"appointment\\\" set \\\"ended_at\\\" = $1, \\\"report_card\\\" = $2, \\\"staff_notes\\\" = $3, \\\"status\\\" = $4, \\\"updated_at\\\" = $5 where \\\"id\\\" = $6 - invalid input syntax for type json\",\"severity\":\"ERROR\",\"code\":\"22P02\",\"condition\":\"invalid_text_representation\",\"detail\":\":\\\" but found \\\".\\\"\", \"where\":\"JSON data, line 1: {\\\" \\\"\\\\\\\"serviceId\\\\\\\":2,\\\\\\\"petId\\\\\\\":5,\\\\\\\"checked\\\\\\\":true\\\\\\\",...\", \"file\":\"json.c\", \"line\":11 }\", \"statusCode\":500, \"payload\":{\"statusCode\":500, \"error\":\"Internal Server Error\", \"message\":\"An internal server error occurred\"}, \"headers\":{}}}
```

It appears the input is being stringified again. Am I missing something here?

Appreciate any help!

brianc commented Jul 25, 2016 Owner

Here's an example of using a json data type & doing a round-trip query from node -> postgres -> and back to node with the json type preserved:

<https://github.com/brianc/node-postgres/blob/master/test/integration/client/json-type-parsing-tests.js>

Hope that helps!

JamesDixon commented Jul 25, 2016

@brianc thanks for this. However, isn't this just testing on a single object and not an array of objects?

jrf0110 commented Jul 25, 2016

@JamesDixon using pg for arrays of objects is working fine for me:

```
require('pg').connect( (error, client, done) => {
  if (error) throw error

  client.query('select $1::json as arr', [JSON.stringify([{'foo': 'bar'}])], (error, result)
    done()
    if (error) throw error
    console.log(result.rows[0].arr) // => [{"foo": "bar"}]
  })
})
```

jrf0110 commented Jul 25, 2016

If you stringify first, you'll need to cast the value to `json` or `jsonb` depending on what you're using

JamesDixon commented Jul 25, 2016

@jrf0110 thanks for this. I've confirmed this does work on insert as well. It appears something is double-stringifying my input.

Appreciate the help, fellas!

charmander reopened this Feb 8, 2017

Error: invalid input syntax for type json #1143 Closed

charmander added the **bug** label Feb 8, 2017

PythonDevOp commented Feb 20, 2017

@charmander so are issue 442 and 1143 overlapping issues? I see they are going back between open/close. @joskuijpers seemed to find my issue, I was wondering if any progress had been made regarding a solution

charmander commented Feb 20, 2017 Collaborator

@PythonDevOp Yes, they're the same issue. I'm not sure how @brianc wants to solve it, if at all, but you can use `JSON.stringify` manually in the meantime as seen above.

brianc commented Jun 16, 2017 Owner

If you need to pass in an array of objects as a single JSON value then do what @jrf0110 has done and call `JSON.stringify` on the parameter yourself before adding it to your array of parameters. node-postgres ignores any array parameter that's a string, passing it directly onto the wire and to the backend. You might need to cast in your query.

brianc closed this Jun 16, 2017

tamiyn commented Jun 29, 2017

Would it make sense for node-postgres to intercept this error and augment it with something like "Are you inserting an array into a JSON column? See <link to docs>? Helpful error messages make for a nice developer experience."

1

sparebytes commented Jun 30, 2017

A way to configure the default behavior would be desirable.

charmander marked this as a duplicate of #1383 Jul 21, 2017

Not able to insert array of json object into Postgres DB via query in Express #1383 Closed

charmander referenced this issue Aug 6, 2017

Insert or update with an empty Array creates Object in a JSON field #864 Closed

charmander referenced this issue Aug 29, 2017

Save an array value to jsonb type column #1432 Closed

charmander referenced this issue Nov 22, 2017

JSON arrays are being converted to PostgreSQL arrays and causing errors #1519 Closed

jgeurts added a commit to [mpink/bigal](#) that referenced this issue Feb 16, 2018

Released v1.0.6

2b4d8e4