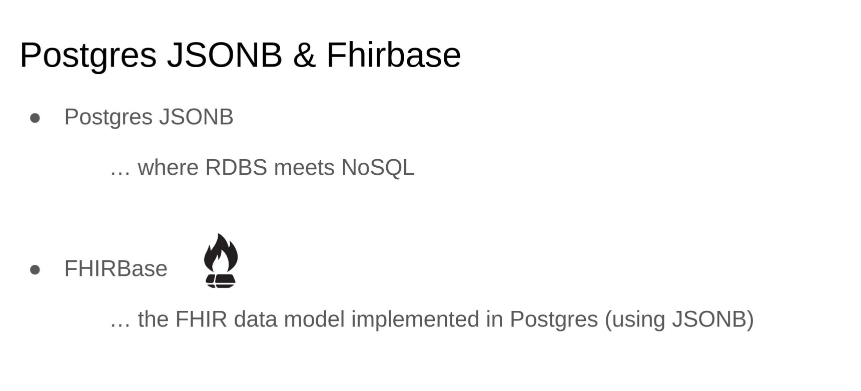
openIMIS

JSON Extensibility

openIMIS - JSON Extensibility

- Slide from Bonn 02/2019 workshop :



Postgres JSONB & Fhirbase

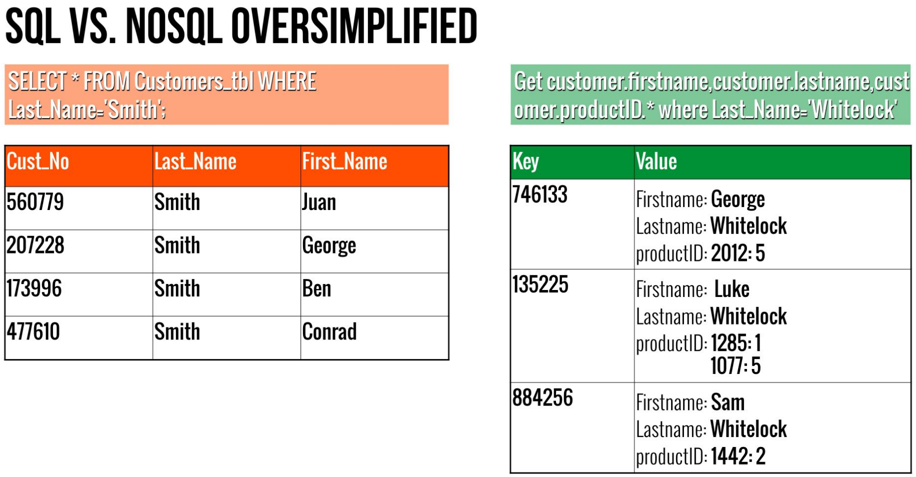
- Postgres JSONB

  … where RDBS meets NoSQL

- FHIRBase

  … the FHIR data model implemented in Postgres (using JSONB)

openIMIS - JSON Extensibility



SQL VS. NOSQL OVERSIMPLIFIED

```
SELECT * FROM Customers_tbl WHERE
Last_Name='Smith';
```

Cust_No	Last_Name	First_Name
560779	Smith	Juan
207228	Smith	George
173996	Smith	Ben
477610	Smith	Conrad

```
Get customer.firstname,customer.lastname,cust
omer.productID.* where Last_Name='Whitelock'
```

Key	Value
746133	Firstname: George Lastname: Whitelock productID: 2012: 5
135225	Firstname: Luke Lastname: Whitelock productID: 1285: 1 1077: 5
884256	Firstname: Sam Lastname: Whitelock productID: 1442: 2

source: https://arstechnica.com/ ars

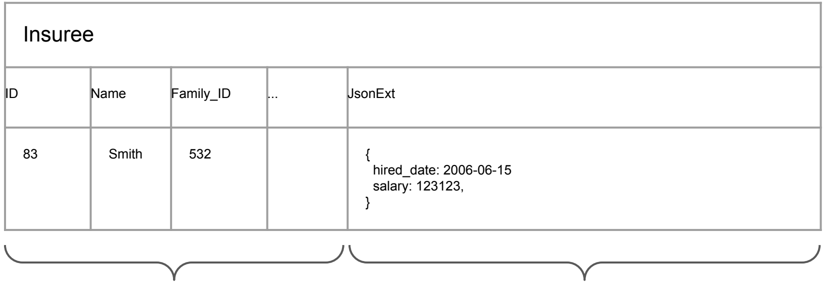
Highlights:

- Structure ‘enforced’ by the columns definitions
- Foreign keys… (delete restrict,...)

Highlights:

- Structure not ‘enforced’ by database (hence flexibility)
- No foreign key (you can refer to another ‘entity’ by its key, but constraints are not enforced by the model)

openIMIS - JSON Extensibility



Insuree

ID	Name	Family_ID	...	JsonExt

83	Smith	532		{
				hired_date: 2006-06-15
				salary: 123123,
				}

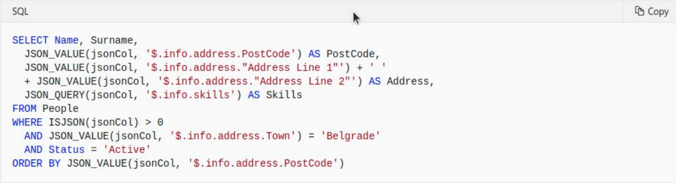
“Relational part”:

- legacy backward-compatible
- enforced FKs
- ...

“NoSQL part”:

- flexible
- ...

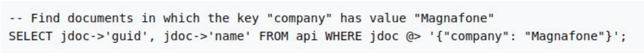
MS-SQL JSON support



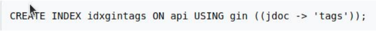
```sql
SELECT Name, Surname,
  JSON_VALUE(jsonCol, '$.info.address.PostCode') AS PostCode,
  JSON_VALUE(jsonCol, '$.info.address."Address Line 1"') + ' '
  + JSON_VALUE(jsonCol, '$.info.address."Address Line 2"') AS Address,
  JSON_QUERY(jsonCol, '$.info.skills') AS Skills
FROM People
WHERE ISJSON(jsonCol) > 0
  AND JSON_VALUE(jsonCol, '$.info.address.Town') = 'Belgrade'
  AND Status = 'Active'
ORDER BY JSON_VALUE(jsonCol, '$.info.address.PostCode')
```

… there are “supporting functions” (upon string) but it not a ‘type’

Postgres JSON(B) support



```sql
-- Find documents in which the key "company" has value "Magnafone"
SELECT jdoc->'guid', jdoc->'name' FROM api WHERE jdoc @> '{"company": "Magnafone"}';
```



```sql
CREATE INDEX idxgintags ON api USING gin ((jdoc -> 'tags'));
```

> ‘language’ not ‘normalized’ (not part of SQL standard)
> more ‘features’ in Postgres (hence its usage in tools like FHIRBase)

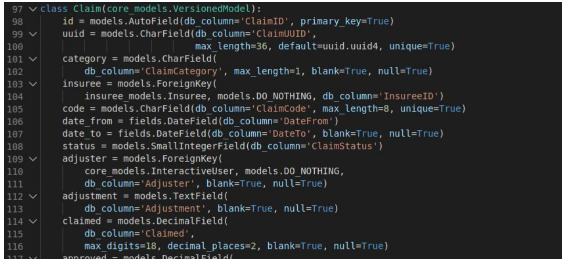
django and JSON fields

The `JSONField` is part of [`django.contrib.postgres.fields`]() module

… **but** there is a ['fallback' library]() (using string as base type)

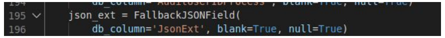
This allows us to have JSONField in our model..

mapped to varchar column in MS-SQL… and exposed as JSON field in GraphQL

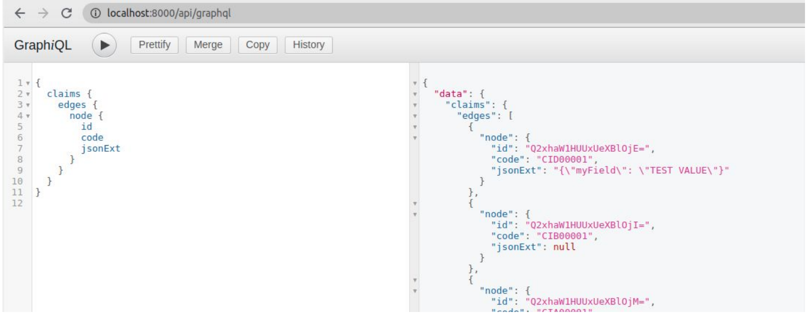


```python
class Claim(core_models.VersionedModel):
    id = models.AutoField(db_column='ClaimID', primary_key=True)
    uuid = models.CharField(db_column='ClaimUUID',
                            max_length=36, default=uuid.uuid4, unique=True)
    category = models.CharField(
        db_column='ClaimCategory', max_length=1, blank=True, null=True)
    insuree = models.ForeignKey(
        insuree_models.Insuree, models.DO_NOTHING, db_column='InsureeID')
    code = models.CharField(db_column='ClaimCode', max_length=8, unique=True)
    date_from = fields.DateField(db_column='DateFrom')
    date_to = fields.DateField(db_column='DateTo', blank=True, null=True)
    status = models.SmallIntegerField(db_column='ClaimStatus')
    adjuster = models.ForeignKey(
        core_models.InteractiveUser, models.DO_NOTHING,
        db_column='Adjuster', blank=True, null=True)
    adjustment = models.TextField(
        db_column='Adjustment', blank=True, null=True)
    claimed = models.DecimalField(
        db_column='Claimed',
        max_digits=18, decimal_places=2, blank=True, null=True)
    approved = models.DecimalField(
```

…



```python
        db_column='AuditUserIDProcess', blank=True, null=True)
    json_ext = FallbackJSONField(
        db_column='JsonExt', blank=True, null=True)
```

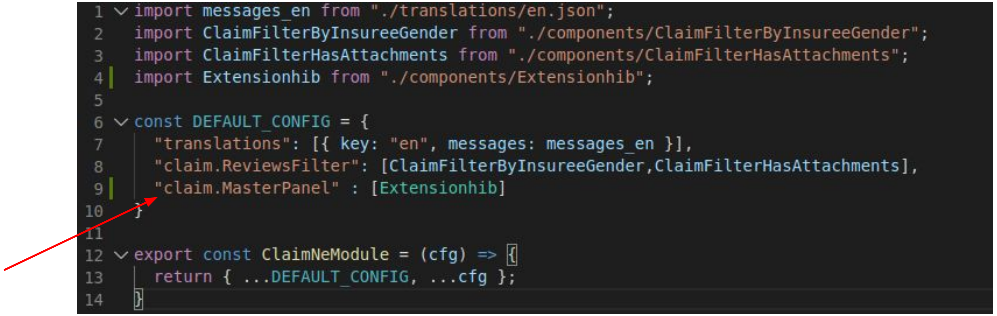


```graphql
{
  claims {
    edges {
      node {
        id
        code
        jsonExt
      }
    }
  }
}
```

```json
{
  "data": {
    "claims": {
      "edges": [
        {
          "node": {
            "id": "Q2xhaW1HUUxUeXBlOjE=",
            "code": "CID00001",
            "jsonExt": "{\"myField\": \"TEST VALUE\"}"
          }
        },
        {
          "node": {
            "id": "Q2xhaW1HUUxUeXBlOjI=",
            "code": "CIB00001",
            "jsonExt": null
          }
        },
        {
          "node": {
            "id": "Q2xhaW1HUUxUeXBlOjM=",
            "code": "CIA00001"
```

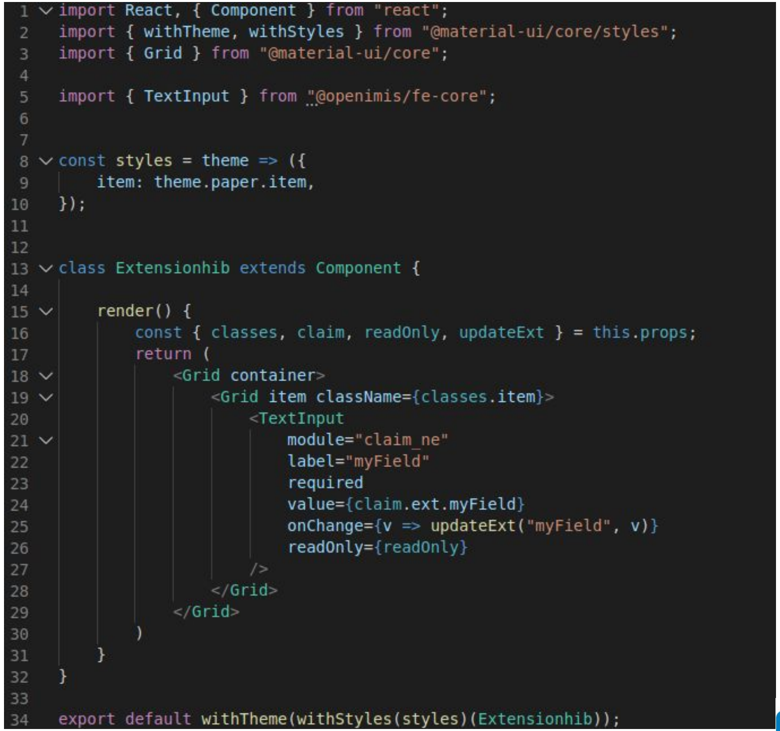
… used via frontend contribution

openimis-fe-**claim_ne**_js/src/index.js



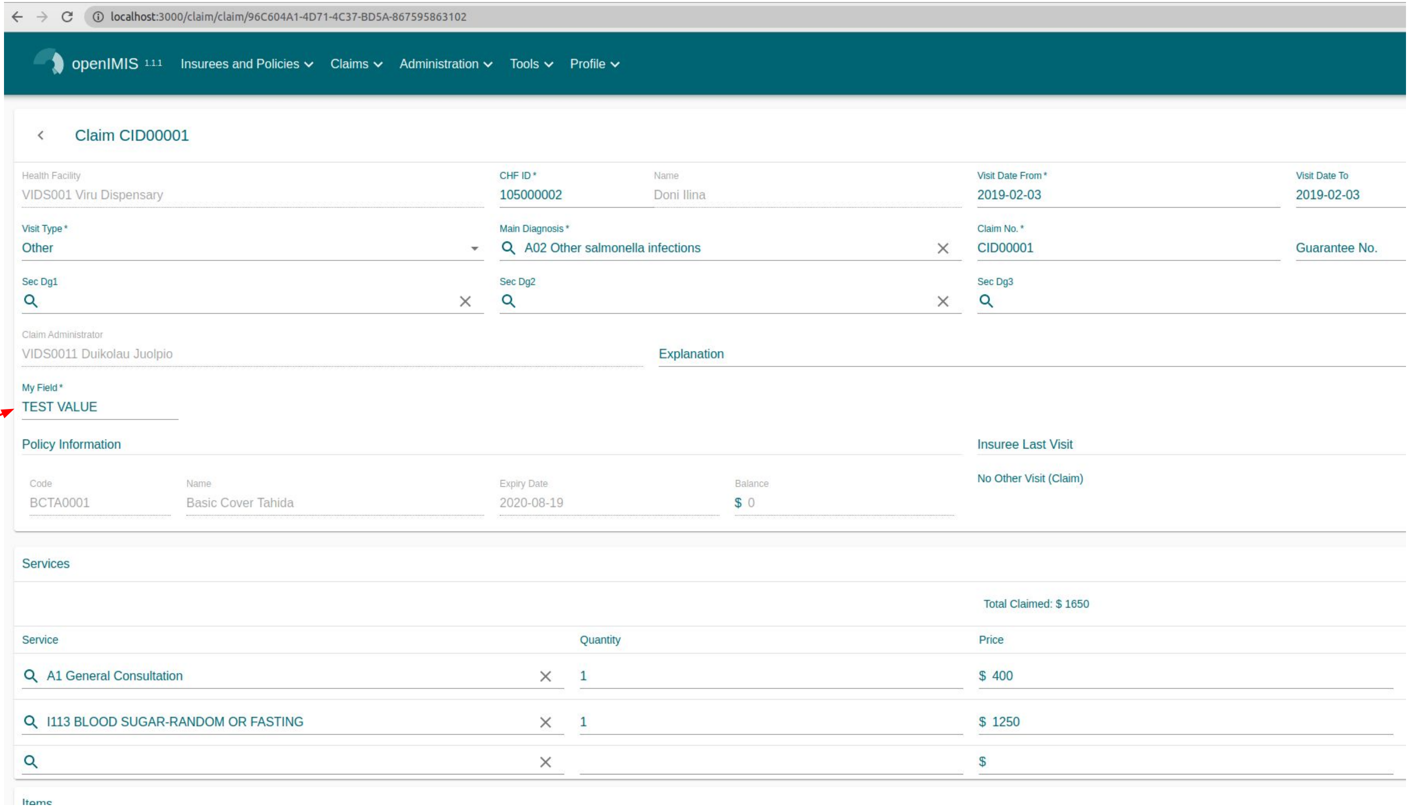
```
import messages_en from "./translations/en.json";
import ClaimFilterByInsureeGender from "./components/ClaimFilterByInsureeGender";
import ClaimFilterHasAttachments from "./components/ClaimFilterHasAttachments";
import Extensionhib from "./components/Extensionhib";

const DEFAULT_CONFIG = {
  "translations": [{ key: "en", messages: messages_en }],
  "claim.ReviewsFilter": [ClaimFilterByInsureeGender,ClaimFilterHasAttachments],
  "claim.MasterPanel" : [Extensionhib]
}

export const ClaimNeModule = (cfg) => {
  return { ...DEFAULT_CONFIG, ...cfg };
}
```

… with openimis-fe-**claim_ne**_js/src/components/Extensionhib.js



```
import React, { Component } from "react";
import { withTheme, withStyles } from "@material-ui/core/styles";
import { Grid } from "@material-ui/core";

import { TextInput } from "@openimis/fe-core";


const styles = theme => ({
    item: theme.paper.item,
});


class Extensionhib extends Component {

    render() {
        const { classes, claim, readOnly, updateExt } = this.props;
        return (
            <Grid container>
                <Grid item className={classes.item}>
                    <TextInput
                        module="claim_ne"
                        label="myField"
                        required
                        value={claim.ext.myField}
                        onChange={v => updateExt("myField", v)}
                        readOnly={readOnly}
                    />
                </Grid>
            </Grid>
        )
    }
}

export default withTheme(withStyles(styles)(Extensionhib));
```

… which gives in UI



Claim CID00001

Health Facility: VIDS001 Viru Dispensary

CHF ID *: 105000002

Name: Doni Ilina

Visit Date From *: 2019-02-03

Visit Date To: 2019-02-03

Visit Type *: Other

Main Diagnosis *: A02 Other salmonella infections

Claim No. *: CID00001

Guarantee No.

Sec Dg1

Sec Dg2

Sec Dg3

Claim Administrator: VIDS0011 Duikolau Juolpio

Explanation

My Field *: TEST VALUE

Policy Information

Code	Name	Expiry Date	Balance
BCTA0001	Basic Cover Tahida	2020-08-19	$ 0

Insuree Last Visit

No Other Visit (Claim)

Services

Total Claimed: $ 1650

Service	Quantity	Price
A1 General Consultation	1	$ 400
I113 BLOOD SUGAR-RANDOM OR FASTING	1	$ 1250
		$

Items