



FPT POLYTECHNIC



Conceive Design Implement Operate

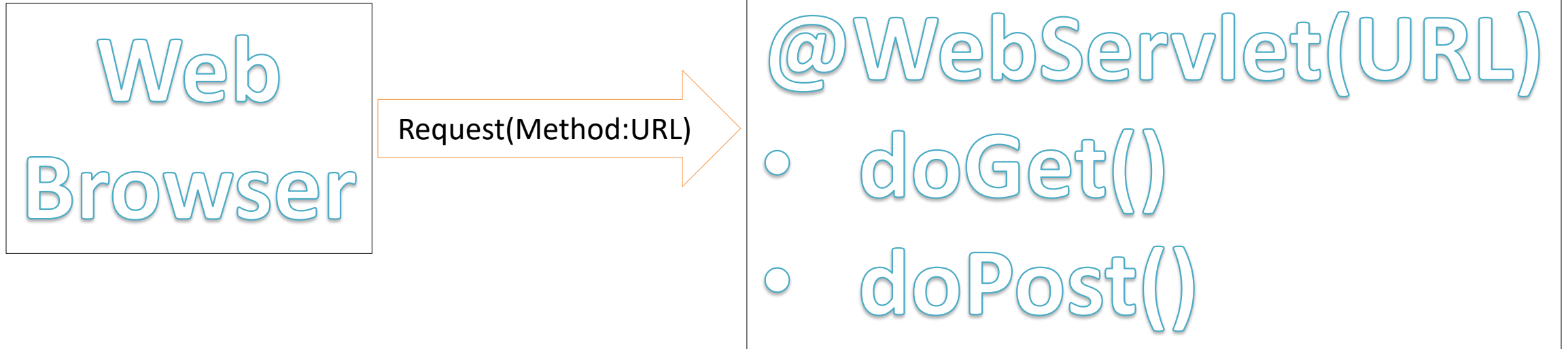
XỬ LÝ FORMS, PHÂN BIỆT GET/POST

LẬP TRÌNH JAVA #3 (P4.1)

www.poly.edu.vn

- ❑ Giới thiệu các phương thức gửi dữ liệu POST, GET
- ❑ Tổ chức servlet xử lý yêu request với phương thức POST, GET
- ❑ Xử lý form chứa nhiều nút submit





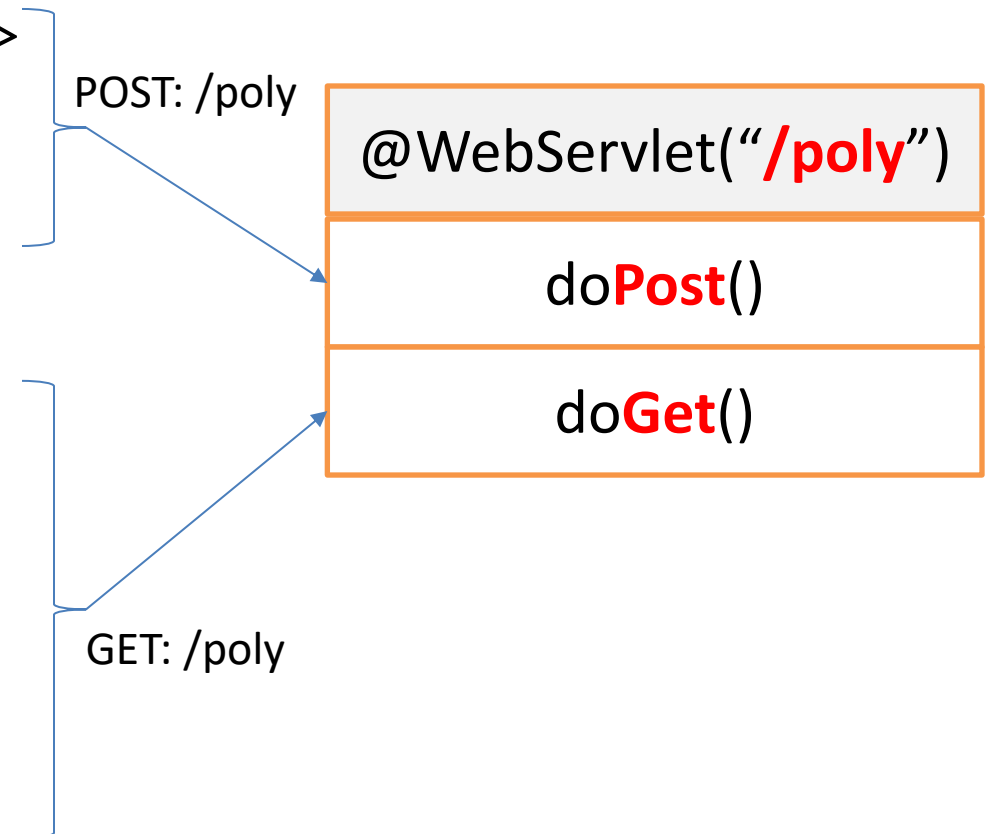
HÌNH THỨC KHẢN CẦU	METHOD	REQUEST	@WEBSERVLET(URL)
Nhập URL vào ô địa chỉ trình duyệt	GET	(GET:URL)	doGet()
Nhấp link có @href=URL			
Submit form có @action=URL và @method=GET			
Submit form có @action=URL và @method=POST	POST	(POST:URL)	doPost()

❑ doPost()

- ❖ `<form action=".../poly" method="post">`
`<button>Submit</button>`
- ❖ `</form>`

❑ doGet()

- ❖ Gõ vào ô địa chỉ của trình duyệt
- ❖ `Liên kết`
- ❖ `<form action=".../poly" method="get">`
`<button>Submit</button>`
- ❖ `</form>`
- ❖ ...



```
@WebServlet("URL")
public class PolyServlet extends HttpServlet{
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        //... code xử lý GET:URL ...//
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        //... code xử lý POST:URL...//
    }
}
```

```
<form action="URL" method="get">
    <!-- GET:URL => doGet() -->
    <button>OK</button>
</form>

<form action="URL" method="post">
    <!-- POST:URL => doPost() -->
    <button>OK</button>
</form>
```



- ❑ Form được sử dụng để lấy dữ liệu từ người sử dụng hoặc hiển thị dữ liệu cho người sử dụng.
- ❑ Thông thường để làm việc với form cần thực hiện 2 request
 - ❖ GET request (link) để hiển thị form (gọi doGet())
 - ❖ POST request (submit form) để xử lý dữ liệu nhập vào form (gọi doPost())

Username?
Password?
<input type="button" value="Login"/>

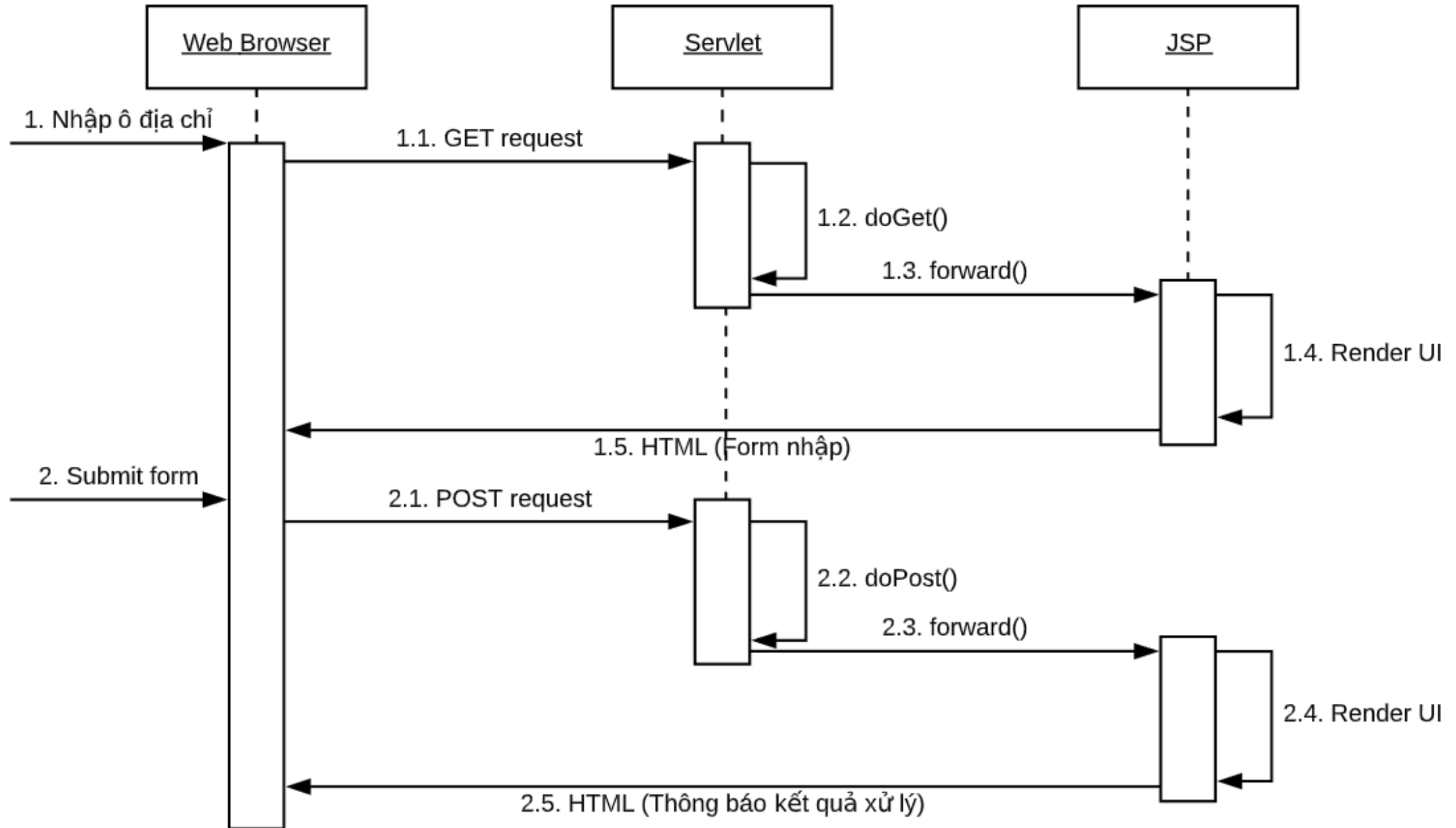
TỔ CHỨC SERVLET XỬ LÝ FORM

GET: http://.../login/form

```
@WebServlet({"login/form", "login/check"})
public class LoginServlet extends HttpServlet{
    //GET: /login/form
    public void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException , IOException {
        req.getRequestDispatcher("/views/login.jsp").forward(req, resp);
    }
    // POST: /login/check
    public void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException , IOException {
        // TODO: check existence of user here
        req.getRequestDispatcher("/views/login.jsp").forward(req, resp);
    }
}
```

POST: http://.../login/check

```
<form action=".../login/check" method="post">
    <button>Login</button>
</form>
```

- ❑ POST và GET là 2 phương pháp truyền dữ liệu từ trình duyệt đến server.
- ❑ Khi submit form với
 - ❖ Method="GET" thì dữ liệu form được thu thập và tạo thành chuỗi truy vấn (ghép vào sau dấu ? của url) để gửi đến server
 - ❖ Method="POST" thì trình duyệt sẽ tạo một kênh riêng để truyền dữ liệu đến server
- ❑ Ưu và nhược điểm của POST và GET
 - ❖ GET nhanh hơn POST nhưng có một số hạn chế như:
 - Dữ liệu nhỏ
 - Chỉ cho phép text (không upload được)
 - Phơi bày dữ liệu tế nhị (password và trường ẩn)
 - ❖ POST khắc phục được những hạn chế của GET

```
<form method="post">  
  <!-- POST:URL1 => doPost() -->  
  <button formaction="URL1">OK</button>  
  <!-- POST:URL2 => doPost() -->  
  <button formaction="URL2">Cancel</button>  
</form>
```

```
<form>  
  <!-- POST:URL1 => doPost() -->  
  <button formaction="URL1" formmethod="post">OK</button>  
  <!-- GET:URL2 => doGet() -->  
  <button formaction="URL2" formmethod="get">Cancel</button>  
</form>
```

```
@WebServlet({ "URL1", "URL2" })
public class PolyServlet extends HttpServlet{
    @Override
    protected void service(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        String method = req.getMethod();
        String path = req.getServletPath();
        //... code xử lý theo method:path ...//
    }
}
```

- ❑ Sử dụng if...else if...else hoặc switch case để phân biệt theo các cặp method:path để thực hiện những đoạn mã theo yêu cầu của nghiệp vụ



- ☑ Giới thiệu các phương thức gửi dữ liệu POST, GET
- ☑ Tổ chức servlet xử lý yêu request với phương thức POST, GET
- ☑ Xử lý form chứa nhiều nút submit





Cảm ơn