ECM3408 - CA1: Skydrive

62007094

February 15, 2015

1 SQLite Database Creation

The following commands can be used to create a table called `documents`

```
$ sqlite3 skydrive.sqlite3
sqlite > CREATE TABLE documents
    ...> ( id INT PRIMARY KEY
    ...> , name VARCHAR
    ...> , message VARCHAR ); # sqlite does not impose length-limits
       on fields, so none is give.
.quit
```

1.1 Testing

Script output of creating a database as described above, storing an unencrypted message and retrieving it:

```
bash-3.2$ sqlite3 skydrive.sqlite3
SQLite version 3.8.5 2014-08-15 22:37:57
Enter ".help" for usage hints.
sqlite> CREATE TABLE documents
   ...> ( id INT PRIMARY KEY
   ...> , name VARCHAR
   ...> , message VARCHAR );
sqlite> INSERT INTO documents VALUES
   ...> (1, "Name", "Message");
sqlite> SELECT * FROM documents;
1|Name|Message
sqlite>
```

2 The Ruby Code

Please note that the class `caesar` was implemented after a model-answer for the problem.

```
#!/usr/bin/env ruby -w
require "sqlite3"
require "webrick"
DATABASE = "skydrive.sqlite3"

def model_list()
        db  = SQLite3::Database.new( DATABASE )
        qry = "select id, name from documents;"
        hash = db.execute( qry )
        db.close
```

```
11          return hash
12  end
13
14  def model_show(id)
15          db  = SQLite3::Database.new( DATABASE )
16          qry = "select message from documents "
17                "where id = \"#{id}\""
18          val = db.get_first_value( qry)
19          db.close
20          return val
21  end
22
23  def view_show(message, shift)
24      cipher = Caesar.initialise(shift)
25      message_plain = cipher.decrypt(message)
26      "<html>"                                                    +
27      "  <body>"                                                  +
28      "    <p>" + message_plain.to_s + "</p>                      +
29      "  </body>"                                                 +
30      "</html>"
31
32  def view_list(vals)
33          output = "<html>"                                              +
34          "  <body>"
35
36          vals.each do |key, value|
37            output <<       "<a href=\"/show/#{key}\">#{value}</a><br \>"
38          end
39
40          output << "    <form action=\"http://localhost:3000/list\""    +
41          "           method=\"GET\">"                                   +
42          "       <input name=\"code\" value=\"Enter id\"/>"             +
43          "       <input type=\"Submit\"/>"                              +
44          "    </form>"                                                  +
45          "  </body>"                                                    +
46          "</html>"
47  end
48
49
50  class Controller < WEBrick::HTTPServlet::AbstractServlet
51      def do_GET ( req, rsp )
52        case req.path
53          when "/list"
54            rsp.status = 200
55            rsp.content_type = "text/html"
56            rsp.body = view_list( model_list() )
57          when "/show"
58            id = req.query[ "id" ] || ""
59            rsp.status = 200
60            rsp.contetn_type= "text/html"
61            rsp.body = view_show( model_show(id), 3)
62        end
63      end
```

```
64 end
65
66 class Caesar
67   def initialise(shift, alphabet = ('a'..'z').to_a.join)
68     chars = alphabet.chars.to_a
69     @encrypter = Hash[chars.zip(chars.rotate(shift))]
70     @decrypter = Hash[chars.zip(chars.rotate(-shift))]
71   end
72
73   def encrypt(string)
74     @encrypter.values_at(*string.chars).join
75   end
76
77   def decrypt(string)
78     @decrypter.values_at(*string.chars).join
79   end
80 end
81
82
83 server = WEBrick::HTTPServer.new( :Port => 3000 )
84    server.mount( "/", Controller )
85    server.start
```