

Postgres and JSON

⚡⚡⚡ lightning talk ⚡⚡⚡



**STORING
JSON IN COLUMNS**



**MODIFYING
JSON IN SQL**



**SQL SELECT
DATA INTO JSON**

Storing JSON

- every database can do this for ages
- JSON maps to text (except for JSONB)
- Postgres zips automatically 😎
- even MySQL has COMPRESS and UNCOMPRESS

Storing JSON - example

Postgres has a JSON type

```
1 CREATE TABLE table_with_json ( id SERIAL PRIMARY KEY NOT NULL, data JSON NOT NULL )
```

EZ

```
1 import 'node-postgres'  
2  
3 const data = {foo: 'bar'}  
4  
5 await client.query('INSERT INTO table_with_json (data) VALUES (1)', [data])
```

-> 🦊 simple-runtypes for type-safety of the json 🦊

Working with JSON in SQL

<https://www.postgresql.org/docs/12/functions-json.html>

weird overall syntax and SQL type system

this is wrong:

```
1 SELECT id FROM table WHERE data->'foo' = 'bar'
```

this compiles:

```
1 SELECT id FROM table WHERE data->>'foo' = 'bar'
```

Useful for migrations:

```
1  update assemblies
2  set body = jsonb_set(
3      jsonb_set_lax(
4          body::jsonb,
5          '{telescoping, innerTubeCenter}',
6          null,
7          true,
8          'delete_key'
9      ),
10     '{telescoping, innerTubeTop}',
11     '200'::jsonb
12 )
13 where body #> '{telescoping, innerTubeCenter}' is not null;
```

Functions I have used so far:

- `jsonb_pretty` - pretty print json
- `jsonb_path_query` - get object at path
- `jsonb_path_exists` - test for presence of path
- `jsonb_set_lax` - create a new, modified json object
- `#>` - path accessor function

Aggregate into JSON

😱😱 json_agg 😱😱

-> ran out of time 😞 and this was supposed to be a lightning talk anyway