Web-based programming with Java

SERVLET & JSP

ThS. Nguyễn Nghiệm
0913.745.789 - NghiemN@fpt.edu.vn

Mục tiêu

- Kết thúc bài học này bạn có khả năng
  - Xây dựng web hoạt động theo mô hình MVC
  - Giải thích vòng đời servlet và jsp
  - Sử dụng filter để giám sát hoạt động
  - Sử dụng listener để xử lý các biến cố
  - Chia sẽ dữ liệu giữa các thành phần
  - Đọc ghi được cookie

NỘI DUNG

- Tổ chức ứng dụng theo mô hình MVC
- Vòng đời Servlet
- Vòng đời JSP
- Bộ lọc – Filter
- Bộ nghe – Listener
- Chia sẽ dữ liệu
- Cookie

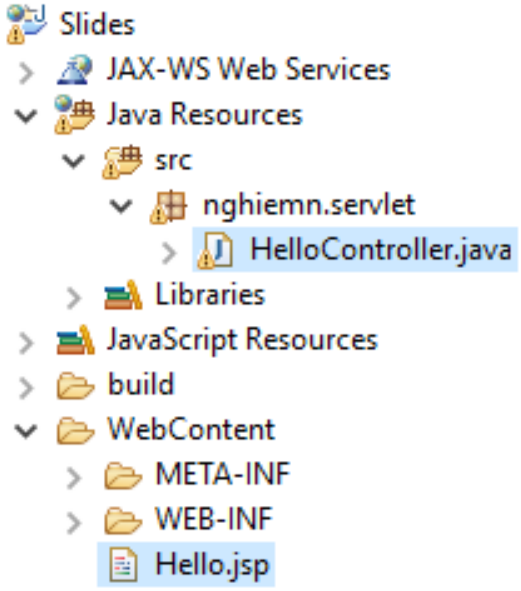
MÔ HÌNH MVC

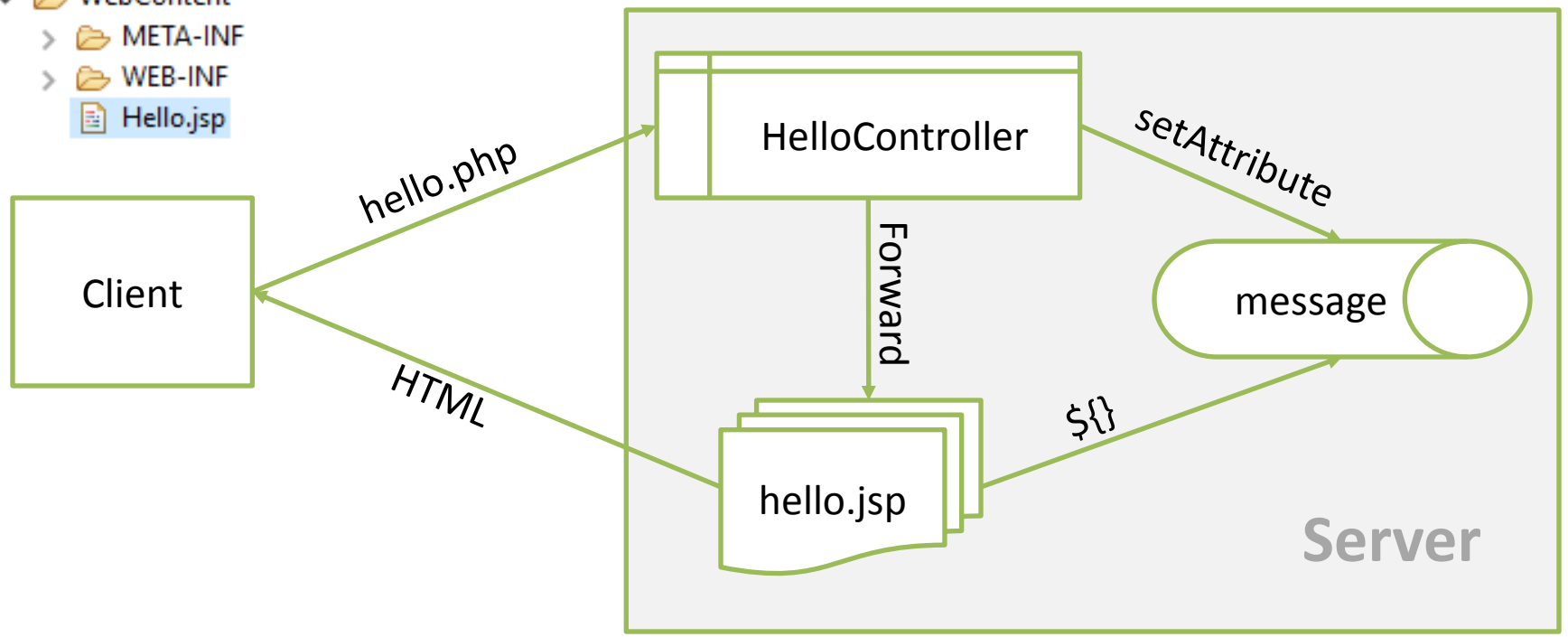
```java
@WebServlet("/hello.do")
public class HelloController extends HttpServlet{
	@Override
	protected void service(HttpServletRequest req, HttpServletResponse resp)
			throws ServletException, IOException {
		req.setAttribute("message", "Hello Servlet & JSP");
		req.getRequestDispatcher("Hello.jsp").forward(req, resp);
	}
}
```

- Controller: Servlet
- View: JSP
- Model: Request

```jsp
<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
	<meta charset="utf-8">
	<title>Insert title here</title>
</head>
<body>
	${message}
</body>
</html>
```

MÔ HÌNH MVC

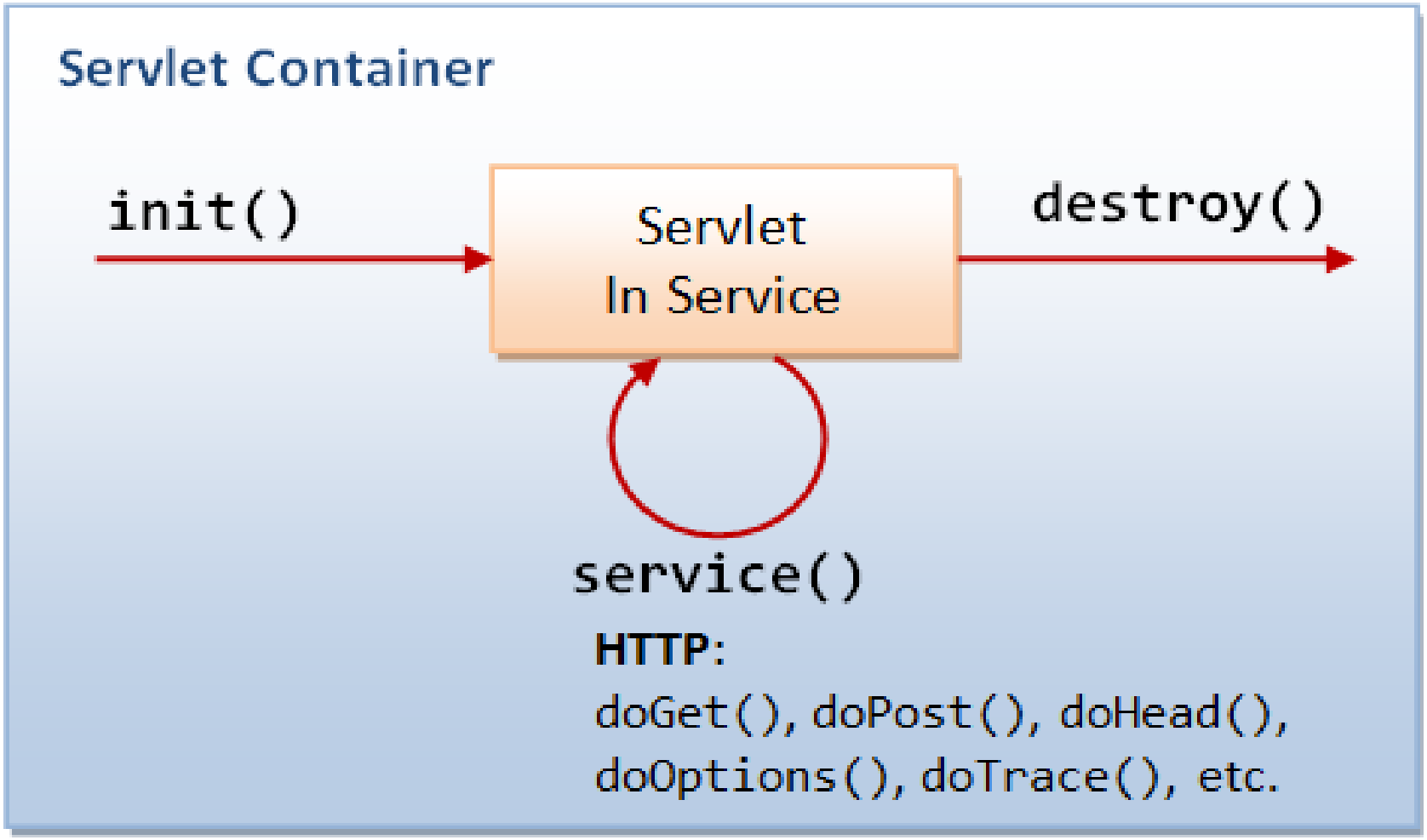

Slides
JAX-WS Web Services
Java Resources
src
nghiemn.servlet
HelloController.java
Libraries
JavaScript Resources
build
WebContent
META-INF
WEB-INF
Hello.jsp


Client
hello.php
HTML
HelloController
setAttribute
Forward
message
${}
hello.jsp
Server

Tổ chức theo mô hình MVC

ĐỀ MÔ

VÒNG ĐỜI SERVLET



VÒNG ĐỜI SERVLET

```
@WebServlet("/my-servlet.php")
public class MyServlet extends HttpServlet{
    @Override
    protected void service(HttpServletRequest req,
            HttpServletResponse resp) throws ServletException, IOException {
    }

    @Override
    public void init(ServletConfig config) throws ServletException {
    }

    @Override
    public void destroy() {
    }
}
```

@WebServlet
extends HttpServlet
init()
service()/doPost()/doGet()
destroy()

Log các sự kiện

ĐỀ MÔ

PHÂN BIỆT POST, GET

- Tạo request với POST
  - <form action="hello.php" method="POST">
  - <button formmethod="POST">
- Tạo request với GET
  - My Servlet
  - <form action="hello.php" method="GET">
  - <form action="hello.php">
  - <button formmethod="GET">
  - ...

HttpServletRequest

- Nhận tham số
  - String getParameter(String)
- Truyền dữ liệu cho jsp
  - setAttribute(String, Object)
- Chuyển tiếp/bao hàm sang jsp
  - getRequestDispatcher(“*.jsp”). forward(req, resp)
  - getRequestDispatcher(“*.jsp”). include(req, resp)
- Một số thông tin khác
  - getRequestURI()/getRequestURL()/getContextPath()

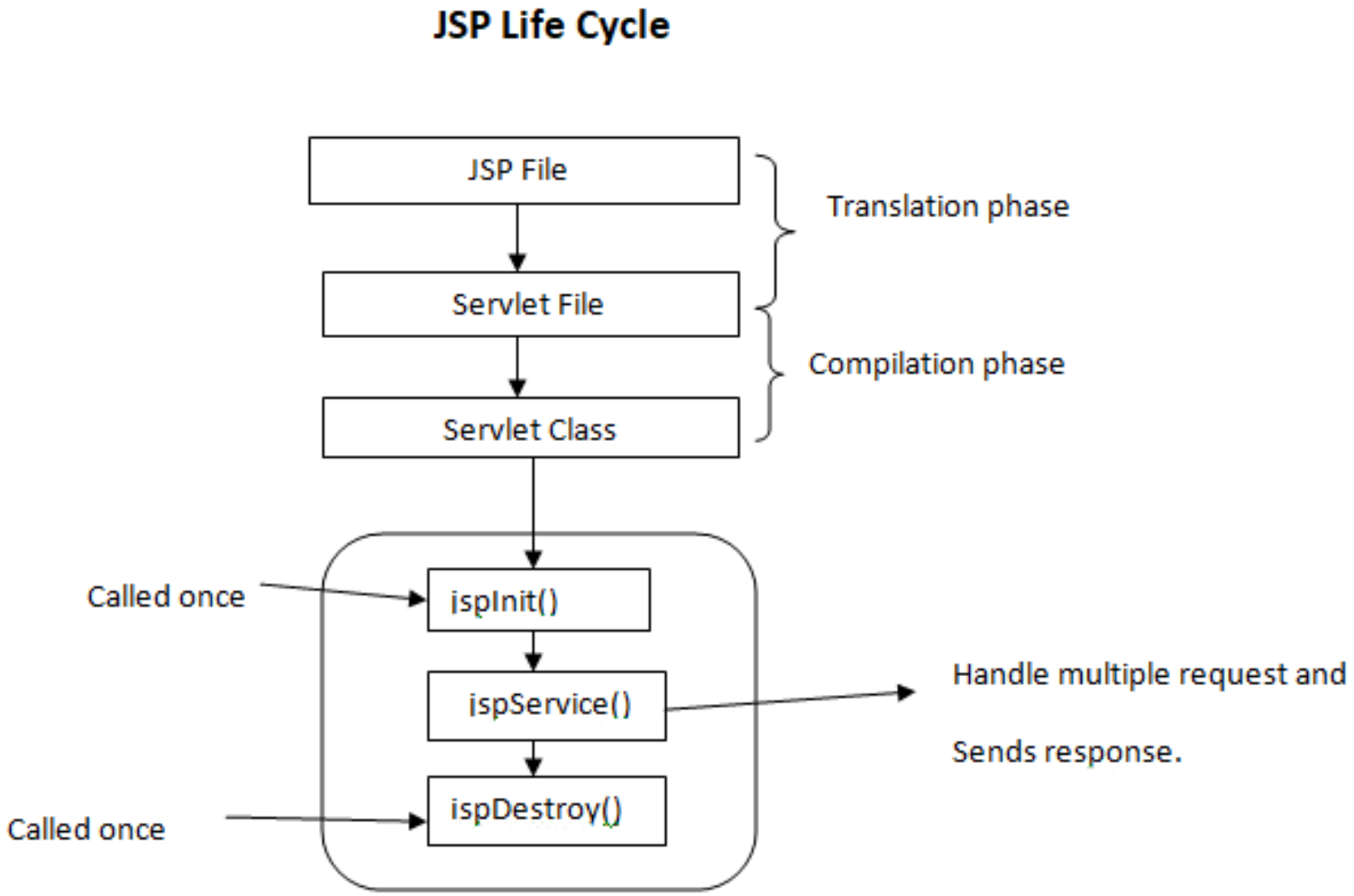
MÃ JSP THƯỜNG DÙNG

- Chỉ thị trang jsp
  - <%@ page pageEncoding="utf-8"%>
- Truy xuất attribute
  - ${message}
- Bao hàm một trang jsp khác
  - <jsp:forward page="x.jsp"/>
  - <jsp:include page="x.jsp"/>
- Truy xuất thông tin của request
  - ${pageContext.request.contextPath}
  - ${pageContext.request.requestURI}
  - ${pageContext.request.requestURL}

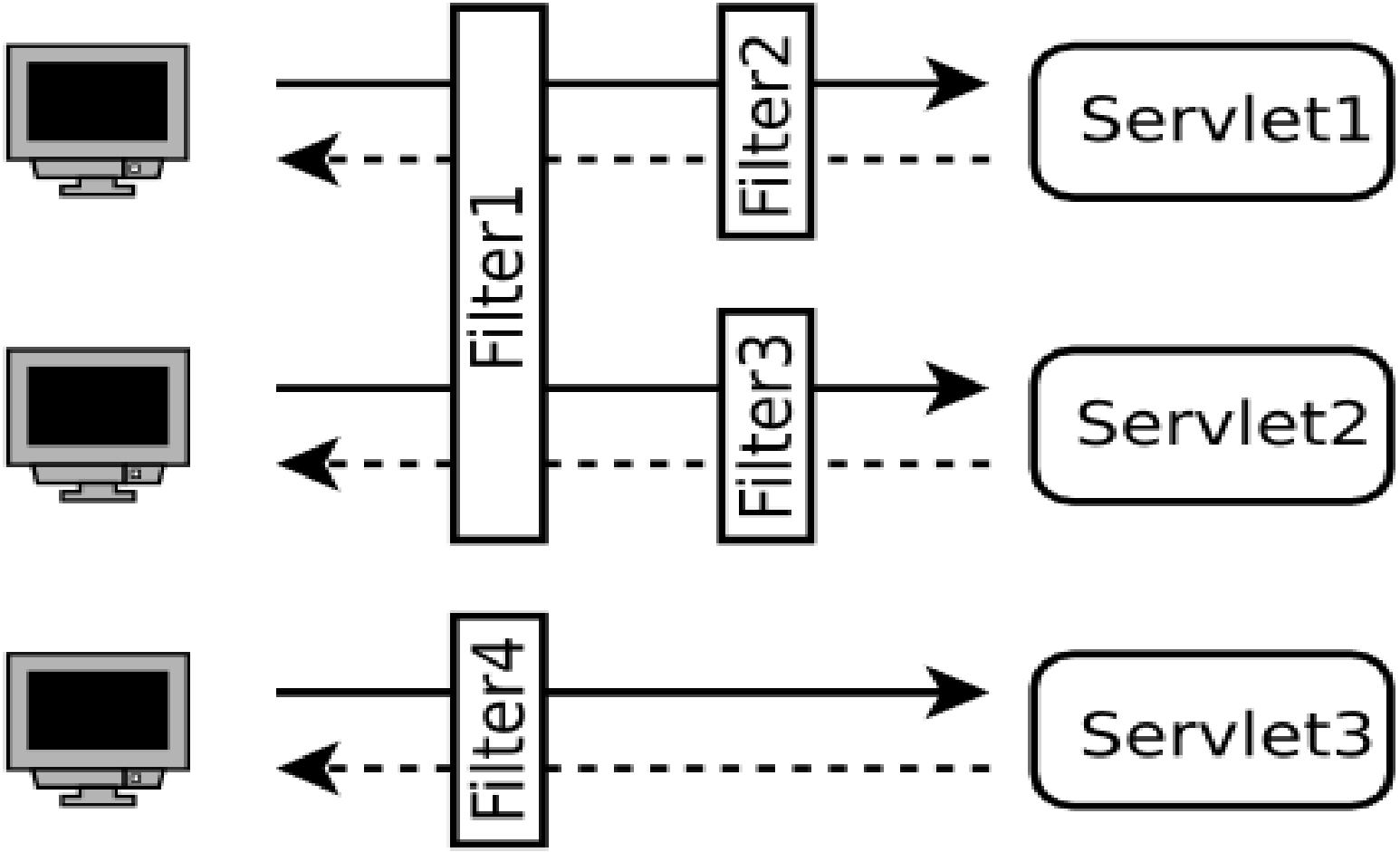
HttpServletRequest

ĐỀ MÔ

VÒNG ĐỜI JSP


JSP Life Cycle
JSP File
Servlet File
Servlet Class
Translation phase
Compilation phase
Called once
jspInit()
jspService()
jspDestroy()
Called once
Handle multiple request and
Sends response.

Bộ lọc (Filter)



HOẠT ĐỘNG CỦA FILTER

```
public class MyFilter implements Filter{

	@Override
	public void destroy() {
		System.out.println(" >> MyFilter.destroy()");
	}

	@Override
	public void doFilter(ServletRequest req, ServletResponse resp,
			FilterChain chain) throws IOException, ServletException {
		System.out.println(" >> MyFilter.doFilter()-before");
		chain.doFilter(req, resp);
		System.out.println(" >> MyFilter.doFilter()-after");
	}

	@Override
	public void init(FilterConfig config) throws ServletException {
		System.out.println(" >> MyFilter.init()");
	}
}
```

CharacterEncodingFilter và LoggerFilter

ĐỀ MÔ

LISTENER

- Listener được sử dụng để lắng nghe các sự kiện xảy ra bên trong ứng dụng web
- Interface ServletContextListener chứa 2 sự kiện về ứng dụng
  - contextInitialized()
  - contextDestroyed()
- Interface HttpSessionListener chứa các sự kiện về phiên làm việc
  - sessionCreated()
  - sessionDestroy()

HttpSessionListener

```
@WebListener
public class AppListener implements HttpSessionListener{

    @Override
    public void sessionCreated(HttpSessionEvent arg0) {
    }

    @Override
    public void sessionDestroyed(HttpSessionEvent arg0) {
    }
}
```

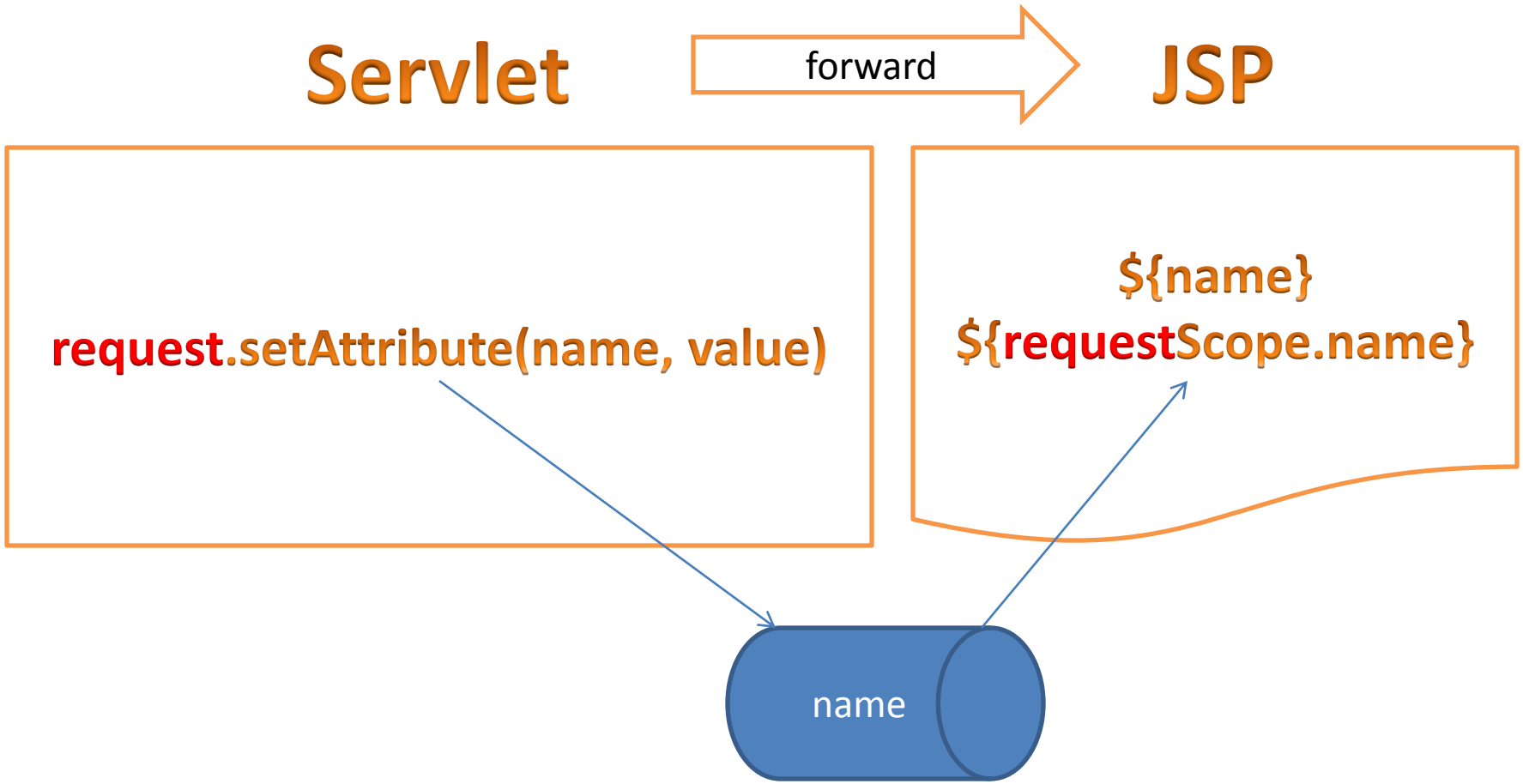
SERVLETCONTEXTLISTENER

```
@WebListener
public class AppListener implements ServletContextListener{

    @Override
    public void contextDestroyed(ServletContextEvent arg0) {
    }

    @Override
    public void contextInitialized(ServletContextEvent arg0) {
    }
}
```

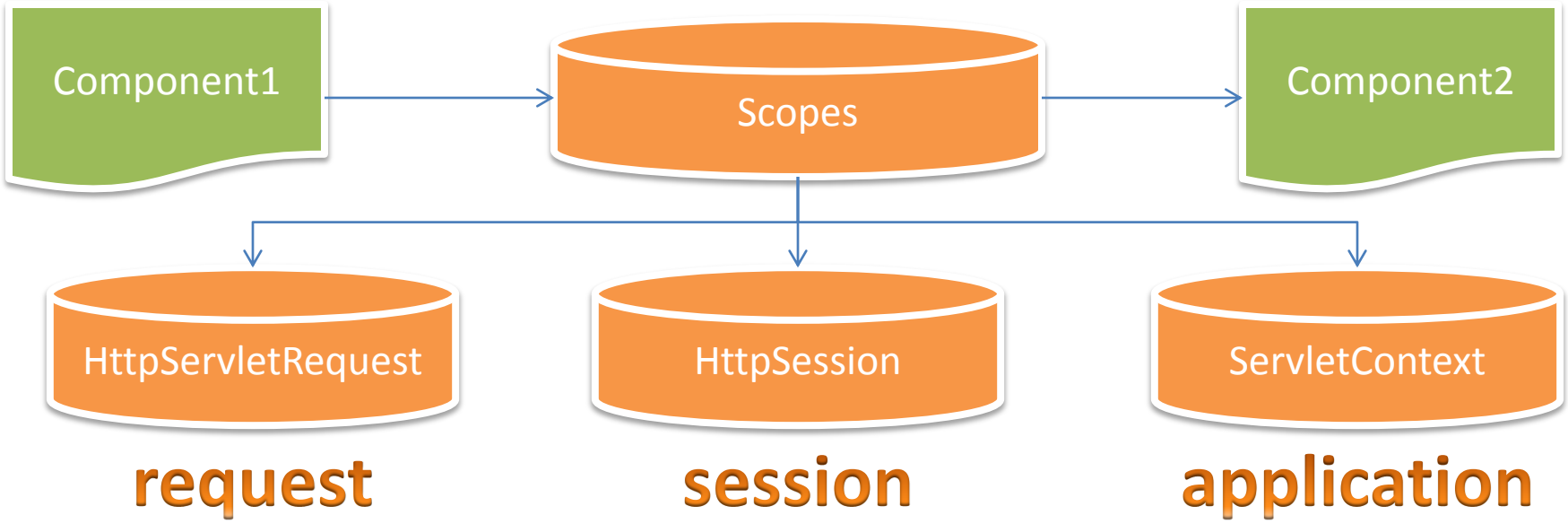
Logging Events

ĐỀ MÔ

CHIA SẺ DỮ LIỆU



CHIA SẺ DỮ LIỆU



- Component 1 và Component 2
  - Cùng hoạt động trên 1 yêu cầu (**request**)
  - Cùng hoạt động trên 1 phiên (**session**)
  - Cung hoạt động trên 1 ứng dụng (**application**)

SCOPE API

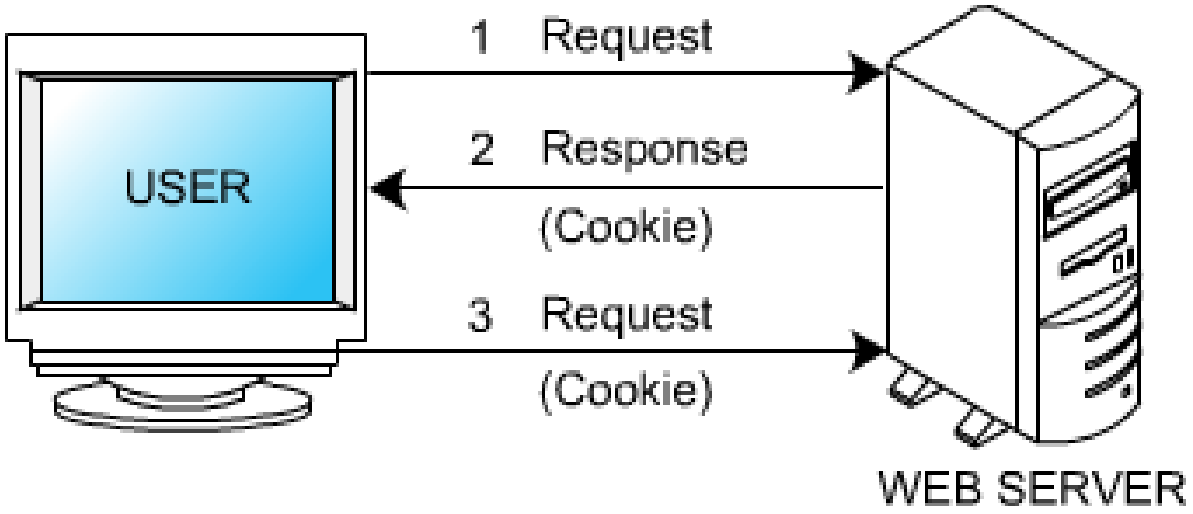
- Tham chiếu Scope
  - HttpServletRequest là đối số của service(), doPost() và doGet
  - HttpSession scope= request.getSession()
  - ServletContext scope = getServletContext()
- Thao tác Scope
  - scope.setAttribute(String name, Object value)
  - Object scope.getAttribute(String name)
  - scope.removeAttribute(String name)

Visitors, User

ĐỀ MÔ

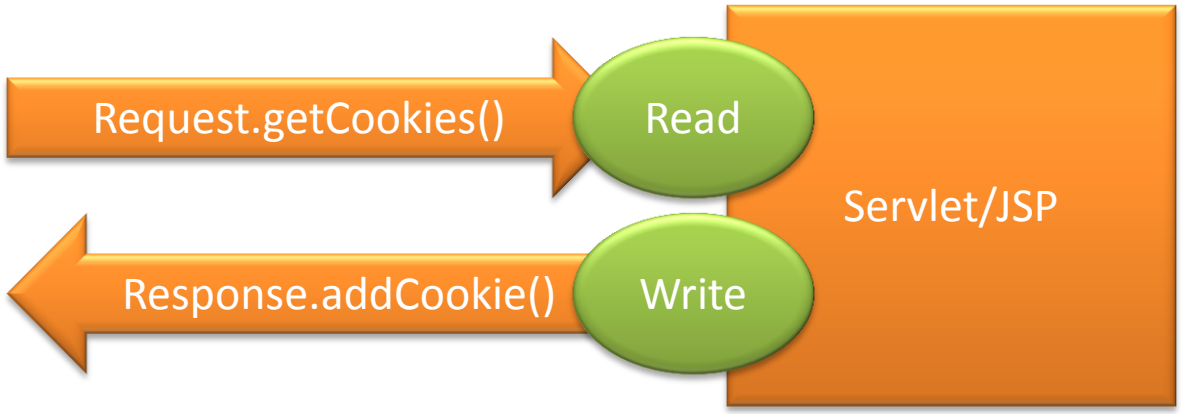
COOKIE

- Cookie là mẫu tin văn bản nhỏ được lưu trên máy client và chỉ tồn tại trong một khoảng thời gian nhất định. Nó được gửi lên server mỗi request.



LÀM VIỆC VỚI COOKIE

- Các thuộc tính cookie
  - Name: tên
  - Value: giá trị
  - MaxAge: thời hạn (mili giây)
  - Path: đường dẫn hiệu lực



LÀM VIỆC VỚI COOKIE

```
String name = "MyCookie";
String value = "My Cookie Value";
int maxAge = 60*60*24*365; // 1 năm
Cookie newCookie = new Cookie(name, value);
newCookie.setMaxAge(maxAge);
response.addCookie(newCookie);
```

Tạo, thiết lập thời gian tồn tại và gửi cookie về client để lưu lại

```
Cookie[] cookies = request.getCookies();
for(int i=0; cookies != null && i<cookies.length; i++)
{
      Cookie cookie = cookies[i];
      String name = cookie.getName();
      String value = cookie.getValue();
}
```

Đọc các cookie được gửi lên server theo request

Tạo, đọc và xóa

ĐỀ MÔ

TÓM TẮT

- Tổ chức ứng dụng theo mô hình MVC
- Vòng đời Servlet
- Vòng đời JSP
- Bộ lọc – Filter
- Bộ nghe – Listener
- Chia sẽ dữ liệu
- Cookie