Faculty of Computer Science, Dalhousie University *21-Jan-2025*

DGIN 5201 — Digital Transformation

Lecture 7: Lec 5: Back-end Processing

Location: LSC C236 Instructor: Vlado Keselj
Time: 13:05–14:25

Previous Lecture

- Notices: Lab 1 deadline postponed, Emacs notes
- Example e3:
  - Simple password protection in Apache
  - Concepts review: rsync, htpasswd, .htaccess
- Unix-style customization
  - Text-based customizations
  - Examples: Bash, Emacs, Apache
- Example e4:
  - HTML form and form fields
  - Concepts review: form, text field, drop-down list

4 Example e5: Backend Server Processing using CGI

Back-end Processing

- How to
  - receive data at the server end from the client (browser)
  - process data and send some results back to the client
- This is called Back-end processing
- Some options to implement:
  - Apache has a way to help:
    - CGI — Common Gateway Interface
  - Use a server: build one or use server-based options

CGI Processing

- CGI — Common Gateway Interface
- Implemented as CGI program; e.g.: `prog.cgi`
- When requested, executed by Web server and output returned
- Input prepared by Web server

Example e5: Backend Server Processing using CGI

- Using `rsync` copy `e4` to `e5`

- Let us first check that CGI scripts are working by creating file `test.cgi` in e5 as follows:

```
#!/usr/bin/perl
use CGI qw/:standard/;

print header;
print "<html><body>Test\n";
```

- The file should be user executable, without permissions to the group and others (`rwx------`)
- Run the command `./test.cgi` and you should get a simple output as follows:

```
Content-Type: text/html; charset=ISO-8859-1

<html><body>Test
```

- Check in browser: `https://web.cs.dal.ca···dgin5201/e5/test.cgi`

Perl Scripting Language

- What we saw in `test.cgi` program is example of a Perl program
- Perl is a scripting language, similar to Python and PHP
- Provides convenient and quick data preprocessing
- Text processing oriented
- Appropriate for rapid prototyping, and CGI programming

Example e5: Preparing form for processing

- Modify `index.html` the table part:

```
<form method="post" action="register.cgi">
<table>
<tr><th align=right>First and last name:</th>
<td><input type="text" name="name"></td></tr>
<tr><th align=right>Email:</th>
<td><input type="text" name="email"></td></tr>
<tr><th>Certificate (DB, HI, DS):</td>
<td><select name="certificate">
  <option>DB</option><option>HI</option>
  <option>DS</option></select></td></tr>
<tr><td align=center colspan=2>
<input type="submit" value="Submit"/></td></tr>
</table>
</form>
```

Example e5: Processing Data

- Prepare user executable file `register.cgi`:

```
#!/usr/bin/perl
use CGI qw/:standard/;
print header;
print "<html><body><h1>Registration</h1>\n";
print "<p>The following registration is received:\n";
$name  = param('name');  $email = param('email');
$certificate  = param('certificate');
print <<"EOT";
<table>
<tr><th align=right>First and last name:</th>
<td>$name</td></tr>
<tr><th align=right>Email:</th><td>$email</td></tr>
<tr><th>Certificate (DB, HI, DS):</td><td>$certificate
</td></tr><tr><td align=center colspan=2>
<a href="index.html">Back to Registration Page</a></td>
</tr></table>
EOT
```

Example e5: Processing Data and Testing

- Submit some registrations and make sure `register.cgi` works well
- This completes Example e5

Concepts Review: Example e5

- Server-side processing, concept of CGI (Common Gateway Interface)
- Perl programming language, Perl with CGI
- `<form method="post" action="...">`
- `<input ...  name="x">`
- `<input type="submit" value="Submit/>`
- CGI processing in Perl

Example e6: Saving Registration Data: Implementation

- Using `rsync` copy `e5` to `e6`; adjust `.htaccess`
- To save registration, add the following line in the script `register.cgi`:

```
...
$email = param('email');
$certificate  = param('certificate');

&save_registration($name, $email, $certificate);

print <<"EOT"; ...
```

- and we add the following function at the end of the program:

```
sub save_registration {
  my ($name, $email, $certificate) = @_;
  open (my $fh, ">>registrations-saved.txt") or die;
  print $fh "\nname: $name\nemail: $email\n".
    "certificate: $certificate\n";
  close($fh);
}
```

Example e6: Saving Registration Data: Testing

- First check syntax: `perl -c register.cgi`
- Test the web site by making several registrations
- Check that registrations are saved in the file `registrations-saved.txt`
- Check permissions of `registrations-saved.txt`
  - If not all-readable, make them all-readable
  - Verify that the file is accessible on the web **(!)**
- Change the permissions of `registrations-saved.txt` to user-only readable and writeable
- Check accessibility on the web; **Lesson learned!**
- Check that the application still works

Concepts Review: Example e6

- Perl subroutine (similar concepts: procedure, function)
- Saving and appending data to a file
- Importance of file permissions
- Possible additional issues to deal with files: concurrency (race conditions), efficiency
- Alternatives: using databases, server or file-based