WebDatabase

Build a full web database application without a line of code

http://webdatabase.canalblog.com

Version 1.4 for Symfony 1.4 / 1.3

Bernard Maison

WHAT IS WEBDATABASE

It is quite easy to develop a small database application on your PC, using basic Office tools, but it is much more difficult to transform it to a Web application that you can share with your colleagues on the Intranet.

WebDatabase is the answer : build a complete Web database application, without writing a line of code, just by describing your data model and behavior in your favorite Spreadsheet . A full PHP web site will be automatically generated.

WebDatabase is a plugin above the Symfony framework. As you will see in this document, it is very quick to build a new application from scratch, even if you have no knowledge of the Symfony framework.

TABLE OF CONTENTS

1 Vocabulary

Here some definitions used in this document

TABLE
A table in the database. Generally corresponds to a Symfony "module"

FIELD
A column in a TABLE.
The field "bar" of table "foo" will be noted foo:bar

REFERENCE
Saying that "the field article:authorid is a reference to table foreign table author" , means that article:authorid is a foreign key that contains the id of the related author.

2 Prerequisites

WebDatabase requires that some rules are observed.
Some of those prerequisites may be not understandable until you have finished reading this document
Just remember that they exist .

Webdatabase 1.4 works only will version 1.4.x and 1.3.x of Symfony , and uses the Propel option

The delimiter for .CSV files is ;

Do not use underscores in Table of Field names.
On one side, Symfony often changes them into capital letters, on the other side, some internal values are prefixed by underscore .
There is an exception to this rule : fields like `created_at` that Symfony manages automatically.

All Tables will have an automatically incremented field "id" which is used as a primary key.
This field will be used when a table has a foreign key pointing to another table.
"id" is a reserved field name

A table cannot have more than 1 reference to the same foreign table
(that's due to Propel limitations, for which I could not find a workaround)

Do not use fields name similar to table names.
For example having a table Note and a field Notes, might cause conflicts in generated PHP function names

The input of WebDatabase is a file named wd.csv , with .CSV format , at the root of your project.
It is better to use a spreadsheet to generate wd.csv file
If you have no spreasheet tools, writing this .csv file will be quite hard.

WebDatabase generates a Symfony application called "backend"

3 A very brief introduction to Symfony

This chapter is intended for users who deal with Symfony for the first time. It helps understanding how it works.

Imagine there is a table XXX in your database. Then Symfony will respond to URL like

`http://localhost/mywebsite/index.php/xxx/list`	list XXX items
`http://localhost/mywebsite/index.php/xxx/edit/id/3`	edit XXX item with id=3

Here, list and edit are called "actions" that will produce HTML templates called "views" which are the way Symfony uses to show and modifiy data from table XXX
XXX will be called by Symfony as a "module"

So , a typical URL is like
`http://host/mywebsite/index.php/module/action`
`http://host/mywebsite/index.php/module/action/param1/param1value/param2/param2value`

When using URL `http://host/ywebsite/index.php/xxx/yyy` , the entry point index.php will do the following

- Read configuration files, and (first call only) build templates in the cache

- Interpret `xxx/yyy` to call executeYyy from actions.class.php in apps/backend/modules/xxx/actions and cache\backend\dev\modules\autoXxx\actions
    - executeYyy will use class Xxx and XxxPeer from lib/model to interact with the database and populate php object Xxx
      (those classes themselves derive from BaseXxx and BaseXxxPeer in lib/model/om)

- Finally produce the Web output, through the template yyySuccess.php , located in cache\backend\dev\modules\autoXxx\templates , and using values from the Xxx object

When using URL `http://host/mywebsite/index.php/xxx/edit/id/3` , it works a similar way .
id/3 will be analysed to produce a parameter "id" with value 3 for the web request.
Note that there is an alternative syntax to this: `http://host/mywebsite/index.php/xxx/3/edit`

4 Using WebDatabase

4.1 Prerequisites

Have an Apache + Mysql environment (you may use another database engine, provided that you read Symfony documentation to modify config/databases.yml configuration file)

Using EasyPHP , I got some troubles with php.ini , so if you encounter troubles , do the following changes in php.ini

- disable extension=domxml.dll (it's an old thing which causes troubles)
- enable extension=php_xsl.dll (otherwise troubles with XSLT parsing)

Download the file check_configuration.php from URL http://sf-to.org/1.4/check.php and execute it with php to check your php.ini configuration .
Be aware that Apache may use a different php.ini , so execute it also from your web server.

4.2 First use

4.2.1 Step1 : Install Symfony

Download the Symfony1.4 archive (.tgz or .zip) from `www.symfony-project.org/installation`
CAUTION : Save it on your disk, don't try to open the .zip directly from the web

Create a home directory for your project, and inside this home directory create lib/vendor .

All the subsequent commands will have to be typed from this home directory

Unpack the archive and move files to `lib/vendor/symfony` (so that file `lib/vendor/symfony/README` exists)

Fix a bug in `lib/vendor/symfony/lib/task/generator/sfGenerateProjectTask.class.php`
find and correct the following lines as follows :

```
sprintf('dirname(__FILE__).\'/../%s/autoload/sfCoreAutoload.class.php\'',
str_replace(sfConfig::get('sf_root_dir'), '', sfConfig::get('sf_symfony_lib_dir'))) :
var_export(sfConfig::get('sf_symfony_lib_dir').'/autoload/sfCoreAutoload.class.php',true);
```

Fix a bug in `lib/vendor/symfony/lib/validator/sfValidatorSchema.class.php`
Find out the following line in function doClean

```
// are non given values required?
```

And replace it by

```
// are non given values required?
$unused=array(); //@@WD
```

Initialize your project , using the Propel environment (Propel is mandatory)

```
php lib/vendor/symfony/data/bin/symfony generate:project PROJECT_NAME  --orm=Propel
```

Check that all is OK

```
php symfony -V
```

Initialise the symfony "backend" application (caution ! this name is mandatory)

```
php symfony generate:app backend
```

Get the right web asset files

```
php symfony plugin:publish-assets
```

then copy `lib/vendor/symfony/data/web/sf` to `web/sf` (or make an alias inside Apache)

4.2.2 Step2 : install Webdatabase

Download WebDatabasePlugin from `www.symfony-project.org/plugins`
CAUTION : Save it on your disk, don't try to open the .zip directly from the web

The plugin contains documentation and a file named "wd.zip" . Get this file and unpack it at the root of the project directory
This will overwrite the following files:

- config/databases.yml ,
- web/css/main.css ,
- apps/backend/templates/layout.php
- apps/backend/config/settings.yml

Caution ! Do not do this before executing the `generate:app` command , otherwise this command will fail

4.2.3 Step3 : create the application

Create an empty mysql database, then tell Symfony to use it
Do this **after** downloading Webdatabase, since this overwrites databases.yml
```
php symfony configure:database "mysql:host=HOSTNAME;dbname=DATABASE" USERNAME PASSWD
```

Create the wd.csv file describing your application's tables and fields
(or simply keep the example one provided by Webdatabase)

Generate the application , build the database, then load it with data
```
php wd makedb
```

This will first generate configuration files for Symfony and some Php code

`config/schema.yml`	Data model
`lib/model/WDXxx.php`	High level functions for each table xxx
`lib/filters/XxxFormFilter.class.php`	code for each table xxx
`lib/form/XxxForm.class.php`	code for each table xxx
`apps/backend/modules/xxx/actions/actions.class.php`	Actions for each table xxx
`apps/backend/modules/xxx/config/generator.yml`	Config file used by Symfony to generate templates for each table xxx
`lib/model/_choices.php`	Class for choice lists
`lib/model/_wd.php`	some common code
`data/fixtures/xxx.yml`	Data to be loaded in the database

Then it will generate the Symfony application files , build database , populate it with data

Modify Apache configuration so that it can respond to URL like `http://localhost/mysite/index.php`
```
Alias /mysite  project-home-directory/web
<Directory "project-home-directory/web">
     Allow from all
       AllowOverride All
</Directory>
```

That's all. You can now test the build-in example
`http://localhost/mysite/debug.php` (debug version)
`http://localhost/mysite/index.php` (production environment)

CAUTION : with Internet Explorer you might encounter a strange bug , where IE asks you whether to open or save the file to disk , rather than silently show the resulting HTML page . If this happen , use instead :
`http://localhost/mysite/index.php/default/index`

4.3 Subsequent uses

If you do not modify the data model (that is create/remove/modify fields in tables), you don't need to recreate the database . Just call :
```
php wd
```

Pay attention before using the "`makedb`" parameter : check that you really want to destroy and recreate your database !

If you want to recreate the database, but do not want to bother with data loading, then issue the command
```
php wd makedbsql
```

5 Format of wd.csv file

The first line contains columns titles. Webdatabase uses those titles to get information, so you must not modify them. You can yet freely insert columns at any place, provided that you do not used titles reserved names.

Blank lines are ignored

Imagine we have two tables ,
workorder:
  date
  content

worker:
  name

then the wd.csv file will have the following structure

TABLE	FIELD	....
workorder		
	date	
	content	
worker		
	name	

It is easy to comment out a FIELD: just begin his name by a * The line will then be ignored
Remember that each table will have a "id" field. But do not add it yourself : it will be automatically added.

The following columns deal with database model

TABLE	the table name
FIELD	the field name
TYPE	type for the database field Standard values are • BOOLEAN • VARCHAR (text <= 255 , size is defined by column SIZE) • INTEGER • LONGVARCHAR (long text > 255) • DATETIME • DATE • (empty) used for References in conjunction with column REF Other specific values are • VIRTUAL used for virtual fields, described later • FILE used for file upload • PLUGIN used for plugins, described later
REF	to deal with References
REQ	put x if the field is mandatory
INDEX	put x if the field is indexed in the database
DEFAULT	init value

Other columns, deal with view rendering

Those columns may apply to a TABLE or to a FIELD

TITLE	For a TABLE : titles for the List/Show/Edit/New views For a FIELD : label associated with the FIELD
CAT	used to group fields into Tabs
SHOW	For a TABLE : tells the number of items per page, for List view For a FIELD : controls the size of input fields in Edit view
HIDELIST	For a TABLE : controls the appearance of "delete" button For a FIELD : controls appearance of FIELD in List view
HIDEEDIT	For a TABLE : controls the appearance of "edit" button For a FIELD : controls appearance of FIELD in Edit view
HIDESHOW	For a FIELD : controls appearance of FIELD in Show view
HIDENEW	For a TABLE : controls the appearance of "new" button For a FIELD : controls appearance of FIELD in New view
ORDER	Used to order FIELDS
FILTER	Tells that the FIELD appear in Filter pane , inside List view
DP1	Used to dump FIELD to .csv file
DP2	Used to dump FIELD to .csv file
HELP	Content of the FIELD's help tip

6 Basic Features

6.1 Standard Views

Webdatabase provides 3 additional views in addition to the ones provided by Symfony

list	standard Symfony view , but limited to a subset of fields
edit	standard Symfony view
new	standard Symfony view
show	readonly view showing all fields
dump1	produces a .CSV file
dump2	produces a .CSV file

Those views are called via URL like `http://host/mywebsite/index.php/tablename/list`

Note that the Show view generate an autonomous HTML file (This means that is it contains all the javascript and css needed) You can store it in a local directory or in a mail : it will be rendered normally .

6.2 Adding a link in List view

Putting ? in the REF column for a field, will make this field transformed into a link in the List view.
The link will then open the Show view in a new window.

TABLE	FIELD	REF
workorder		
	Urgency	
	Description	?

workorder list

Urgency	Description	Leader	Date	Actions
Low	Reboot server	John Smith	2009/01/01 00:00	
1 result				

6.3 Dealing with References

Let's say that a Workorder must be fulfilled by a Worker

wd.csv will contain

TABLE	FIELD	REF
workorder		
	date	
	content	
	assignedto	worker
worker		
	name	?

`workorder:assignedto` will be implemented in the Database as INTEGER , and will contain `worker:id`

Putting a ? for `worker:name` tells to use this field when editing or showing `workorder:assignedto`
(when showing `workorder:assignedto` , Symfony follows the link and show the result of `Worker:_toString` function. This function is implemented to display the field marked with ? , here it is "name")

CAUTION
If MYTABLE has a field MYREF which is a reference to table MYTARGET , then you can't have another table MYOTHER with a field MYREF being a reference to MYTARGET :
That's because you can't reuse the name MYREF in another table : you should give him another name, for example MYOTHERREF
(otherwise mysql will issue an error)

WARNING : setting a default value for a Reference
You might decide that everybody can create a workorder, but only an authorized supervisor can assign it to a worker. This means the Field assignedto will only be visible to the supervisor.
Two solutions are possible:

- Do nothing . The Field assignedto will appear as NULL , which is not very user friendly
- Set a default value of 1 to Field assignedto . In this case, this value must be a valid ID inside table worker , corresponding for example with a dummy worker called "undefined" . .
  Be aware that if you load table worker by deleting old data without recreating the table, values of `worker:id` would not begin at 1

6.4 Choice lists as an alternative to references

Imagine we have different priorities for Workorders.
A classical method would be to list the possible values in a table "prior" and make a reference from Wordorder to this table. When editing workorder:priority a listbox will display the possible prior:level values

TABLE	FIELD	REF
workorder		
	date	
	content	
	priority	prior
	category	cat
prior		
	level	?
cat		
	name	?

There is a lighter alternative, that avoids to handle plenty of tables : tell Webdatabase to store the possible choices in an internal PHP class .

TABLE	FIELD	REF
workorder		
	date	
	content	
	priority	$PRIOR
	class	$CLASS

Putting $PRIOR (begins with a $) tells Webdatabase to make the list of possible choices from the file _choices.csv , with the following format :

PRIOR		
	urgent	urgent workorder
	medium	workorder might be delayed
	low	might be forgotten
CLASS		
	bugfix	bugfix
	improvement	improvement

Column 1 contains the key to search for inside _choices.csv
Column 2 is the value that will be stored in the database
Column3 is what is shown to the user

This will be converted to a php file: lib/model/_choices.php

6.5 Defining an initial value

Do not enclose init values by quotes.
For LONGVARCHAR fields, which appear in Edit mode as a multi-line textarea, it is possible to force newlines, by using \n
Note that some databases like mysql do not accept init values for LONGVARCHAR . So the init value is written in the class implementing the table, not in the database model.

TABLE	FIELD	TYPE	DEFAULT
workorder			
	date	DATETIME	2009-01-01 10:30
	content	LONGVARCHAR	First line of text\nSecond line of text

6.6 Setting Titles

The title for each field is to be placed in column TITLES

In addition to that, it is possible to define the titles for List, Show, Edit , New views : write them at the table level, in the order List,Show,Edit,New , separated by |
If nothing is set for New view , it will reuse what is set for Edit view

TABLE	FIELD	TITLE
workorder		title for LIST | title for SHOW | title for EDIT | title for NEW
	date	wished date
	content	description

6.7 Adding help tips

Each Field can have a Help tip : put it in column HELP

TABLE	FIELD	TITLE	HELP
workorder		list of workorders | workorder details | edit workorder | create workorder	
	date	wished date	The date when you would like the work to be done
	content	description	Tell here the main objectives of the workorder

6.8 Controlling which fields are showed to user

Ten levels are provided : Level0 ... Level9 that control which fields are displayed in List/Show/Edit/New views

The user enters levelX by issuing listX or newX action, corresponding to urls like
`mysite/index.php/tablename/listX` or `mysite/index.php/tablename/newX`
this sets the level, then forwards to List , New actions.

Subsequent List , New , Show , Edit will not modify the currently set level (this allows action `tablename/list3` to show a web page containing links like `tablename/show?id=yyy` that will keep the same level of displayed fields)

Remember : Only listX and newX exist . There is neither editX nor showX CAUTION : levels are common to all modules !

Levels are mutually exclusive : entering level X causes exiting level Y

Assigning levels for List view

By default all fields are showed. Putting a combination of figures 0 1 2 3 4 5 6 7 8 9 in column HIDELIST , will cause those fields visible in LIST view , only if the user has entered one of the associated levels

If you put a letter then, since there exists no associated level, the field will never be displayed. This is the method that allows List view to show only a subset of fields.

TABLE	FIELD	HIDELIST
workorder		
	date	
	content	
	assignedto	15
	budget	x

Field `assignedto` only appears through `workorder/list1 or workorder/list5` actions
Field budget is never shown

Assigning levels for Edit Show New views
Column HIDEEDIT, controls the visibility in Edit views
Column HIDENEW, controls the visibility in New views
Column HIDESHOW, controls the visibility in Show view

WARNING: To be consistent, if levelX makes a field visible in view New , then it should make it also visible to the view Edit !

TABLE	FIELD	HIDELIST	HIDESHOW	HIDEEDIT	HIDENEW
workorder					
	date				
	content				
	assignedto			5	5
	budget	x		5	5
worker					
	available	15	15	5	5

Only level5 allow to modify `workorder:assignedto` . But this field is always shown in List and Show views.

`worker:available` can be seen by level1 or level5 . but only level5 can modify it

Using `workorder/list5` will not display field budget , but will enter level5 which will cause the field to be visible through `workorder/edit` and `workorder/show`

6.9 Hiding Delete , Edit and New buttons

We use the mechanism of levels, but applied to tables
HIDELIST controls visibility of "delete" button
HIDEEDIT controls visibility of "edit" button
HIDENEW controls visibility of "new" button

REMARK
I strongly invite you **not** to define level0 : but to use it to cancel the current level in use.
Your root menus should call `xxxx/list0` to return to a state where no level is defined.

WARNING
If you try to hack by directly calling workorder/new while the current level would not allow the button New to be visible, then you will get a “Unauthorized” error message

TABLE	FIELD	HIDELIST	HIDEEDIT	HIDENEW
workorder		5	15	15
	....			

Button "delete" is only visible if you enter level5 through `workorder/list5` and `workorder/new5`
Buttons "edit" and "new" are only visible with level1 and level5

6.10 Ordering fields

You'd like not to change everything if your users require to change the order of appearance of fields.

Just put a number in column ORDER
You don't need to assign a number for all fields. Those having no ORDER value will be numbered incrementally starting at 100. So putting 10 for a field will cause it to appear first , putting 200 will cause it to appear last .

TABLE	FIELD	ORDER
workorder		
	date	
	content	
	assignedto	200
	budget	210
	details	
	sender	1

order will be : `sender,date,content,details,assignedto,budget`

6.11 Defining filters

View LIST provides filters.
To define which fields should appear in filter pane, just put x in the FILTER column

In addition to that, if you put 0 instead of x, then the checkbox "is empty" will be added for the field filter , which allows to filter items where the field is empty . (not available for references)

6.12 List view : setting how many items per page

Put that in column SHOW at the table level

TABLE	SHOW
workorder	10

LIST view will show 10 items per page

It is also possible to setup this information dynamically by using parameter "nb" in a request

`workorder/list/nb/50`	from now, we will show 50 items per page
`workorder/list`	50 items will still be shown
`workorder/list/nb/0`	come back to what has been defined in wd.csv (here 10)

6.13 Setting the size of INPUT tags in Edit view

This applies to fields containing text (VARCHAR , LONGVARCHAR)

For VARCHAR

- SIZE defines the size inside the database : it is also the max number of characters that the form will allow you to type in for this field
- SHOW defines the size in chars of the form input tag

For LONGVARCHAR

- SHOW defines the size of the textarea tag , according to the format COLSxROWS

For FILE

- SHOW gives the length of the area containing the filename of file to upload

TABLE	FIELD	TYPE	SIZE	SHOW
workorder				
	content	LONGVARCHAR		80x10
	assignedto	VARCHAR	50	20

6.14 Uploading files

This is done with fields whose TYPE is FILE.
They are implemented by VARCHAR(255) that will contain a modified file name.
The original file is not stored in the database, but in "`web/uploads`" directory

The filename is preserved, so that if you upload file `/some-dir/demand.txt`

- the name stored in database will be `table[field]----MD5 hash------!demand.txt`
- Inside List/Show/Edit views, the file will appear as "demand.txt" which will be a link to `uploads/table[field]----MD5 hash------!demand.txt`

A button "delete file" is automatically provided in Edit view
You can use column SIZE to specify the maximum size (in KBytes) of the uploaded file

Using a template file
Webdatabase can provide a template for the file to upload. If no file has been uploaded yet, it will show a link to the template.
To use this feature, just put the filename (without directory path) in column DEFAULT . Webdatase will seek for this file in directory `web/uploads`

TABLE	FIELD	TYPE	SIZE	SHOW	DEFAULT
worker					
	resume	FILE	10	150	resume.doc

In the above example, field "resume" will contain an uploaded file

Setting 150 in SHOW column gives the size of the input tag in Edit view
Setting 10 in SIZE column tells that the file must not be more than 10Kbytes
We use the template `web/uploads/resume.doc`

6.15 Using Tabs in Show and Edit views

It is possible to define Tabs in Edit or Show view

This is done by putting the Tab name in CAT column

TABLE	CAT	FIELD
workorder	E	
	1WHAT	date
	1WHAT	content
	2WHO	assignedto
	1WHAT	priority

Here , we create 2 Tabs , with names WHAT and WHO

The first character in the Tabs names does not appear in the view. It is used to define the order of appearance of Tabs in the view.
Here, first tab will be WHAT , second tab will be WHO

Tabs are not enabled by default . You have to put S or E or SE in the line describing the Table .

- If S is present , Tabs will appear in Show view
- If E is present , Tabs will appear in Edit and New view

In the above example, Tabs are defined only for Edit and New views

6.16 CSS Tuning

You can modify the look and feel for each table , by adding styles in `web/css/main.css`

The example below shows a specific tuning made for table Workorder

```
/* set the width of the filter pane in workorder/list       */
/* the sum sf_admin_container + sf_admin_bar  must be 99% */
#sf_admin_container #sf_admin_content.workorder-small {  width: 84%; }
#sf_admin_container #sf_admin_bar.workorder    {  width: 15%; }

/* set the label width and background  in workorder/show  */
.showtable .workorder-label { background-color: #EE0; width: 10%; }

/* set the label width and background  in workorder/edit  */
.edittable .workorder-label { background-color: #EE0; width: 10%; }
```

7 Pseudo fields

Those are fields which appear in List/Show view, but are the result of some computation, and do not really exist in the database.
So they do not appear in Edit view

7.1 Pointers

TABLE	FIELD	TYPE	REF
workorder			
	assignedto		`worker`
	service		`assignedto:service`
	workerinfo		`assignedto:resume`
worker			
	name	VARCHAR	?
	resume	FILE	
	service	VARCHAR	

In this example, a workorder is assigned to a worker. That's why `workorder:assignedto` is a Reference to the associated worker.

In addition to that, a workorder can directly show fields "resume" and "service" of the associated worker.
Those fields do not exist in the table workorder , they are computed after making a JOIN request to the database.

We say :

- `workorder:service` is a pointer to `worker:service`
- `workorder:workerinfo` is a pointer to `worker:resume`

The syntax in REF column is `source:dest`

- source= name of the field in the source table, that is a reference to the foreign table
- dest= name of the field in the foreign table

Note that the TYPE column is left empty : the TYPE is the one of the dest field . It even works with type FILE !

7.2 Virtual fields

It's a field whose value is computed from other fields in the same table.

This is done by using "VIRTUAL" for TYPE , and writing a computation rule in column DEFAULT.
The computation rule uses enclosing % to address other fields in the same table.
You can also add newlignes by using \n

TABLE	FIELD	TYPE	REF	DEFAULT
worker				
	name	VARCHAR	?	
	resume	FILE		
	service	VARCHAR		
	quickview	VIRTUAL		Name: %name%\nResume: %resume%

In the above example, we create a virtual field " quickview " whose content is a combination of fields " name " and " resume "

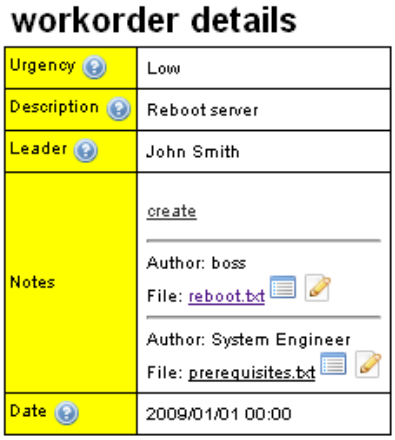
7.3 Backreferences

Lets consider now that a workorder should be linked to an unlimited number of notes, each note being attached to a workorder and only one.

Technically speaking, each "note" will have a reference to a "workorder" , and considering a workorder , we will get all the notes that refer to this workorder.

With backreferences , Webdatabase allows to create a "notelist" field in "workorder" table , with the following capabilities :

- Lists the associated notes
- For each note, provide 2 buttons to Show or Edit the note
- Provide a "new" button to add a new note



Three fields must be defined

- Table "note" must have a reference to table "workorder"
- Table "note" must have a Virtual field that represents a "note" (we will call it "V")
- Table "workorder" must have a backreference to field "V" of table "note" , with the syntax "note&v"

TABLE	FIELD	TYPE	REF	DEFAULT
note				
	author	VARCHAR		
	doc	FILE		
	v	VIRTUAL		`Author: %author% File: %doc%`
	wo		workorder	
workorder				
	notelist		note&v	

Important remark concerning Edit/Delete rights
BackReferences introduce a funny problem concerning whether associated items can be modified or not.

Following our example, a Note associated to a Workorder could be interpreted by two ways

- If we consider that a Note is like a file that we attach to a Workorder, then adding/modifying a note should be considered as a modification of the Workorder and reserved to some trusted people only.

- If we consider that a Note is like a comment added to a Workorder , then we should not consider them as a modification of the Workorder and many people should be able to create them.

The solution chosen by Webdatabase is the following :
Buttons to create/edit/delete a Note will depend on the New , Edit and Delete rights of table Note . That is :

- Button to create a Note will appear in both views `workorder/show` and `workorder/edit` , only if the New right has been defined for table Note (refer to column HIDENEW)

- Button edit a Note will appear in both views `workorder/show` and `workorder/edit` , only if the Edit right has been defined for table Note (refer to column HIDEEDIT)

- Button to Delete a Note will appear in both views `workorder/show` and `workorder/edit` , only if the Delete right has been defined for table Note

- Inside the view "`note/edit`", button Delete will appear only if the Delete right has been defined for table Note (and similarly button Save won't appear if Edit right has not been defined for table Note)

8 Plugins

A plugin is a field implemented as a LONGVARCHAR in the database, whose content is so complex that it will be managed by a Javascript addon.

Webdatabase will generate two html elements

- a hidden input field that will contain the field value
- an empty <DIV> section, whose inner content will be drawn by the javascript code

Webdatabase offers two plugins

- WDGRID which implements a grid
- MCE which is an encapsulation of the TinyMce html editor

It is easy to encapsulate an existing Javascript component , to be used within Webdatabase.

8.1 Usage

To use a plugin for a field, just do the following

- set column TYPE to PLUGIN
- put parameters in DEFAULT column , with the format `CLASS:initfile.txt`

CLASS is the plugin name
`initfile.txt` is the name of the file containing the field's init value , located in directory `data .` You can choose a different extension than .txt .
In addition to that , a file `initfile.json` must exist in the same directory than `initfile.txt` , that contains the javascript declaration of an associative array, containing the plugin's options for this field.

You can use the same plugin for different fields (unless it is badly designed) . Just provide a different `initfile`

When the plugin is called, a "mode" information is provided as parameter, whose possible values are `list,edit, show`.
This allows the plugin to have a different rendering depending on the view .
For example, WDGRID will provide buttons to add/delete rows/columns in edit mode , but will turn to a readonly behaviour in list or show mode.

TABLE	FIELD	TYPE	DEFAULT
workorder			
	details	PLUGIN	`WDGRID:details.txt`
worker			
	infos	PLUGIN	`MCE:infos.txt`

Field details uses details.txt and details.json
Field infos uses infos.txt and infos.json

How to define the init file

If the plugin internal format is complex, you may find it difficult to encode the init file (in our example `data/details.txt`) by hand.

You can use the following method :

- Using Webdatabase , create a new item .
- Edit the field content : put the initial value you want to have
- Save
- Look at the HTML source code : it will contain the init value for the field . Just copy it to the init file.

8.2 The WDGRID plugin

It 's a grid which allows to type-in data in a tabular form . It has the following capabilities :

- Insert / delete rows and columns
- Select rows , columns or a range of cells
- Copy / Paste rows, columns or cells .
- Paste can create new rows/columns if necessary
- Highlight row
- Force values inside a column to belong to a predefined list

Options

They look like:

```
{
size: [ 5,4 ],
noinsertcols: 1 ,
noinsertrows: 1 ,
choices: { status: [ "", "not started" , "in progress" , "closed" ] } ,
typecol: { 3: ["choice" , "status"] }
}
```

size	If there is no initial value, the grid will be created with 5 rows and 4 columns . Note that rows will be from 0 to 5 and columns from 0 to 4 . Row 0 and Column 0 exist , but are reserved for internal handling. They can't contain data
noinsertcols	If this option is present, buttons for insert/delete columns will be suppressed
noinsertrows	If this option is present, buttons for insert/delete rows will be suppressed
choices	Used for predefined lists of values. Here we define one list called "status" with values : `empty,not started,in progress,closed`
typecol	Used to assign a type to a column. Here column 3 will be of type "choice" (which means value are got from a predefined list) , and the list of values to use is "status" Note : Highlighted rows are used for titles. Normally, if a column is of type "choice" , then clicking inside will show a list a possible values. But if the row is highlighted, clicking will allow to type-in the cell text..

:

Internal format of Data

rows are separated by #n
inside a row, columns are separated by #t
inside a cell #v indicates a line feed
is the character

If a cell in column 0 contains "highlight" , then the row is highlighted

8.3 Interface : how to integrate your own Javascript component

Let's see how TinyMce editor has been integrated into Webdatabase , as a plugin called MCE

In Edit mode, this plugin uses TinyMce to show a wysiwig html editor. In List or Show mode , it just shows the HTML content of the field.

Let's consider that column DEFAULT contains `MCE:infos.txt`

How the Plugin is called

First the html page will call `web/mce/mce.php` (lowercases) using <?php require?> to declare all CSS and javascript needed .

Then the following Javascript is executed

```
mce_id = new  MCE ( IDCONTAINER , IDTARGET );
mce_id.init( 'MODE' , OPTIONS , 'URLBASE' );
```

- IDCONTAINER is the html id of the empty <DIV> section that will be drawn by the plugin
- IDTARGET is the html id of the hidden input tag containing the field value
- MODE will contain show,edit,list
- OPTIONS is replaced by the content of the options file, infos.json
- URLBASE is some stuff which is sometimes useful for accessing the plugins icons.
  (If the url is http://host/mywebsite/index.php/worker/edit it will contain mywebsite/mce)

In Edit mode , if the user clicks SAVE button to save the form, the following javascript is executed to repopulate the IDTARGET element, which Symfony will then save to database.

```
mce_id.updateTarget();
```

So we need to develop an interface `web/mce/mce.js` that will implement the class MCE and its 2 methods `init` and `updateTarget`

Step1: Store and reference the plugin's files

Store the tinyMce files under `web/mce` .
Create a file `web/mce/mce.php` (lowercases) which will be called (using <?php require ?>) from the html page, whose mission is to declare all CSS and JS files used for the plugin.
Note that this file is called only once , even if we use the plugin for several fields.

`web/mce/mce.php`

```
<?php  use_javascript('/mce/tiny_mce.js') ?>
<?php  use_javascript('/mce/mce.js') ?>
```

`tiny_mce.js` is the TinyMce javascript .
`mce.js` is our encapsulation

Since we like Show view to provide an autonomous HTML file (containing all necessay javascript and css) , we can improve this file , by using the parameter $action which is available when the file is called

`web/mce/mce.php`

```
<?php if ( $action != "show" ) : ?>
<?php  use_javascript('/mce/tiny_mce.js') ?>
<?php  use_javascript('/mce/mce.js') ?>
<?php else : ?>
<script language='javascript'>
     <?php require 'mce.js' ?>
</script>
<?php endif; ?>
```

`use_javascript` is a Symfony helper routine used to declare javascripts

Note that TinyMce does not need a CSS . See WDGRID plugin for an exemple with CSS .

Step2 : Create mce.js containing the class MCE

```
function MCE( zcontainer, ztarget)
{
  var idcontainer=zcontainer;
  var objcontainer = document.getElementById( idcontainer);
  var objtarget=document.getElementById(ztarget);

  this.init=function( zaction, zoptions, zurlbase)
  {
    if ( zaction == "edit") this.initedit(zoptions,zurlbase);
    else  this.initshow(zoptions,zurlbase);
  }

  // "show" behaviour
  this.initshow=function(  zoptions, zurlbase)
  {
    objcontainer.innerHTML = objtarget.value;
  }

  // "edit" behaviour
  this.initedit=function(  zoptions, zurlbase)
  {
    // inject the initial value to the container
    objcontainer.innerHTML=objtarget.value;

    // launch an instance of tinyMCE linked with the container
    zoptions ["mode"] = "exact" ;
    zoptions["elements"] = idcontainer ;
    tinyMCE.init( zoptions );
  }

  //routine to retrieve the value in the target
  this.updateTarget=function()
  {
    var out=tinyMCE.editors[idcontainer].getContent() ;
    objtarget.value= out.replace(/\r?\n/g, ' ');
  }

}
```

The class constructor does nothing but memorizing parameters.

The `init` method has two behaviours

- in Show mode, we don't need TinyMce to run . We just copy the value from the hidden input field (whose content comes from the database) to the empty <DIV> section
- in Edit mode , we add a few options to TinyMce , telling him to draw itself inside the <DIV> section , tell we launch it with `tinyMCE.init`

The `updateTarget` method retrieves the value from TinyMce, and stores it back to the hidden input field.

Step3 : defining the options file

data/infos.json

```
{
theme: "advanced",
theme_advanced_buttons1 :
"bold,italic,underline,|,bullist,numlist,|,outdent,indent,|sub,sup,|,justifyleft, ...",
theme_advanced_buttons2 :
"undo,redo,|,link,unlink,anchor,image,|,code,preview,|,hr,removeformat,visualaid,charmap",
theme_advanced_buttons3 : "",
height : "100%",
width: "100%"
}
```

This collection of options is specific to TinyMce . It defines the toolbar's content.

You might decide to hardcode those options inside mce.js .

9 Managing Access Control

9.1 Principles

It is possible to restrict some Urls to users depending on some authorization level.
Authorization is not done on a user basis, but on a group basis.
We extract from the Url the associated `module/action` . (Refer to Chapter 3 for the definition of module and action)
If the Url is protected, this `module/action` is associated with a group
To access the Url , the user must login to the group.

User logins to a group by entering the group name and the associated password.
Thus, it is possible for a user to login to several groups , and finally to disconnect from all groups.

This will use the standard HTTP authorization mechanism : when trying to access a protected page, the user will be prompted to enter a username and a password. The username is the group name, and the password must match the group's password.

We use DIGEST authentication . If you agree to loose logout capability , you can move back to BASIC authentication, by modifying class wdAddon in `lib/wdAddon.class.php`

CAUTION
If you want to use authentication, then you should increase the timeout of the user session to something like 12hours. Otherwise, user may suddenly loose access in the middle of his work .

`apps/backend/config/factories.yml` (never use Tabulation in .yml file , use blanks to indent)

`all:`
`  user:`
`  param:`
`  timeout: 43000`

9.2 Built-in implementation

Webdatabase comes with a simple built in implementation, which is managed by class `lib/_auth.class.php`
This class uses two arrays. You just need to modify those arrays according to your needs :

- $keys gives the groupname required for a module/action
- $passwd contains the password associated with the group

When issuing a request, `module/action` is searched for match with elements in $keys.
If match, the password is asked.

$keys can contain

`module/action`	`this specific module/action is protected`
`module/`	`all actions of the module are protected`
`/action`	`action is protected whatever the module ( typically used to protect a level like list5 )`
`/.*1`	`use regular expression` `all actions of level1 will be protected ( list1 new1 )`

By modifying _auth.class.php , you can easily

- build a more complex protection mechanism
- use something simpler like a unique group .
- disable authentication by setting $keys an empty array

9.3 Interface

You can provide your own authentication mechanism .
Just rewrite class `lib/_auth.class.php`

This class must implement 6 methods, using a parameter $todo containing `module/action`

```
// tells if $todo requires login to a group
// if yes, return array( groupname, passwd )
// if no return ""
public static function getSecret($todo)

// tells if $user is authorized  for $group
// returns true or false
public static function isOK($user,$group)

// memorizes the fact that $user is authorized for $group
public static function setOK($user,$group)

// memorizes the nonce for DIGEST authentication
public static function setNonce($user,$nonce)

// gets the nonce for DIGEST authentication (previously stored by setNonce )
public static function getNonce($user)

// logout user from all groups
// IMPORTANT : this must also remove all credentials , to avoid keeping some undue
//              visibility
public static function logout()
```

10 Making the main menu page

[this chapter gives only minimal information . See Symfony documentation for more details]

10.1 Basics

Symfony is configured so that `http://localhost/mywebsite/index.php` is equivalent to `http://localhost/mywebsite/index.php/default/index`

In order to respond to the Url `default/index` , Webdatabase has implemented 2 files :

- `apps/backend/modules/default/actions/actions.class.php`
  You don't need to modify this file
- `apps/backend/modules/default/templates/indexSuccess.php`
  This is the final main menu template, which you must modify according to your needs

indexSuccess.php

```
Administrator menu
<ul>
<li><?php echo link_to("Dump list of Workorders", "workorder/dump1?file=wo") ?></li>
<li><?php echo link_to("Create/Modify Workorders", "workorder/list1") ?></li>
<li><?php echo link_to("View Workers (full view)", "worker/list1") ?></li>
</ul>
```

link_to is a helper routine that produces an HTML url

- first parameter is the text
- second parameter is the target of the url , with the format `module/action?param1=param1value`

actions.class.php

```
<?php

class defaultActions extends wdActions
{
  public function executeIndex($request)  { return sfView::SUCCESS }
}
```

Since the module name is `default`, we must implement a class `defaultActions` .
Since it must respond to action "`index`" , this class must implement a method `executeIndex`
By returning sfView::SUCCESS this routine tells Symfony to show the template indexSuccess.php

Note:
Normally, according to Symfony rules, `defaultActions` should inherit from class `sfActions` . But since we may use the forcing mechanism (see dedicated chapter) , we must use `wdActions` instead .

10.2 Add the Logout menu item

We will add an action `default/logout` that will disconnect the user from all groups. (see previous chapter concerning authentication)

indexSuccess.php

```
<li><?php echo link_to("logout user", "default/logout") ?></li>
```

actions.class.php

```
<?php

class defaultActions extends wdActions
{
  public function executeIndex($request)  {return sfView::SUCCESS;}

  public function executeLogout($request)
  {
   _auth::logout();
   $this->forward("default" , "index");
  }
}
```

We must implement a method `executeLogout`.

- it does the work, using class `_auth`
- it then forwards to `default/index` in order to return to the main menu

10.3 Add a link to the main menu in all pages

File `apps/backend/templates/layout.php` contains the global layout of pages.
The pages content is inserted by `<?php echo $sf_content ?>`
We just need to add a link to the main menu before

```
<body>
  <?php echo link_to("Main menu" , "default/index"); ?>
    <?php echo $sf_content ?>
</body>
```

10.4 Modify the "Page not found" template

Just modify `apps/backend/modules/default/templates/error404Success.php`

11 Forcing

Consider now that we receive Workorders from different clients , and we would like to manage them as separate accounts.

- The first solution is to duplicate our database : one for each client.
- The second solution is to use the forcing mechanism . Have only one database , but in the main menu page choose the client we want to work with .

Forcing the client will have the following consequences

- The list of workorders will be reduced to the ones associated with the forced client.
- When editing a workorder , the client field will be predefined and not modifiable.
- Backreferences will work in a similar way : when getting all the workorders assigned to a worker, the list will be limited to the workorders associated with the forced client.

Forcing is implemented by storing some information is the user session. Once a table is forced, it remains forced, until forcing is cancelled by setting an empty value.

CAUTION
If you intend to force client, then the filter pane in page workorder/list should include the item client , so that user can see that the list of workorders is reduced to ones associated with the forced client. Otherwise, user would not understand why all the workorders are not shown.

Usage
To implement forcing, you need to insert 4 lines in your main menu page

```
<?php use_helper('wd') ?>
<?php use_javascript('/wd/js/wd.js', 'first') ?>
```
Those Symfony commands make the Webdatabase stuff available

```
<?php echo force_f( "ID", "TABLE") ?>
```
This shows a select box containing all the values of the table we want to force.
ID is the id you chose for the html select element
TABLE is the name of the table in the database

```
<?php echo link_f("TEXT","TARGET","ID","TABLE") ?>
```
This shows a link whose text is TEXT .
Clicking this link will launch Url TARGET by forcing table TABLE to the value selected in the select box whose html id is ID. For example, TARGET can be your main menu page `default/index` of a list view `workorder/list` .
If the forced value is empty, then forcing is cancelled.
You can replace `link_f` by `popup_f` . In this case, the link will open in a new window.

12 Utilities

12.1 Dump data to a .CSV file

It is possible to dump all fields (or a subset) of a table to a .csv file.
Be aware that some Excel versions do not support more than 1024 characters in a cell.

For a table xxx, two dumps are possible, by getting the url
`xxx/dump1`
`xxx/dump2`
those urls will show a page containing a link to file `/dump/xxx.csv`

Defining which fields will appear in the dump is controlled by column DP1 (for dump1) and DP2 (for dump2)
Just put a number or a letter in those columns.

Ordering the fields
You can define the order of appearance of fields in the .CSV file : the content of DP1 , DP2 will be sorted, and the resulting order will define the order in the .CSV file

Adding extra columns
It may happen that the CSV file you have to produce, needs to respect some predefined column numbers.
You can do that by putting extra dummy fields in your database model.
It is easier to ask Webdatabase to add some empty columns after fields.

The content of DP1 , DP2 will be `X:Y`

- X = number defining the order of the fields in the CSV
- Y = number telling how many empty columns to add after the field

Scope
The Dump feature also works with fields which are References or Pointers, implementing a JOIN
In this case, we dump the foreign field in the foreign table.

Other pseudo fields like Virtual field or Backreference are never dumped

Options

It is possible to control the first line of the CSV dump , in order to add a header.
`xxx/dump1?cmd=0` no header
`xxx/dump1?cmd=1` header made with the titles declared in column TITLE
`xxx/dump1?cmd=2` (default) header made with the field names declared in column FIELD

It is also possible to control the filename
`xxx/dump1?file=myfile` produces file `/dump/myfile.csv`
note that only letters,figures and _ are allowed in the filename

Implementation

The SQL request is written in the PHP code in lib/model
During execution, the data is directly written to the .CSV file, without populating PHP objects.

Launching Dump from command line
`php wd -dump1`
`php wd -dump2`
this generates `dump-xxx.csv` in the current directory, for all tables XXX that have fields with DP1 or DP2 set in wd.csv

12.2 Load data from .CSV file

Webdatabase will detect xxx.csv files in the data/fixtures directory, where xxx is the name of a table.

It will find out which columns in the .csv file correspond to existing fields in the table (other columns are ignored), then produce the .yml file that will be loaded by the command php symfony propel:data-load.

The yml files generated have a name like `@@pppp-xxx.yml`
where

- xxx is the table name
- pppp is a prefix computed by WebDatabase so that the files will be loaded in the right order by Symfony. This needs that there is no circular references in the tables (like A references B which references C which references A)

The first line of the .CSV file must contain fields name.
Webdatabase will ignore what is after a blank or newline in those names, then keep those which match a field name in the table . Comparison is case insentive . for example BOSS NAME will be interpreted as "boss"

Automatic value adjustment

- For BOOLEAN fields, an empty or 0 value will be converted to 0 . Everything else is converted to 1

How to launch it

Generate whole stuff , including data for loading
```
php wd
```

Generate data for loading only
```
php wd light
```

Recreate database
```
php symfony propel:insert-sql
```

Load files , erasing previous data
```
php symfony propel:data-load –-application=backend
```

Caution !
If you don't recreate the database before loading data , "id" fields (which are automatically incremented) will not start at 1 , which might eventually disturb you if you made such assumption.

If a field is defined with a ? in the wd.csv REF column, then follow those rules :

- this field init data should not contain duplicate values (otherwise only one line will be kept)
- this field init data should not contain newlines (otherwise the yml outpout file will be incorrect)

13 Incompatibilities from 1.1

A few features have been modified from Webdatabase 1.1

Action “create” does not work anymore . You should use action “new” instead