

Table of contents

- Tutorial for Database interfacing
 - Things to note:
 - Working with these examples
 - Examples
 - Display all data from a table:
 - Display data from a table with nested links
 - Adding new records
 - Counting records in a table
 - Using the Navigator
 - More
-

Display all data from a table:



```
{DATATABLE()} {DATATABLE}
```

```
{REQUEST()}  
SELECT * FROM Students  
{REQUEST}
```

Comments:

- Your SQL statements are executed using the REQUEST plugin, notice it is at the bottom of the page.
- In this example the REQUEST plugin has no db parameter, so it will take the default [DSN](#) (=dsn named *default*).
 - The DATATABLE plugin generates a table from the SQL statement found in the REQUEST plugin
- When the DATATABLE is used without any parameters it will displays the data from the first 'default' request

Display data from a table with nested links



```
{DATATABLE(title=>1)}
  Id_|_First name_|_Last name_|_'Links''
*Id|*First_Name|*Last_Name|LINK:More info:StudentInfo:Id=*Id
{DATATABLE}

{REQUEST()}
SELECT Id, First_Name, Last_Name FROM Students
{REQUEST}
```

Comments:

- The *DATATABLE* plugin has the *title* parameter set, so the first row is the title row.
 - Rows are separated by | , as are Tiki tables.
- The second line contains the template for the remaining rows. This relies on normal wiki formatting such as links. Database field names have a * before them (*First_Name)
 - The SQL statement must specify each field name to be returned
- The link in the Links column is intended to provide more information on the student selected, see 'Using the navigator' example.

Adding new records



```
{DATATABLE(title=>1)}
  Id. | __First name__ | __Last name__ | __''Links''__
*Id|*First_Name|*Last_Name|LINK:More info:StudentInfo:Id=*Id
{DATATABLE}

  {FORM(update=>Create)}
  __First name:__ {FIELD(name=>First_Name)}{FIELD}
  __Last name:__ {FIELD(name=>Last_Name)}{FIELD}
  {FORM}

  {REQUEST(cursor=>1)}
  SELECT Id, First_Name, Last_Name FROM Students
  {REQUEST}

  {REQUEST(id=>new_student,cond=>update)}
  INSERT IGNORE INTO Students SET First_Name='?First_Name', Last_Name='?Last_Name'
  {REQUEST}
```

Comments:

- A FIELD can be used to display or edit data. A FORM can create a submit button which will perform an action involving the encapsulated FIELD's.
 - The *update* parameter of the *FORM* plugin is simply some text for the *Submit* button, which will display automatically unless you specify it.
- The *cond=>update* parameter on the *REQUEST* field tells that this *REQUEST* will only be executed if an *update* parameter has been given to the page, which is automatically the case when you submit a form using *{FORM}* (unless you specify otherwise with the *noupdate* parameter in *FORM*)
- The *id* parameter is the name of the request, which you could use in *FIELD* or *DATATABLE* with their *from* parameter. If you don't specify a name, unnamed requests will be numbered, beginning with 0 (which will be the default request).
 - If you set the *cursor=>1* on *REQUEST*, the data table will display navigation buttons for moving between result pages.

Counting records in a table



There are __{FIELD(edit=>0,from=>Row_Count, src=>Record_Count)}{FIELD}__ Rows in this table

```
{REQUEST(id=>Row_Count)}  
SELECT COUNT(*) Record_Count FROM Students  
{REQUEST}
```

Comments:

- This will display the number of rows in the table
- Notice how the FIELD statement is mixed in with the wiki formatting

Using the Navigator



```
{FORM(id=>Form_Name, from=>Request_Name)}
__Text Box: __ {FIELD(name=>Field_Name1, from=>Request_Name, src=>First_Name)}{FIELD}
__Check Box: __ {FIELD(type=>chk, src=>Last_Name, size=>150, from=> Last_Name)}{FIELD}
__Comment box: __ {FIELD(type=>memo, src=>Last_Name, from=>Request_Name)}{FIELD}
__Select: __ {FIELD(type=>select, src=>First_Name, from=> Request_Name, choices=>Student, ch_field=>Student.First_Name, nopos=>1)}{FIELD}
{FORM}

{NAVIGATOR(from=>Request_Name, key=>Id, table=>Student)}{NAVIGATOR}

{REQUEST(id=Request_Name)}
SELECT Id, First_Name, Last Name from Students where Id='?Id'
{REQUEST}
```

Comments:

- The NAVIGATOR will product the following error 'Navigator: position unknown..' at the end of the URL you will need to pass &Id=1 now the Navigator will work and the fields will populate.
 - The check box will be ticked if there is data in the field, if the field is null then the check box will be empty
- nopos= if the table contains a column 'Pos' it can be used to order the options. Set this value to 1 if you do not have this column in the table.
 - choices=DNS name/SQL table name. if using default as the DNS use just the SQL table name.
 - ch_field= field name in the sql statement to return the results from (Table_Name.Field_Name)
 - noadmin= removes the admin link on combobox's, default is active

