

1. GRAPE REPORTS

Introduced in Grape 0.0.8

Grape provides a rudimentary infrastructure for writing and executing reports. Reports are written as stored procedures and stored in the table `grape.report`. A history of executed reports are stored in `grape.reports_executed`. When saving a report, a JSON field `input_fields` can be specified. The input parameters for the report should be specified in a JSON array. Each of the fields having the following values:

- `name` - Containing the field name
- `type` - Containing the field type (for example, date)

The report functions (stored procedures), is given one JSON parameter as input.

Report output are used as follows:

- If the reporting function's return type is JSON, the value is returned without any changes, to the API
- If the return type is a record (like a TABLE), the results will be put in an array and returned to the API call in a field called "result"
- Otherwise, the output will be automatically converted to JSON, and returned to the API call in a field called "result"

1.1 Examples

1.1.1 Example 1

Table: temps

CITY	DATE	TIME	TEMPERATURE
Pretoria	2016/01/01	06:00	20
Pretoria	2016/01/01	12:00	31
Pretoria	2016/01/02	06:00	20
Pretoria	2016/01/02	12:00	33
Cape Town	2016/01/01	06:00	17
Cape Town	2016/01/01	12:00	34
Cape Town	2016/01/02	06:00	19
Cape Town	2016/01/02	12:00	25

```
CREATE OR REPLACE FUNCTION maximum_city_for_date (JSON) RETURNS TEXT AS $$
DECLARE
    _date DATE;
    _city TEXT;
BEGIN
    _date := ($1->>'date')::DATE;
```

```
SELECT city INTO _city FROM temps WHERE date = _date ORDER BY temperature DESC LIMIT 1;

RETURN _city;
END; $$ LANGUAGE plpgsql;

SELECT grape.save_report (NULL, 'City with highest temperature for a date',
                        'public', 'maximum_city_for_date',
                        ' [{"name": "date", "type": "date"} ] '::JSON);
```

1.2 API Calls

1.3 Future ideas

- Add the ability to write reports that does not return or accept JSON
- Save report results