



CST 243-3 Rapid Application Development

Lesson 04: Web Technologies for Rapid Development

By Subhash Ariyadasa

Part II

Java Server Pages (JSP)



From Java Servlets to JSPs

- Java Servlets are good to create dynamic web pages
- But surely, it's a pain to use those `println()` statements to generate HTML
- Also hard to maintain as well
- But there is a way to include Java codes inside an HTML, to create a dynamic content
- That is **Java Server Pages (JSPs)**

Java Server Page (JSP)

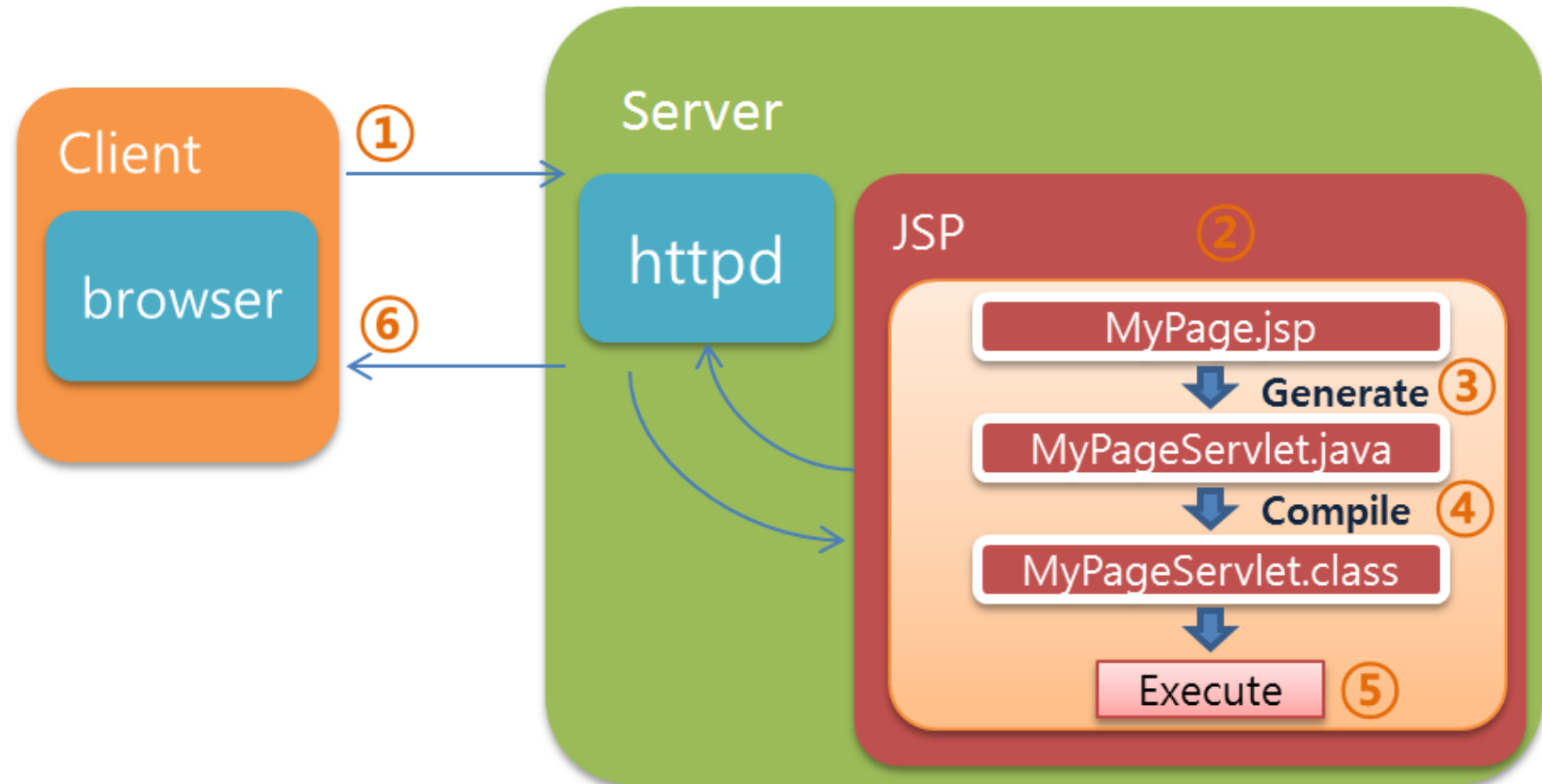
- A technology for developing web pages that support dynamic content which helps developers insert Java code in HTML pages
 - Use special JSP tags
 - Most of which start with `<%` and end with `%>`
 - e.g: `<% Java code goes here... %>`
- JSP is similar to PHP and ASP, but it uses the Java programming language

Java Server Page (JSP)...

- Using JSP,
 - You can collect input from users through web page forms
 - Present records from a database or another source
 - Create web pages dynamically
- Compare to pure servlets, it is more convenient to write and to modify regular HTML than to have plenty of println statements that generate the HTML
- JSP file saves with .jsp file extension

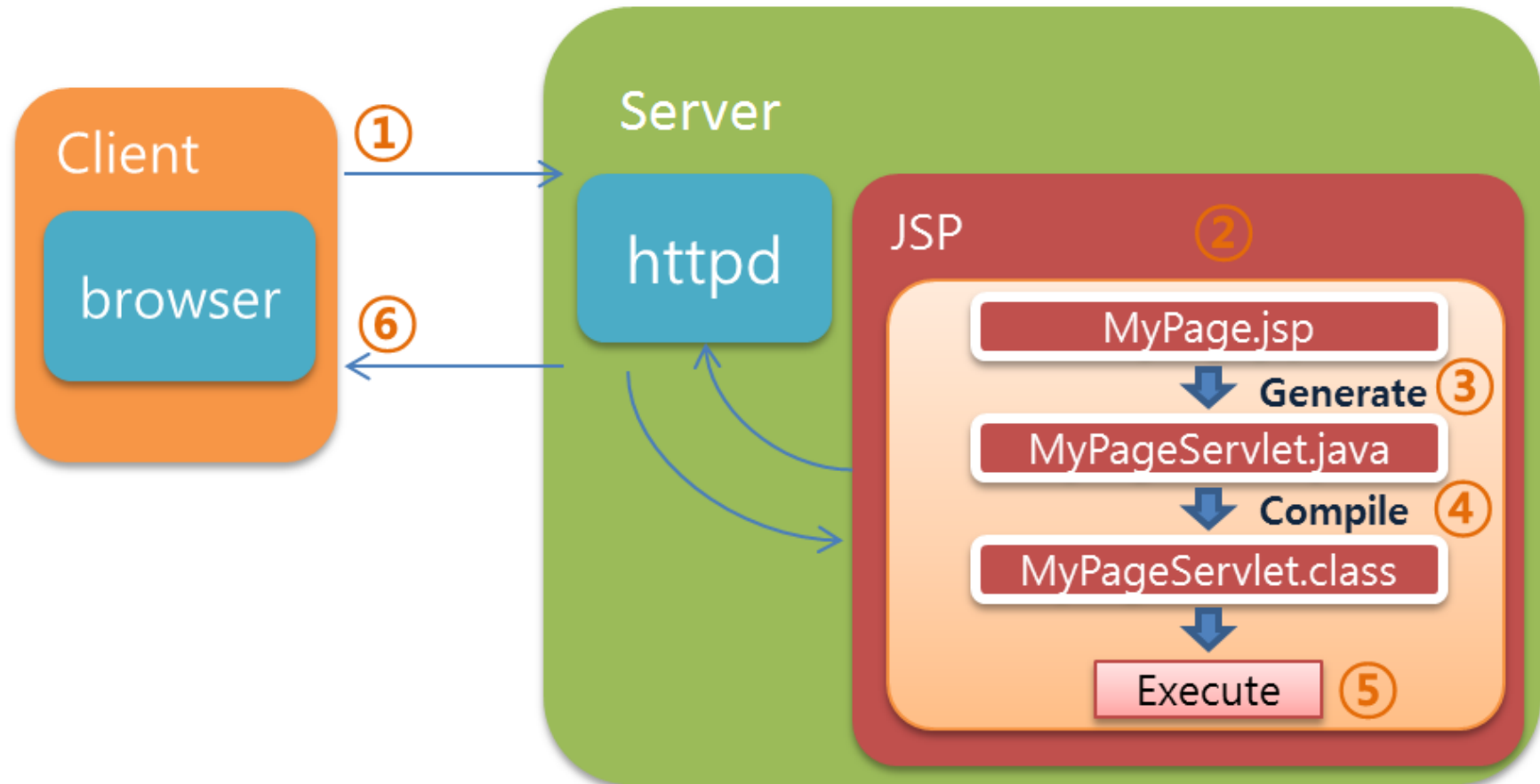
Processing a JSP File

- ① Browser sends an HTTP request to the web server



Processing a JSP File...

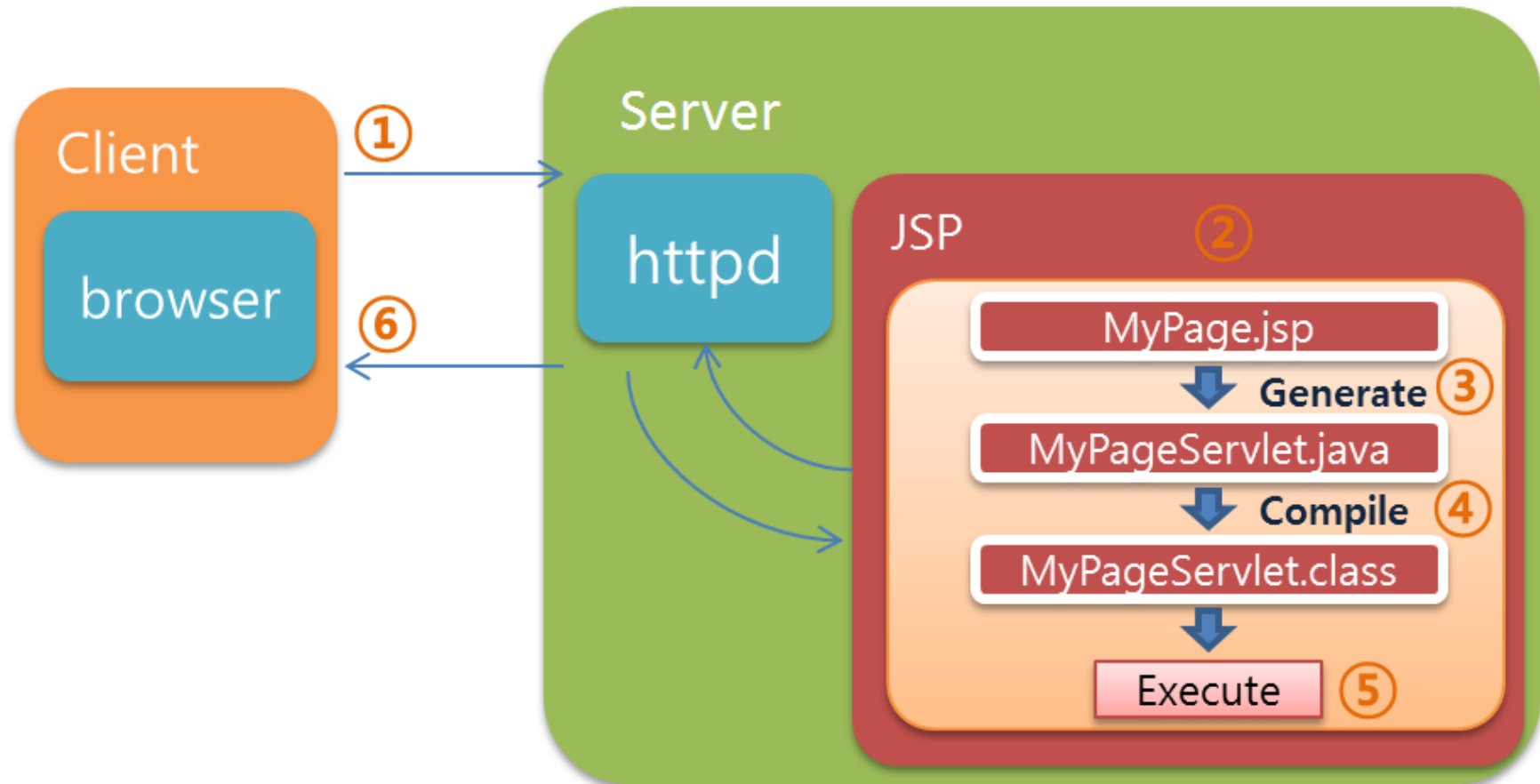
- ② Web server recognizes that the HTTP request is for a JSP page and forwards it to a JSP engine
- Web server will use file extension



Processing a JSP File...

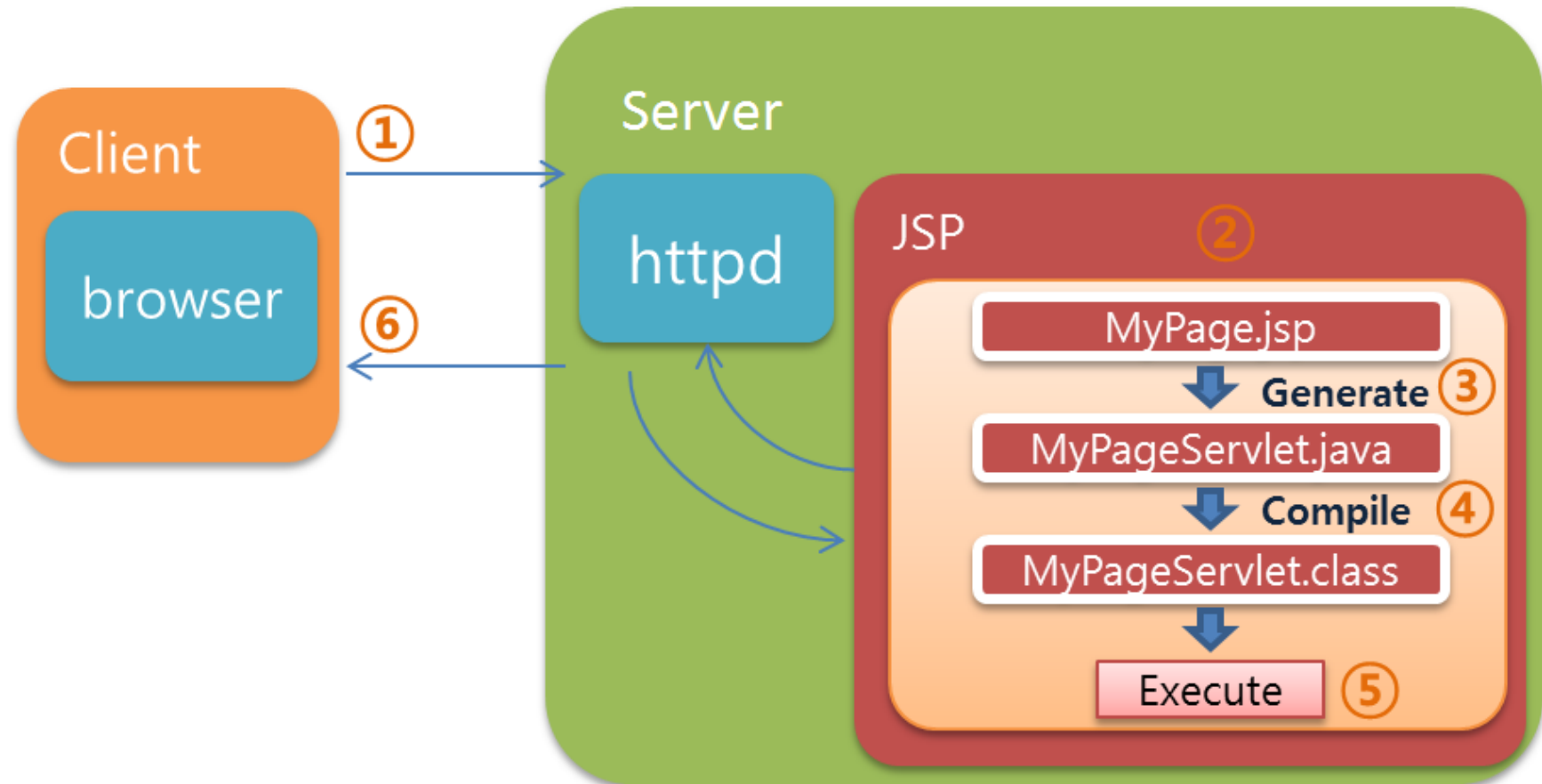
③ JSP engine loads the JSP page from disk and converts it into a servlet content

- All template text is converted to `println()` statements
- All JSP elements are converted to Java code



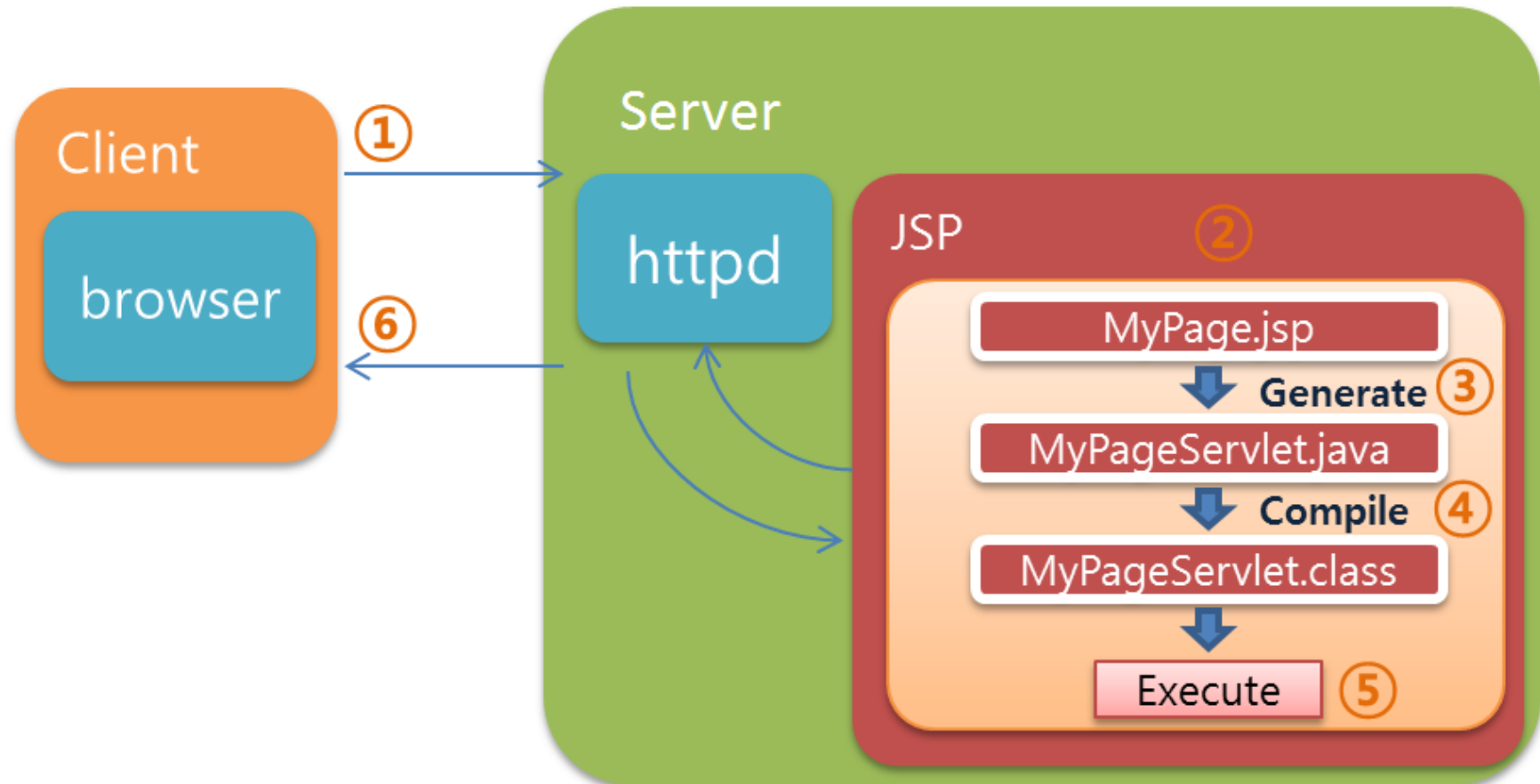
Processing a JSP File...

- ④ JSP engine compiles the servlet into an executable class and forwards to a servlet engine



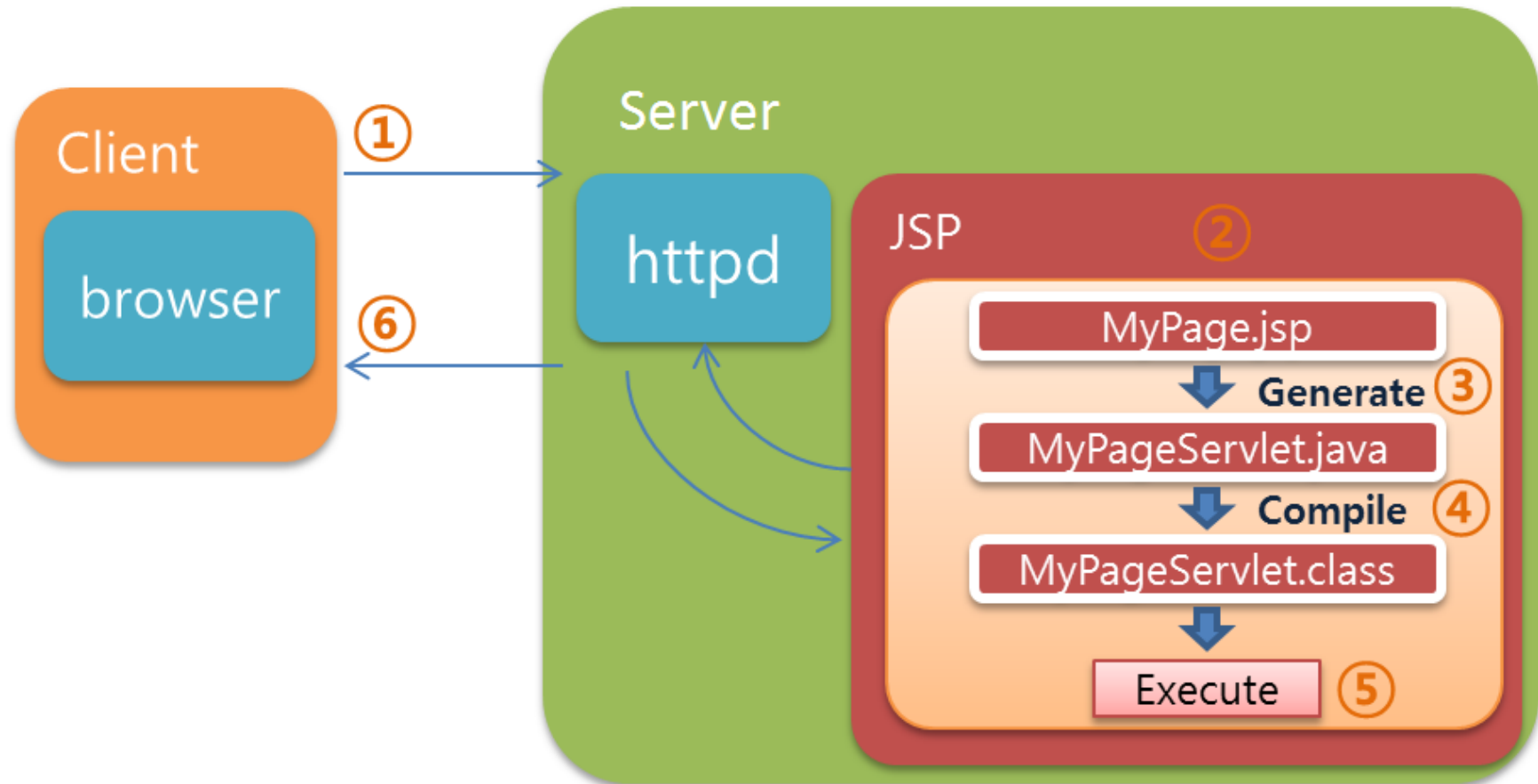
Processing a JSP File...

- ⑤ Servlet engine loads the Servlet class and executes it
- Produces an output in HTML format



Processing a JSP File...

- ⑥ Web server forwards the HTTP response to the browser



JSP Life Cycle

- Translation Phase → JSP page converts into Servlet content (.java extension)
- Compilation Phase → The translated .java file is compiled into the Java servlet .class file
- Initialization Phase → jsplnit() method for initializing the servlet instance
- Execution Phase → invoke the _jspService() method
- Destruction(Cleanup) Phase → jspDestroy() method is invoked

JSP Syntax

- Scriptlet
 - A scriptlet can contain any number of,
 - JAVA language **statements**
 - **Variable** or **method** declarations
 - **Expressions**
 - Syntax:
 - **<% code part goes here %>**
 - Example:
 - **<% out.println("hello world"); %>**

JSP Syntax...

- Example code for a Scriptlet syntax:

```
<html>
```

```
<head><title> Scriptlet Syntax Example </title></head>
```

```
<body>
```

```
    Hello World! <br/>
```

```
        <%out.println("Your IP address is" +  
request.getRemoteAddr());%>
```

```
</body>
```

```
</html>
```

JSP Syntax...

- JSP Declarations
 - A declaration declares **one or more variables or methods** that you can use in Java code later in the JSP file
 - Syntax:
 - `<%! Declaration; %>`
 - Example:
 - `<%! int i = 0; %>`
 - `<%! int a, b, c; %>`

JSP Syntax...

- JSP Expression
 - A JSP expression element contains a scripting language expression that is evaluated, converted to a String, and inserted where the expression appears in the JSP file
 - Syntax:
 - `<%= expression %>`
 - Example:
 - `<%= (new java.util.Date()).toLocaleString() %>`

JSP Syntax...

- Example code for a Expression syntax:

```
<html>
```

```
<head><title> Expression Syntax Example </title></head>
```

```
<body>
```

```
<%out.println("The expression number is "); %>
```

```
<%! int num1=10; int num2=5; %>
```

```
<%= num1*num2; %>
```

```
</body>
```

```
</html>
```

The expression number is 50

JSP Syntax...

- JSP Comments
 - JSP comment marks text or statements that the JSP container should ignore
 - Syntax:
 - `<%-- This is JSP comment --%>`
 - Example:
 - `<%-- This comment will not be visible --%>`

JSP Directives

- Directives tell to the web container how to translate JSP page into the corresponding Servlet
 - Syntax:
 - `<%@directive attribute="value" %>`
- There are three types of directives:
 - page directive
 - include directive
 - taglib directive

JSP Directives

- page directive
 - Defines attributes that apply to an entire JSP page
 - Different set of page directives like import, errorPage, language, info, extend, etc.
 - Example:
 - `<%@page import="java.util.Date" %>`
 - `<%@page import="org.my.java.DBConnector" %>`
 - `<%@page errorPage="error-404.jsp" %>`

JSP Directives

- include directive
 - Used to include the contents of any resource
 - Example:
 - `<%@include file= "myfirst.html" %>`
 - `<%@include file= "myfirst.jsp" %>`
- taglib directive
 - Used to define a tag library that defines many tags
 - E.g. : JSTL tags
 - Example:
 - `<%@taglib uri= "http://java.sun.com/jsp/jstl/core" prefix= "c" %>`

Exception Handling

- Mainly two methods are used to handle exceptions
 - Try....Catch Block
 - Default Error Page
 - `<%@ page errorPage = "ShowError.jsp" %>`
- In any of these cases, methods like `getMessage()` or `getCause()`, can be used to get specific details about the exception

Things We Covered

- Part I: Java Servlets
 - Dynamic Web Pages
 - Servlet Basics
 - Java Servlets with JDBC
 - Session Management and Cookies
- Part II: JSP
 - JSP basics
 - Processing a JSP File
 - JSP Life Cycle
 - JSP Syntax
 - JSP Directives

