

```

@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
    Cookie[] cs = req.getCookies(); // null !!!
    // String here = req.getContextPath() + req.getRequestURI() + req.getQueryString();
    Optional.ofNullable(cs)
        .flatMap(cc ->
            Arrays.stream(cc).filter(c -> c.getName().equals("id"))
        ).ifPresentOrElse(
            () -> {
                c -> {
                    String userId = c.getValue();
                    try (PrintWriter w = resp.getWriter()) {
                        log.getAll(userId).forEach(s -> w.println(s));
                    } catch (Exception e) {
                        throw new RuntimeException(e);
                    }
                },
                () -> {
                    System.out.println("HistoryServlet.no cookie");
                    try {
                        resp.sendRedirect(location: "/login"); // "/login" + "?from=" + here
                    } catch (IOException e) {
                        throw new RuntimeException(e);
                    }
                }
            }
        );
}

```

Handwritten notes: "if" (with a checkmark) next to the `ifPresentOrElse` block, and "ELSE" (with a checkmark) next to the `else` block.

```

@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    Cookie[] cs = req.getCookies(); // null !!!
    // String here = req.getContextPath() + req.getRequestURI() + req.getQueryString();
    Optional.ofNullable(cs)
        .flatMap(cc ->
            Arrays.stream(cc).filter(c -> c.getName().equals("id")).findFirst()
        ).ifPresentOrElse(
            () -> {
                //cs.get("check")
                // compare check with check in the DB
                // delete check from db (not to deal with duplicate stuff)
                String userId = c.getValue();
                System.out.println(userId);
                String xs = req.getParameter(name: "x"); // null
                System.out.println(req.getQueryString()); // x=5&y=7
                Map<String, String> allParameters = req.getParameterMap();
                Optional<String> oy = safeGet(req, paramName: "y");
                Optional<Integer> oyi = oy.flatMap(x -> safeToInt(x));
                int y = oyi.orElse(0);
                int x = Integer.parseInt(xs);
                int z = x + y;

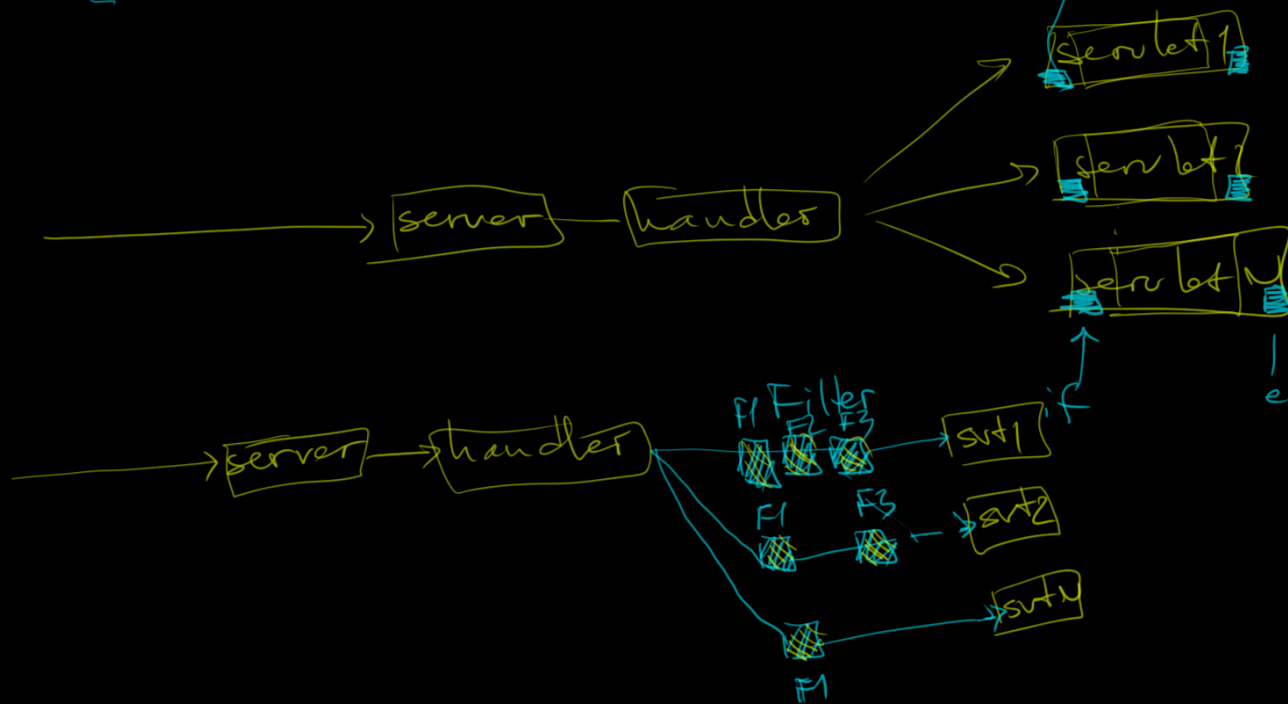
                try (PrintWriter w = resp.getWriter()) {
                    log.store(userId, x, y, z);
                    w.printf("%d + %d = %d", x, y, z);
                } catch (Exception e) {
                    throw new RuntimeException(e);
                }
            },
            () -> {
                System.out.println("CalcServlet.no cookie");
                try {
                    resp.sendRedirect(location: "/login"); // "/login" + "?from=" + here
                } catch (IOException e) {
                    throw new RuntimeException(e);
                }
            }
        );
}

```

Handwritten notes: "if" (with a checkmark) next to the `ifPresentOrElse` block, and "ELSE" (with a checkmark) next to the `else` block.

$1+1 \rightarrow /calc$

$\leftarrow 1+1=2 + cookie$



```
Cookie[] cs = req.getCookies(); // null !!!
String here = req.getContextPath() + req.getRequestURI() +
Optional.ofNullable(cs)
    .flatMap(cc ->
        Arrays.stream(cc).filter(c -> c.getName().equals("
    ) ifPresentOrElse(
```

```
-> {
    System.out.println("CalcServlet: no cookie");
    try {
        resp.sendRedirect(location: "/login"); // "/"
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

Servlet

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp)
```

Filter

doFilter (Request, Response, next)

if (r.equals("...")) next(r, rs)

```
else {  
    rs.sendRedirect(...)  
    rs.setStatus(...)  
    rs.addCookie(...)  
}
```

SQL V001_table_students.sql

SQL V002_initial_data.sql

V999_..._sql

V2022-12-07 -- ...

Sortable

001 2022
002 2022

~~create~~

drop index
insert 1000
create index