⇐Back to the list of samples

Struct: Medium complex sample

This small guide is intended to facilitate the work with struct and to provide a small insight into its possibilities. We create multiple schemas and assign namespaces to them. We use lookups and create new entries with Bureaucracy. In the end, we will build a small database of systems, their suppliers, and the ability to track maintenance. It is a step by step guide.

Prerequisites:

- installed Sqlite Plugin (needed for Struct Plugin)
- installed Struct Plugin
- installed Bureaucracy Plugin

Step 1 - Initial thoughts

What do we want to achieve? We would like to create pages on which our equipment will be presented. There the most important data of the system should be clearly presented. In addition, contact data for problem cases should be visible. In addition, we would like to document the maintenance of the systems. Everything should look similar for each system and be realizable with little effort.

What do we need? Struct is a database, to keep the effort low later we can keep data relational. In databases tables are used, this is not different with struct. They are called schema and hold this data. That's why it is also important to think about the structure of the data among each other before starting.

In the assumed case, reused data (master data) should be the following:

- System types (one schema)
- Suppliers (another schema)

In addition, we want to document the maintenance:

- Maintenance (one more schema)

Last but not least, we want to document our systems:

- System (the main schema)

Step 2 - Schema creation

We create the required schemas. We go to the admin page and follow the link «Struct Schema Editor».

Schema: systemtype

Under «Create new Schema» we enter the name of the new schema at «Schema Name:», in this case `systemtype`. Our scheme system type is the simplest scheme, we have only one text field.

- Sort: 10
- Field Name: type
- Multi-Input?: No
- Configuration: –> Label: System Type
- Type: Text
- Enabled: Yes

Fields (**Configuration - only not default is mentioned**): Select first the Type of the field.

Schema: systemtype					
Sort	**Field Name**	**Multi-Input?**	**Configuration**	**Type**	**Enabled**
10	type	false	label: System Type	Text	true

For the moment, thats all. Click «Save». We have now created our first schema, only one field, but created.

Schema: systemsupplier

We now create the `systemsupplier` schema according to the same principle:

Fields (**Configuration - only not default is mentioned**): Select first the Type of the field. Click «Save» after each Field, you get a new sort-number and a new row.

Schema: systemsupplier					
Sort	**Field Name**	**Multi-Input?**	**Configuration**	**Type**	**Enabled**
10	name	false	label: Name	Text	true
20	main contact	false	label: Contact	Text	true
30	email	false	label: Email	Mail	true
40	phone	false	label: Phone	Text	true

Schema: system

A more complex scheme is `system`. Here we have lookup fields that refer to the previously created schemas. Here are the steps for the first three fields:

1. type, Type: Lookup, label: Type, schema: systemtype, field: type
2. supplier, Type: Lookup, label: Supplier, schema: systemsupplier, field: name
3. construction year, Type: Date, label: Construction Year, pastonly: true
4. and so on

Fields (**Configuration - only not default is mentioned**): Select first the Type of the field. Click

«Save» after each Field, you get a new sort-number and a new row.

Schema: `system`					
Sort	**Field Name**	**Multi-Input?**	**Configuration**	**Type**	**Enabled**
10	type	false	label: Type, schema: systemtype, field: type	Lookup	true
20	supplier	false	label: Supplier, schema: systemsupplier, field: name	Lookup	true
30	construction year	false	label: Construction Year, pastonly: true	Date	true
40	machine number	false	label: Machine number	Text	true
50	internal number	false	label: Internal number	Text	true
60	image	false	label: Image, width: 100, height: emty	Media	true
70	description	false	label: Description	Long Text	true
80	related systems	true	label: Related systems, usetitles: true	Page	true

For fully working you need one field for maintenance:

- internal number, Type: Text, label: Internal Number

For all fields please download the json.

Schema: maintenance

The schema `maintenance` also contains a lookup, namely to the mentioned internal number.

Fields (**Configuration - only not default is mentioned**): Select first the Type of the field. Click «Save» after each Field, you get a new sort-number and a new row.

Schema: `maintenance`					
Sort	**Field Name**	**Multi-Input?**	**Configuration**	**Type**	**Enabled**
10	system	false	label: System, schema: system, field: internal number	Lookup	true
20	type	false	label: Type, values: «Maintenance, Repair, Other»	Dropdown	true
30	description	false	label: Description	LongText	true
40	comment	false	label: Comment	LongText	true
50	date	false	label: Date	Date	true
60	user	false	label: User, existingonly: false	User	true

Since creating a scheme is a lot of work and the principle should be clear now, here are the schemes for easy download.

Assign the main scheme

It is time to assign our main schema to the namespaces where it should be active. Go to the admin page and follow the link «Struct Schema Assignments».

On the page «Schema Assignments»:

- **Page/Namespace:** infrastructure:systems:**
- **Schema:** `system`

Click the «Add» Button.

For the namespace `infrastructure:systems` we have connected the schema `system` for all pages (first *) and all subnamespaces (second *) and their pages.

Step 3 - Simple Lookup

Now you need to enter data into the schemas. To do this, it is a good idea to create a page where you maintain the "master data". In our example these are the `systemsupplier` and the `systemtype`. We create a «Settings» page with the following content:

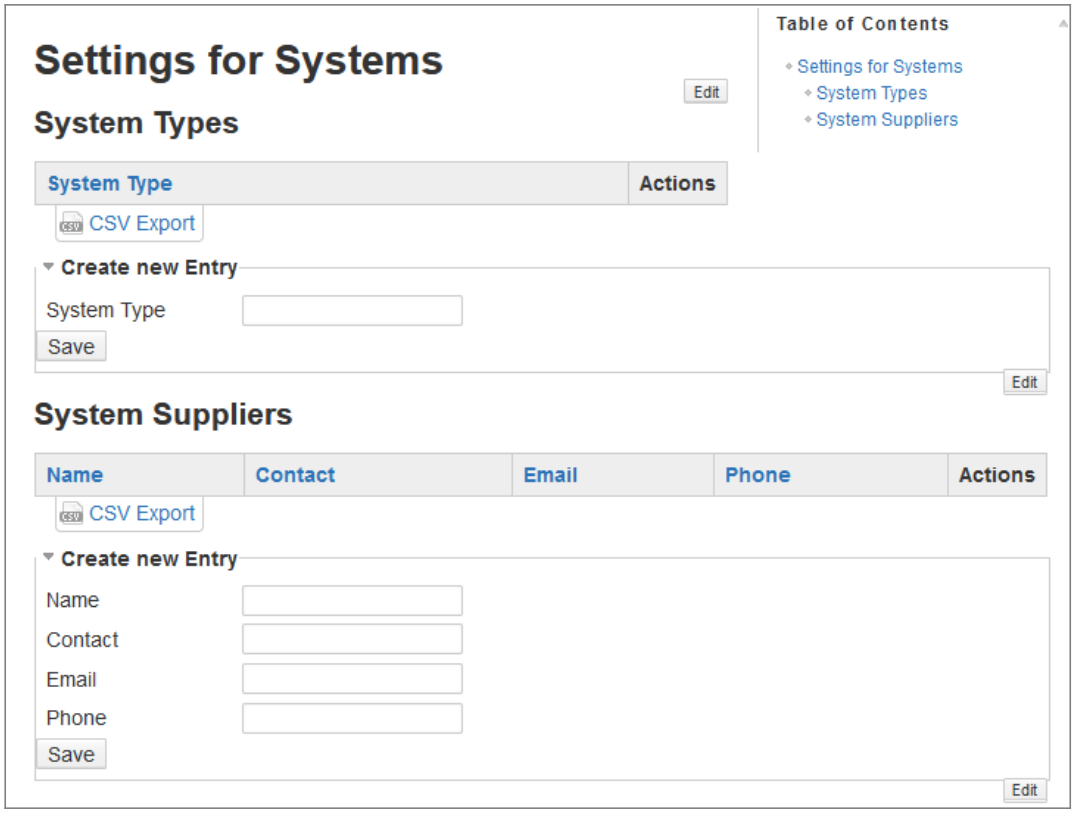
settings.txt

```
====== Settings for Systems ======

===== System Types =====
---- struct global ----
schema: systemtype
----


===== System Suppliers =====
---- struct global ----
schema: systemsupplier
----
```

The page gives us the possibility to fill the created schemas for the suppliers and system types.



We should now enter some data, whatever it is. It is enough for understanding, and nothing is for eternity.

Step 4 - Create a System

Our namespace for systems are `infrastructure:systems`. We create a page called like a system, e.g. `srv01`. If we have done everything correctly, an input mask will appear below the editor where we can store data. Actually, a headline and the filled-in data is enough to generate a fully-fledged page. However, we want to try further possibilities.

Fill the page with following:

srv01.txt

```
====== Server 01 ======

===== Supplier Contact: =====

---- struct list ----
schema: systemsupplier
cols: main contact, email, phone
header: "Name: ", "- Mail: "," - Phone: "
sepbyheaders: yes
filter: name = $STRUCT.system.supplier$
----

===== Maintenance =====

---- struct table----
cols: date, type, description, comment, user
schema: maintenance
filter: system = $STRUCT.system.internal number$
sort: ^date
csv: 0
----

===== Add Maintenance =====

<form>
action struct_lookup
struct_field "maintenance.date" =%Y-%m-%d
struct_field "maintenance.type"
struct_field "maintenance.description"
struct_field "maintenance.comment" !
struct_field "maintenance.user"
struct_fieldhidden "maintenance.system" "=[""@FORMPAGE_ID@"",0]"
```

```
submit "Add Enty"
thanks "Please reload the Page!"
</form>
```

Supplier Contact:

We use the struct List.

```
---- struct list ----
schema: systemsupplier
cols: main contact, email, phone
header: "Name: ", "- Mail: "," - Phone: "
sepbyheaders: yes
filter: name = $STRUCT.system.supplier$
----
```

- schema: our created `systemsupplier` scheme
- cols: in this case we use 3 fields, we are also able to use * for all columns (more headers required)
- header: we want descriptions by the data, with « you can create spaces
- sepbyheaders: necessary to display the header
- filter: We search for the name (`name`) of the system supplier using the data entered on this page (`$STRUCT.system.supplier$`).

The result is a one-liner:

- Name: John Doe - Mail: j.doe@cd-c.com - Phone: +12 345 6789

Maintenance

We use the struct Table.

```
---- struct table----
cols: date,type,description,comment,user
schema: maintenance
filter   : system = $STRUCT.system.internal number$
sort: ^date
csv: 0
----
```

- cols: we use only the required columns
- schema: our created `maintenance` scheme
- filter: only lines in the schema `maintenance` corresponding to the current `internal number` of the system
- sort: we want the newest on top
- csv: we don't want export option

As a result, we get a table with all the maintenance entered for the system.

Add Maintenance

We use now the Bureaucracy Integration.

```
===== Add Maintenance =====

<form>
action struct_lookup
struct_field "maintenance.date" =%Y-%m-%d
struct_field "maintenance.type"
struct_field "maintenance.description"
struct_field "maintenance.comment" !
struct_field "maintenance.user"
struct_fieldhidden "maintenance.system" "=[""@FORMPAGE_ID@"",0]"
submit "Add Enty"
thanks "Please reload the Page!"
</form>
```

- `action struct_lookup` we want use data
- `struct_field «maintenance.date» =%Y-%m-%d` prefill the date field from schema `maintenance` with current date
- `struct_field «maintenance.type»`, `struct_field «maintenance.description»` and `struct_field «maintenance.user»` input area for this **mandatory** fields
- `struct_field «maintenance.comment» !` a field with input area, not mandatory
- `struct_fieldhidden «maintenance.system» »=[«»@FORMPAGE_ID@«»,0]«` assign data from current page to the `maintenance`, in this case the `internal number` (`maintenance.system` is a lookup field to `system.internal number`, you remember Step 2?). This field is hidden, no input area!
- `submit «Add Enty»` creates the button to submit the data
- `thanks «Please reload the Page!»` a message to reload the page after submit

For completness: it is also possible to work simply with serial data. This way is described here.

Re-useable Code

Each page in the assigned namespace can contain the same text, just with a different title. Here a `_template.txt` for Namespace Templates offers itself.

What we get

Sample Schemas

Save the json local, create the schema (systemtype, systemsupplier, maintenance, system) and then

go to Import/Export and Import a Schema from JSON. Select the file you have downloaded for the selected Schema.

systemtype.struct.json


```
{
    "structversion": "2021-08-11",
    "schema": "systemtype",
    "id": "41",
    "user": "admin",
    "config": {
        "allowed editors": "",
        "label": {
            "en": ""
        }
    },
    "columns": [
        {
            "colref": 1,
            "ismulti": false,
            "isenabled": true,
            "sort": 10,
            "label": "type",
            "class": "Text",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "prefix": "",
                "postfix": "",
                "label": {
                    "en": "System Type"
                },
                "hint": {
                    "en": ""
                }
            }
        }
    ]
}
```


systemsupplier.struct.json


```
{
    "structversion": "2021-08-11",
    "schema": "systemsupplier",
    "id": "40",
```

```
    "user": "admin",
    "config": {
        "allowed editors": "",
        "label": {
            "en": "System Supplier"
        }
    },
    "columns": [
        {
            "colref": 1,
            "ismulti": false,
            "isenabled": true,
            "sort": 10,
            "label": "name",
            "class": "Text",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "prefix": "",
                "postfix": "",
                "label": {
                    "en": "Name"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 2,
            "ismulti": false,
            "isenabled": true,
            "sort": 20,
            "label": "main contact",
            "class": "Text",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "prefix": "",
                "postfix": "",
                "label": {
                    "en": "Contact"
                },
                "hint": {
                    "en": ""
```

```
                }
            }
        },
        {
            "colref": 3,
            "ismulti": false,
            "isenabled": true,
            "sort": 30,
            "label": "email",
            "class": "Mail",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "prefix": "",
                "postfix": "",
                "label": {
                    "en": "Email"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 4,
            "ismulti": false,
            "isenabled": true,
            "sort": 40,
            "label": "phone",
            "class": "Text",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "prefix": "",
                "postfix": "",
                "label": {
                    "en": "Phone"
                },
                "hint": {
                    "en": ""
                }
            }
        }
    ]
}
```

maintenance.struct.json


```
{
    "structversion": "2021-08-11",
    "schema": "maintenance",
    "id": "56",
    "user": "admin",
    "config": {
        "allowed editors": "",
        "label": {
            "en": "Maintenance"
        }
    },
    "columns": [
        {
            "colref": 1,
            "ismulti": false,
            "isenabled": true,
            "sort": 10,
            "label": "system",
            "class": "Lookup",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "schema": "system",
                "field": "internal number",
                "label": {
                    "en": "System"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 2,
            "ismulti": false,
            "isenabled": true,
            "sort": 20,
            "label": "type",
            "class": "Dropdown",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "values": "Maintenance, Repair, Other",
```

```
                "label": {
                    "en": "Type"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 3,
            "ismulti": false,
            "isenabled": true,
            "sort": 30,
            "label": "description",
            "class": "LongText",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "prefix": "",
                "postfix": "",
                "rows": "5",
                "cols": "50",
                "label": {
                    "en": "Description"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 4,
            "ismulti": false,
            "isenabled": true,
            "sort": 40,
            "label": "comment",
            "class": "LongText",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "prefix": "",
                "postfix": "",
                "rows": "5",
                "cols": "50",
                "label": {
                    "en": "Comment"
                },
```

```
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 5,
            "ismulti": false,
            "isenabled": true,
            "sort": 50,
            "label": "date",
            "class": "Date",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "format": "Y\/m\/d",
                "prefilltoday": false,
                "pastonly": false,
                "futureonly": false,
                "label": {
                    "en": "Date"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 6,
            "ismulti": false,
            "isenabled": true,
            "sort": 60,
            "label": "user",
            "class": "User",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "existingonly": false,
                "autocomplete": {
                    "fullname": true,
                    "mininput": 2,
                    "maxresult": 5
                },
                "label": {
                    "en": "User"
```

```
                },
                "hint": {
                    "en": ""
                }
            }
        }
    ]
}
```

system.struct.json

```
{
    "structversion": "2021-08-11",
    "schema": "system",
    "id": "50",
    "user": "admin",
    "config": {
        "allowed editors": "",
        "label": {
            "en": "System"
        }
    },
    "columns": [
        {
            "colref": 1,
            "ismulti": false,
            "isenabled": true,
            "sort": 10,
            "label": "type",
            "class": "Lookup",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "schema": "systemtype",
                "field": "type",
                "label": {
                    "en": "System Type"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 2,
            "ismulti": false,
            "isenabled": true,
            "sort": 20,
```

```
            "label": "supplier",
            "class": "Lookup",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "schema": "systemsupplier",
                "field": "name",
                "label": {
                    "en": "Supplier"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 3,
            "ismulti": false,
            "isenabled": true,
            "sort": 30,
            "label": "construction year",
            "class": "Date",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "format": "Y\/m\/d",
                "prefilltoday": false,
                "pastonly": true,
                "futureonly": false,
                "label": {
                    "en": "Construction Year"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 4,
            "ismulti": false,
            "isenabled": true,
            "sort": 40,
            "label": "machine number",
            "class": "Text",
            "config": {
```

```
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "prefix": "",
                "postfix": "",
                "label": {
                    "en": "Machine number"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 5,
            "ismulti": false,
            "isenabled": true,
            "sort": 50,
            "label": "internal number",
            "class": "Text",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "prefix": "",
                "postfix": "",
                "label": {
                    "en": "Internal number"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 6,
            "ismulti": false,
            "isenabled": true,
            "sort": 60,
            "label": "image",
            "class": "Media",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "mime": "image\/",
                "width": 100,
                "height": "",
```

```
                "agg_width": "",
                "agg_height": "",
                "label": {
                    "en": "Image"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 7,
            "ismulti": false,
            "isenabled": true,
            "sort": 70,
            "label": "description",
            "class": "LongText",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "prefix": "",
                "postfix": "",
                "rows": "5",
                "cols": "50",
                "label": {
                    "en": "Description"
                },
                "hint": {
                    "en": ""
                }
            }
        },
        {
            "colref": 8,
            "ismulti": true,
            "isenabled": true,
            "sort": 80,
            "label": "related systems",
            "class": "Page",
            "config": {
                "visibility": {
                    "inpage": true,
                    "ineditor": true
                },
                "usetitles": true,
                "autocomplete": {
                    "mininput": 2,
```

```
                    "maxresult": 5,
                    "namespace": "",
                    "postfix": ""
                },
                "label": {
                    "en": "Related systems"
                },
                "hint": {
                    "en": ""
                }
            }
        }
    ]
}
```