

# CGI 4

---

## DBSearch vers 2

### CGI.pm

It has lots of features, but for now we'll just use it to extract bindings.

**use CGI;**

**\$query = new CGI;**

**\$param = \$query->param('param-name');**

Will be undef if there is no such binding.

## Maintaining State

**page #1 : pick database**

**page #2 : search in that database**

**State?**

When the form is submitted from page #2, how do we know which database to search?

**hidden field**

Put a hidden field in the form to remind you (the server) what's going on when the form comes back.

We will look at other solutions too.

## Code

```
#!/usr/bin/perl -w
## dbsearch.pl
## Re-written to use CGI.pm
# Implements a simple search on a local tab-delimited database
# and returns the results in an HTML table.
# When called with no arguments, returns a simple search form.
# A submit of the search form returns a table of all the
# records which contain the target search string.

use CGI;

##
```

```

## Header
##
$header = <<EOT;
<html><head>
<title>Database Search</title>
</head>
<body bgcolor=white>
<h1>Database Search</h1>
EOT

$strailer = "</body></html>\n";

##
## Start Output
##
$| = 1;    ## set STDOUT to be unbuffered
print "Content-type: text/html\r\n\r\n";
print $header;
$query = new CGI;  ## this allocates the CGI module state

my(@filenames, $fileNum, $target, $filename);

# List of filenames we are willing to open. This is the simplest way
# to make sure that the client only opens the server files we really intend.
@filenames = ("movies.txt", "books.txt");
$fileNum = -1;
$filename = "";

if (defined($query->param('fileNum'))) {
    $fileNum = $query->param('fileNum');
    $filename = $filenames[$fileNum];
}

##
## Choose Form
##
## (uses the "<<" trick to put the text in the file)
$chooseForm = <<EOT;
<h2>Choose Database</h2>
<p>
<form  method=get>
<select name=fileNum>
<option value=0>movies
<option value=1>books
</select>
<input type=submit name=choose value="Choose Database">
</form>
EOT

##
## Search Form

```

```

##
## Uses a hidden field to remember its fileNum
$searchForm = <<EOT;
<h2>$filename</h2>
<p>

<form method=get>
<input type=hidden name=fileNum value="$fileNum">
Search string:<input type=text name=target size=20><br>

<p>
<input type=submit name=search value="Submit Search">

</form>
EOT

## If there is no fileNum, give 'em the chooser
if (!defined($query->param('fileNum'))) {
    print $chooseForm;
    print $trailer;
    exit(0);
}
## Otherwise, if there is fileNum but no target
elsif (!defined($query->param('target'))) {
    print $searchForm;
    print $trailer;
    exit(0);
}
else {
## Put out the answer with another copy of the search form
    $target = $query->param('target');

    print "<p>Search for '<b>$target</b>' in the database '<b>$filename</b>'\n";

    open(DATA, "$filename") || ReportError("Cannot open '$filename'");

    ## Make a table out of all the rows which matched...
    print "<table border=1 width = 100%>\n";

    my($labels);
    $labels = <DATA>;      # grab the labels from the first line
    chop($labels);
    TableRow($labels, "th");

    my($count) = 0;
    while (defined($line = <DATA>)) {
        $target = quotemeta($target);
        if ($line =~ /$target/i) { # Case insensitive search for target
            chop($line);
            TableRow($line, "td");
            $count++;
        }
    }
}

```

```

}

close(DATA);

print"</table>\n";

# print count with the pluralization correct
if ($count == 1) { print "<p>1 matching record\n"; }
else { print "<p>$count matching records\n"; }

## Another copy of the search form at the bottom...
print "<hr>$searchForm\n";

print $trailer;
exit(0);
}

# Helper which prints out one row of a table.
# The text should be tab delimited, and the type
# should be "th" for table header, or "td" for
# a table row (the default).
sub TableRow {
    my($text, $type) = @_;

    my(@fields, $elem);

    @fields = split(/\t/, $text, -1);

    print "<tr>\n";
    foreach $elem (@fields) {
        print "<$type>$elem</$type>";
    }
    print "\n</tr>\n";
}

## Print out an error message to HTML and exit
sub ReportError {
    my($err) = @_;

    print "<h1>Error</h1><p>$err\n</html>\n";
    exit(0);
}

```



