

Laboratory work №2

In this laboratory I chose blog as web project, implementing several web patterns like MVC, template view, transform view and two-step view. I implemented 3 pages, that include main page, gallery page and contact page using HTTP server class in Java. An embedded HTTP server can be added to a Java program using classes from the package `com.sun.net.httpserver`. The method described here has three steps:

1. Construct an HTTP server object.

```
import java.io.*;
import java.net.InetSocketAddress;
import com.sun.net.httpserver.HttpHandler;
```

It is not possible to directly construct an `HttpServer` instance because it is an abstract class. Therefore I used `create()` method:

```
HttpServer server = HttpServer.create(new InetSocketAddress(8000), 0);
```

The first argument is the socket address (IP address and port number) on which the server should listen. In this instance only the port number has been specified. This causes the server to be bound to the wildcard IP address, allowing it to accept connections via any local network interface. The second argument is the backlog of outstanding connections that the operating system should queue while they are waiting to be accepted by the server process. If set to zero then a default value is used.

2. Attach one or more HTTP handler objects to the HTTP server object.

HTTP server object needs to be told what content to serve in response to a given URL. This is done by attaching one or more handler objects to the server. Handlers must implement the interface `com.sun.net.httpserver.HttpHandler`. It has a `handle()` method that the server calls in response to each HTTP request.

The handler must send a set of HTTP response headers back to the client by calling `sendResponseHeaders()`, send the body of the HTTP response to an `OutputStream` obtained by calling `getResponseBody()` and close the above output stream to indicate that the response is complete.

```
static class MainHandler implements HttpHandler {
```

```
    public void handle(HttpExchange t) throws IOException { ...
```

```
        String response = "404 (Not Found)\n";
```

```
        t.sendResponseHeaders(404, response.length());
```

```
        OutputStream os = t.getResponseBody();
```

```
        os.write(response.getBytes());
```

```
        os.close();
```

```
    }
```

```
}
```

The first argument `sendResponseHeaders()` the HTTP status code. A value of 404 means that the URL was not found. The second argument to `sendResponseHeaders` is the length of the response body in bytes.

While using the handler class, it must be instantiated then attached to the HTTP server object:

```
server.createContext("/", new MainHandler());
```

The path must be an absolute one so always begins with a forward slash character.

3. Start the HTTP server.

The HTTP server will not accept connections until the method `server.start()`;

Control returns to the caller immediately: the server runs in a background thread.

So, within a handler the requested URL can be obtained from the `HttpExchange` object using its `getRequestURI()` method. For normal HTTP requests this contains only the path and perhaps also a query string. The path can be extracted from the URI object using its `getPath()` method.

```
public void handle(HttpExchange t) throws IOException {
```

```
    String path = t.getRequestURI().getPath();
```

```
    if (path.equals("/") || path.equals("/main")) {
```

```
        Header.render(t);
```

```
        Main.render(t);
```

```
        Footer.render(t);
```

When serving content via HTTP you should specify the type of the data if it is known.

In controller for:

Image:

```
private static class ImageHandler implements HttpHandler {
```

```
    public void handle(HttpExchange t) throws IOException {
```

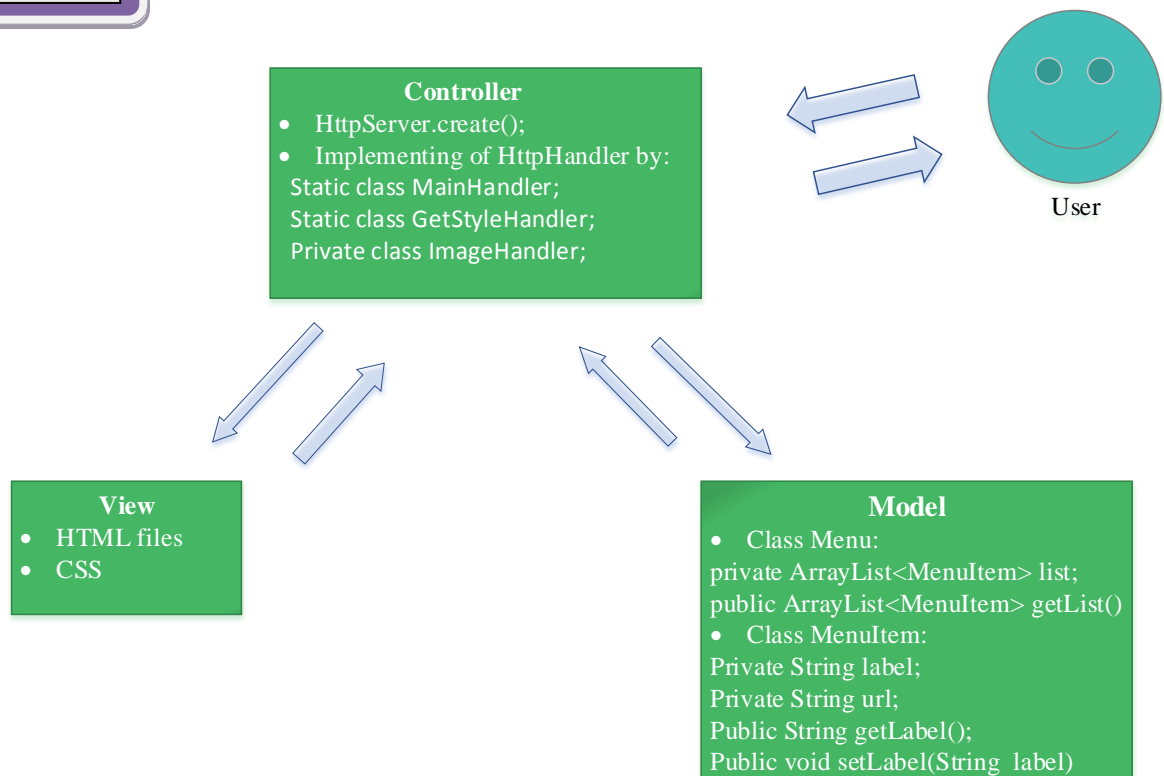
```
        File image = new File("image.png").getCanonicalFile();
```

Css:

```
static class GetStyleHandler implements HttpHandler {
```

```
    public void handle(HttpExchange t) throws IOException {
```

```
        File file = new File("template/style.css").getCanonicalFile();
```

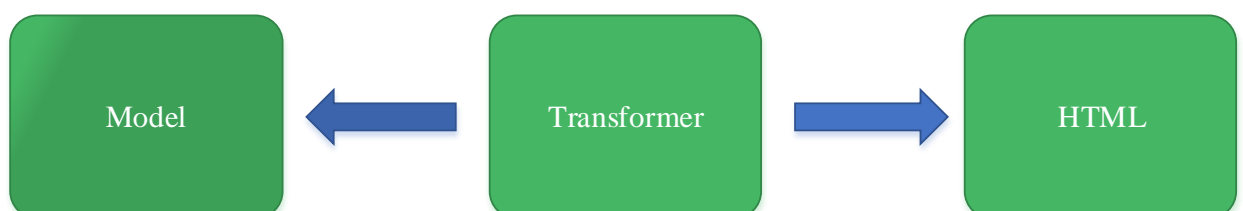


The Controller is essentially the main part of the application, which shows view when it needs to load up and interacts with the appropriate models, it controls the data flow into model object and updates the view whenever data changes. The view is the visible interface that the user interacts with, displaying buttons, forms, and information. Generally speaking, the controller calls up the view after interacting with the model, which is what gathers the information to display in the particular view. The Model is where data from the controller is passed into and manipulated, it is an abstraction of something that you want to present the user in a view.

As I understood, two-step view is a mix of template view and transform view. Therefore I decided to make two-step view which includes:

- 1) Template view - parts of blog that cannot be changed
 - a. css, html templates stored in files;
 - b. header with title and menu
 - c. footer with sidebar and copyright menu

I stored header, footer and sidebar html files as separate template.



- 2) Transform view - involves a view mechanism that processes data structures one element at the time, and transforms them into an end-user representation like HTML.
- a. Content – part of blog that can be changed

Blog This is my blog. It's awesome.

[Main page](#)

[Gallery](#)

[Contact](#)

Blog Post Title

Written by [John Smith](#) on August 12, 2014.

Blogging which started in mid nineties as a personal hobby to share one's thoughts has indeed come a long way. Initially, it was nothing more than an extension of diary writing but in no time, it turned into a powerful medium. Soon, people started using it as a source of secondary income and some were even lucky enough to be full time, professional bloggers. Today, it is used widely by businesses as part of their marketing strategy.

Pork drumstick turkey fugiat. Tri-tip elit turducken pork chop in. Swine short ribs meatball irure bacon nulla pork belly cupidatat meatloaf cow. Nulla corned beef sunt ball tip, qui bresaola enim jowl. Capicola short ribs minim salami nulla nostrud pastrami. Nulla corned beef sunt ball tip, qui bresaola enim jowl. Capicola short ribs minim salami nulla nostrud pastrami.

Blog Post Title

Written by [John Smith](#) on August 13, 2014.

FACT: Blogging can really move the needle. Blog articles influence purchases. Companies that blog have 55% more website visitors. B2C companies that blog generate 88% more leads per month than those who do not. B2B companies that blog generate 67% more leads per month than those who do

not. Companies with >51 blog articles experience a 77% lift in median monthly leads. Blogging frequency has a direct & significant impact on lead-gen.

Categories

[News](#)

[Code](#)

[Design](#)

[Fun](#)

Featured

Pork drumstick turkey fugiat. Tri-tip elit turducken pork chop in. Swine short ribs meatball irure bacon nulla pork belly cupidatat meatloaf cow.

[Read More →](#)

