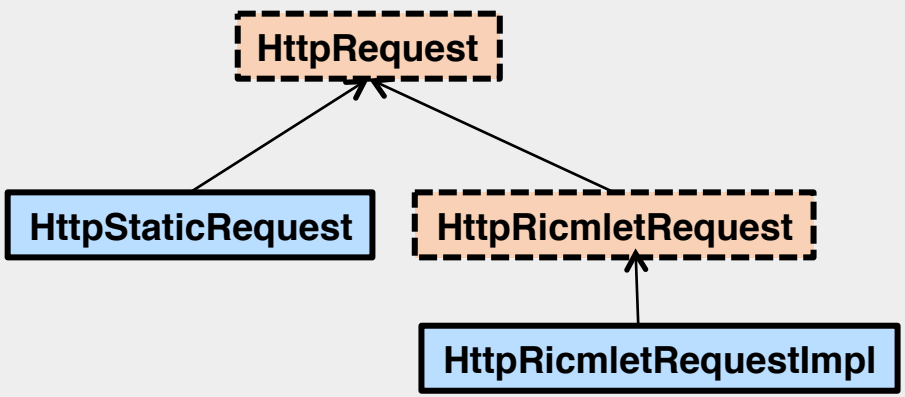
Distributed Systems
Project

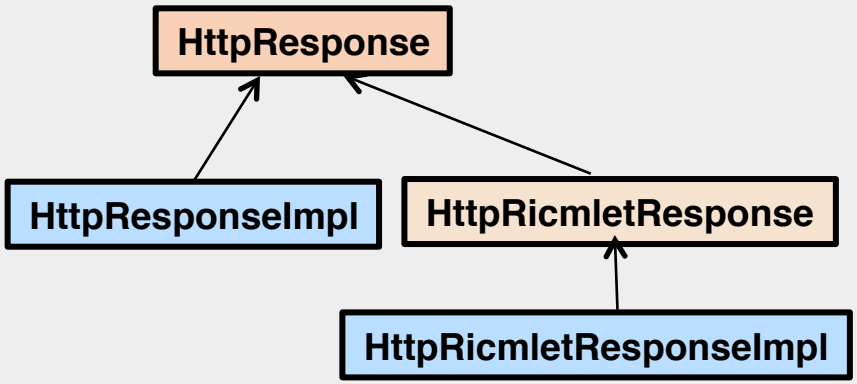
Polytech/INFO 4, 2022-2023
Fabienne Boyer, Olivier Gruber,
UFR IM2AG, LIG, Université Grenoble Alpes
Fabienne.Boyer@imag.fr

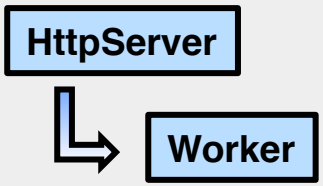
Designing & programming your own HTTP Server

- Binôme
- 2 sessions with teachers + on your own
- Soutenances on 05/04/23

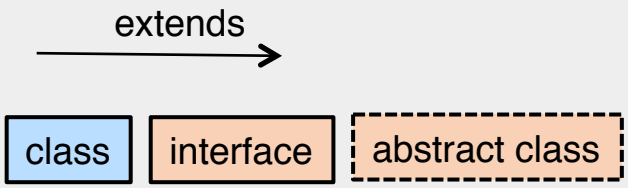
Server Classes



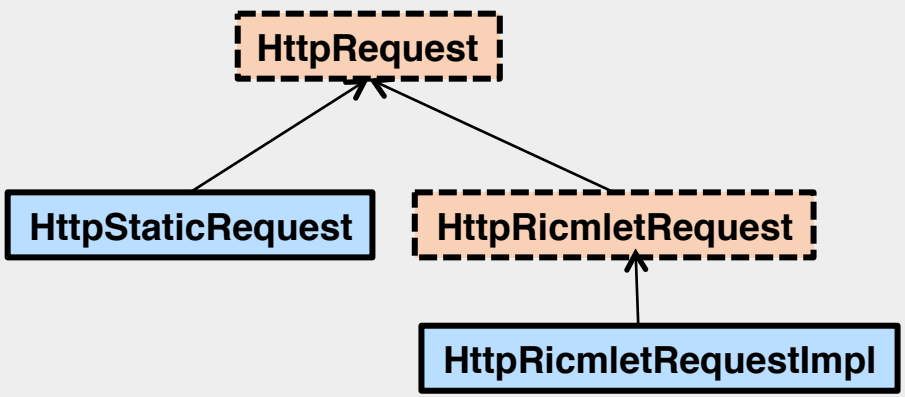


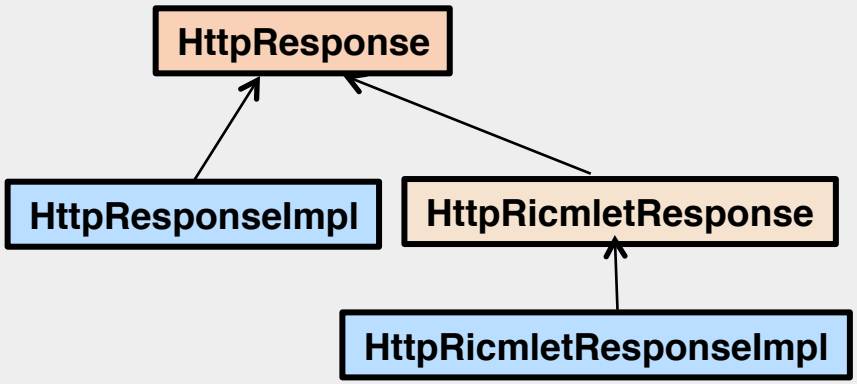


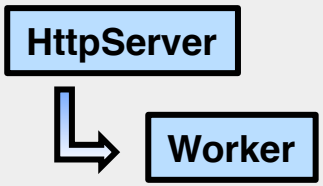
Get socket input & output streams is & os
Build HttpRequest / is
Build HttpResponse / os
Process the request

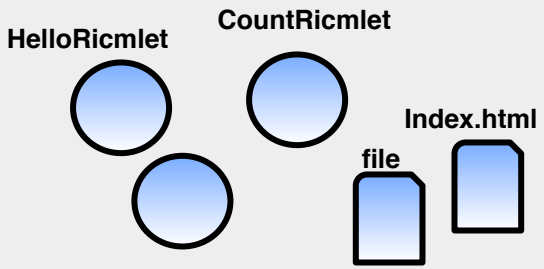


Server Classes

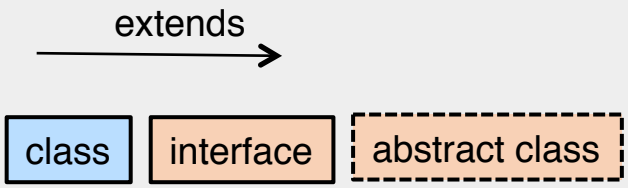




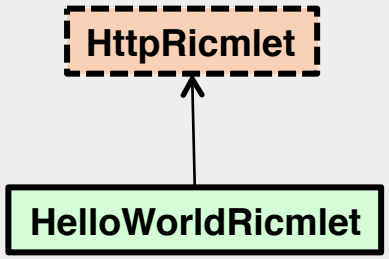


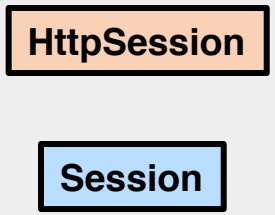


Get socket input & output streams is & os
Build HttpRequest / is
Build HttpResponse / os
Process the request



Applicative Classes (used by the Web programmer)






```
public class HelloRicmlet implements httpserver.itf.HttpRicmlet{

   public void doGet(HttpRicmletRequest req,  HttpRicmletResponse resp) throws IOException {
    resp.setReplyOk();
    resp.setContentType("text/html");
    PrintStream ps = resp.beginBody();
    ps.println("<HTML><HEAD><TITLE>…
    …
}
```

Designing & programming your own HTTP Server

- **Remote communication on top of TCP/IP**
- **Access to remote resources**
  - Files
  - Code (execution of ricmlets)
- **Manage session data**
  - Cookies
  - Session
- **Manage applications**
  - Class loading aspects
  - Classically, *a class loader* is created *for each* web *application* that is deployed in *a single* HTTP server instance
  - Lecture on this aspect next Friday