User-Interface ideas:

- User has floating design window where he has dropdowns for 'container', 'table' and a multiselect for 'fields in table'

- Eg he chooses individualForm, item and ticks name, width, expression (id is always ticked)

- When he clicks in UI on an item, following Dropin Script is generated in floating window:

update item (name, xxx, width: 100px, expression: yyy) WHERE id = 123;

SYNTAX:

Eg.

INSERT INTO [ database\_name . [ schema\_name ] . | schema\_name . ] table\_name

[ ( column\_name [ ,...n ] ) ]

{

VALUES ( { NULL | expression } )

| SELECT <select\_criteria>

}

[ OPTION ( <query\_option> [ ,...n ] ) ]

[;]

SQL:

define sql xxx

- SELECT a, b, c FROM tb WHERE ...

**CREATE BASE CONTAINER**

create base *container\_name*

template: *tmpl\_container\_name*

[menu: <size> [, options (*option\_caption* [, ...n])] eg xl, options (Main, Reports)

[logo: *file\_path*] eg /files/xxx/logo.png

[tabs: *caption* [, …n]

<size> := xl | md | … angularjs options

**CREATE CONTAINER**

create <dropin> *name*

[<container\_field\_name> : *value* [\r\n …n]]

[items: { <component> : (<item\_field\_name>: *value* [, …n]) [{<child\_records>}] [\r\n …n]}

[ports: {(<port\_field\_name>: value [, …n]) [\r\n …n]}

EG:

items: {

row: (name: xxx, flex: flex) {

col: (name: xxx, flex: flex) {

inputtext: (name\_field: first\_name, w: 10, …)

inputtext: (name\_field: last\_name, …) {

**events: {**

**'change' : ()**

}

}

}

}

}

ports: {

}

events: {

}

**UPDATE**

update <table> (<item\_field\_name>: *value* [, …n]) [where <criteria>]