

# Dynamic content: programming the web servers



# Step 2:

## let's use our web server to generate dynamic pages



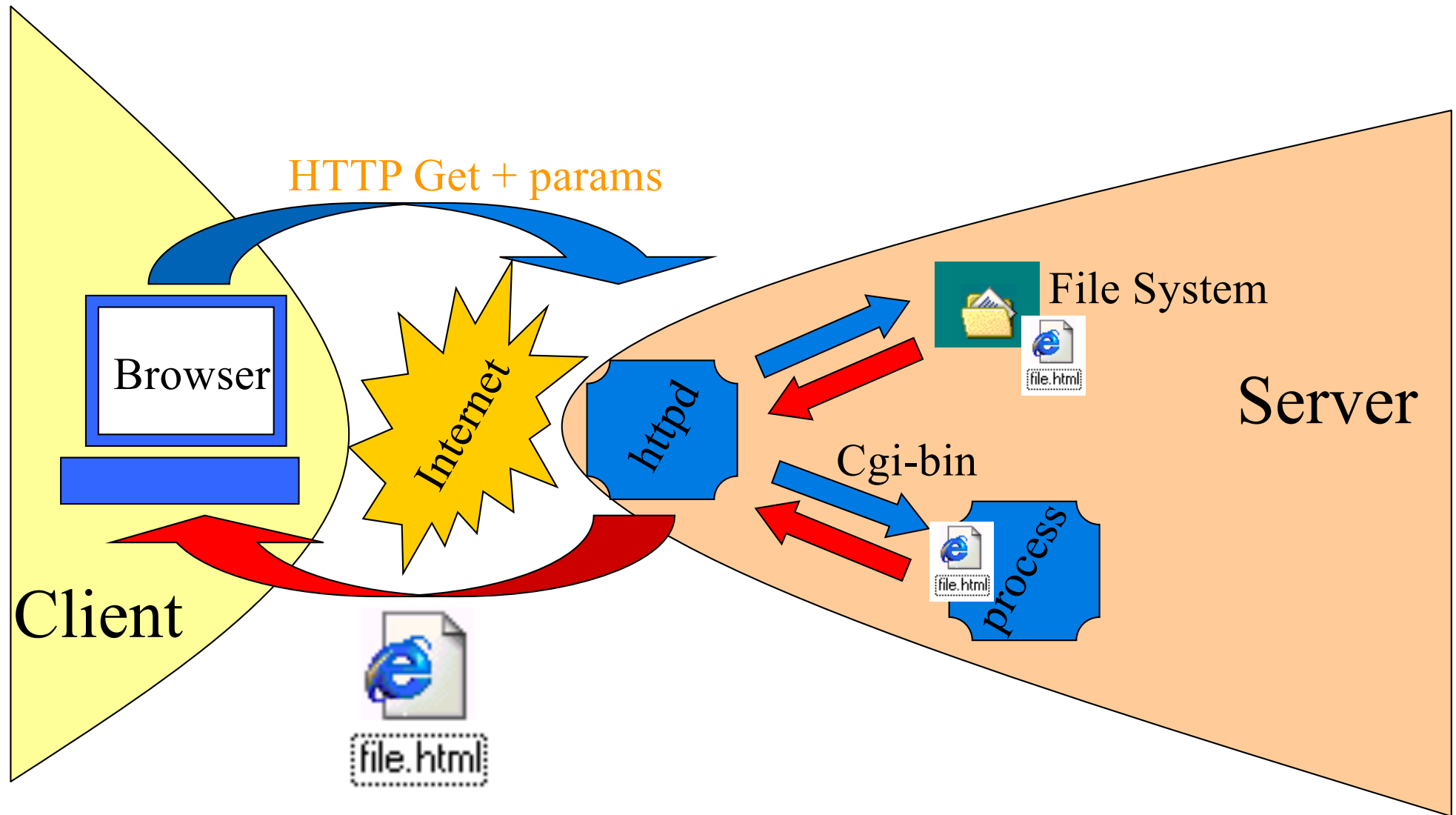
# Dynamic pages: the main idea:

We want to obtain NON STATIC information from the server. This implies executing some code on it, and send the results to the user.

e.g.: what's the time?

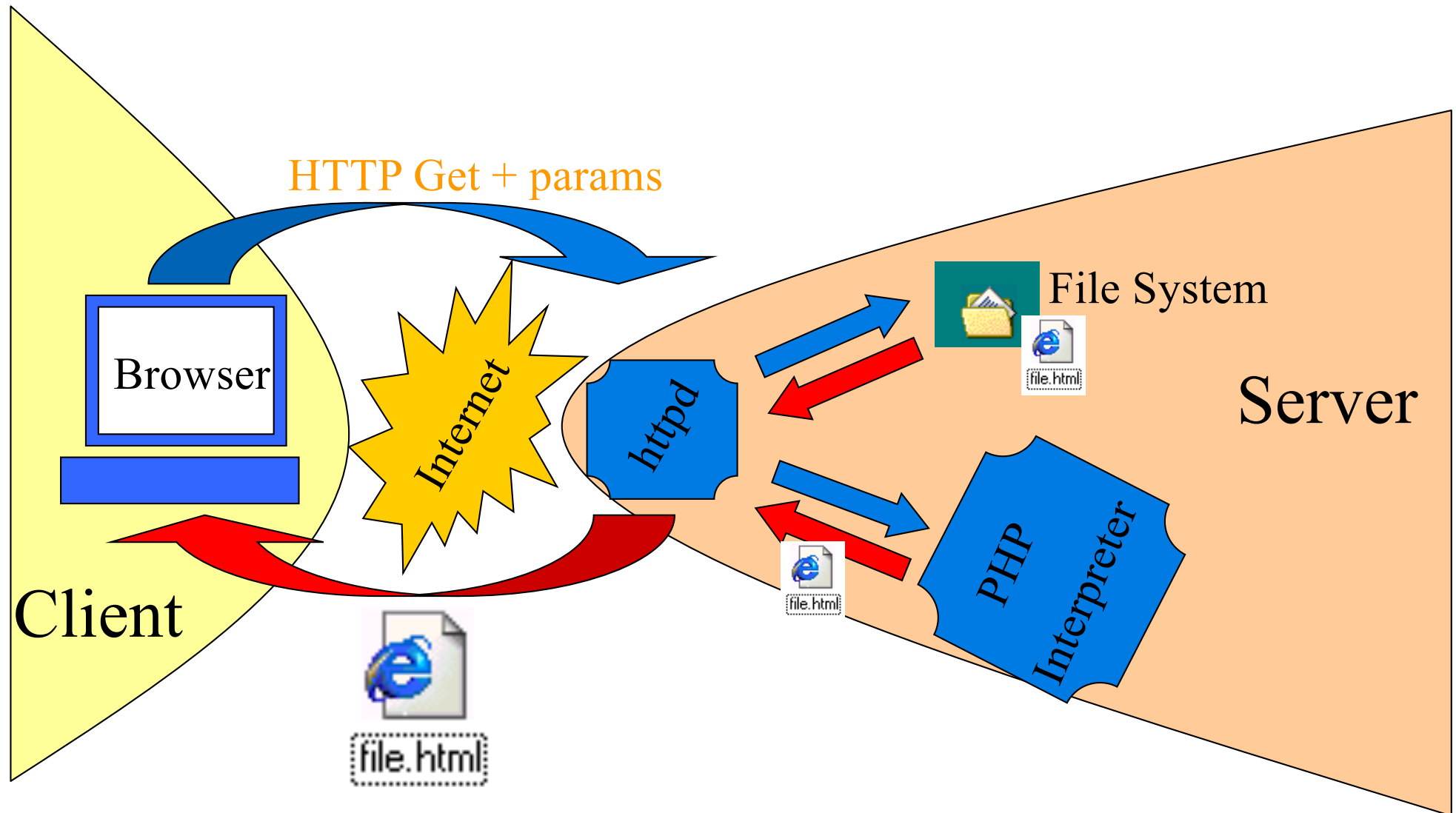


# The original web architecture: dynamic pages



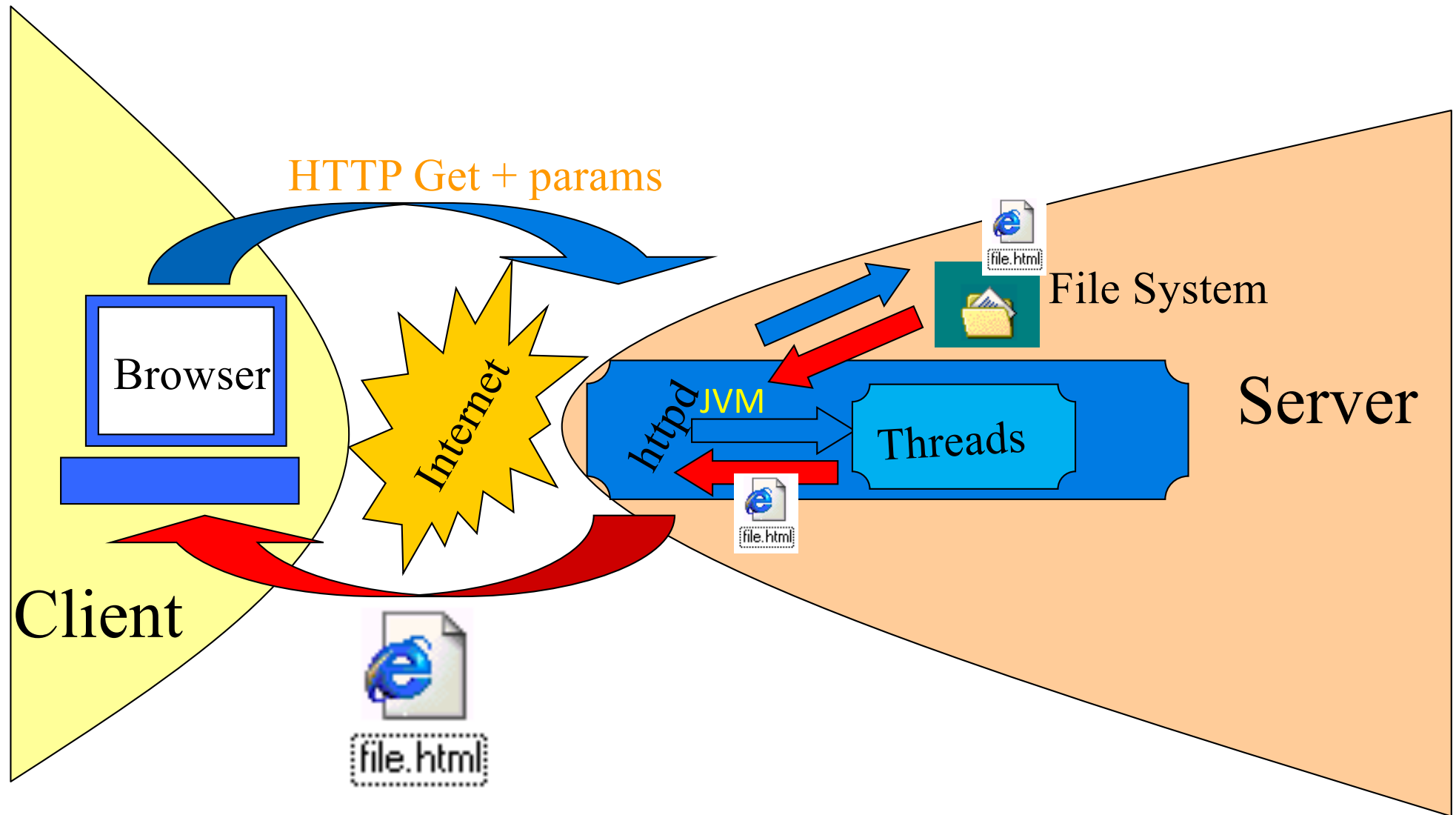
**Evolution 1: dynamically create (interlinked) documents**

## The original web architecture: dynamic pages



**Evolution 1: dynamically create (interlinked) documents**

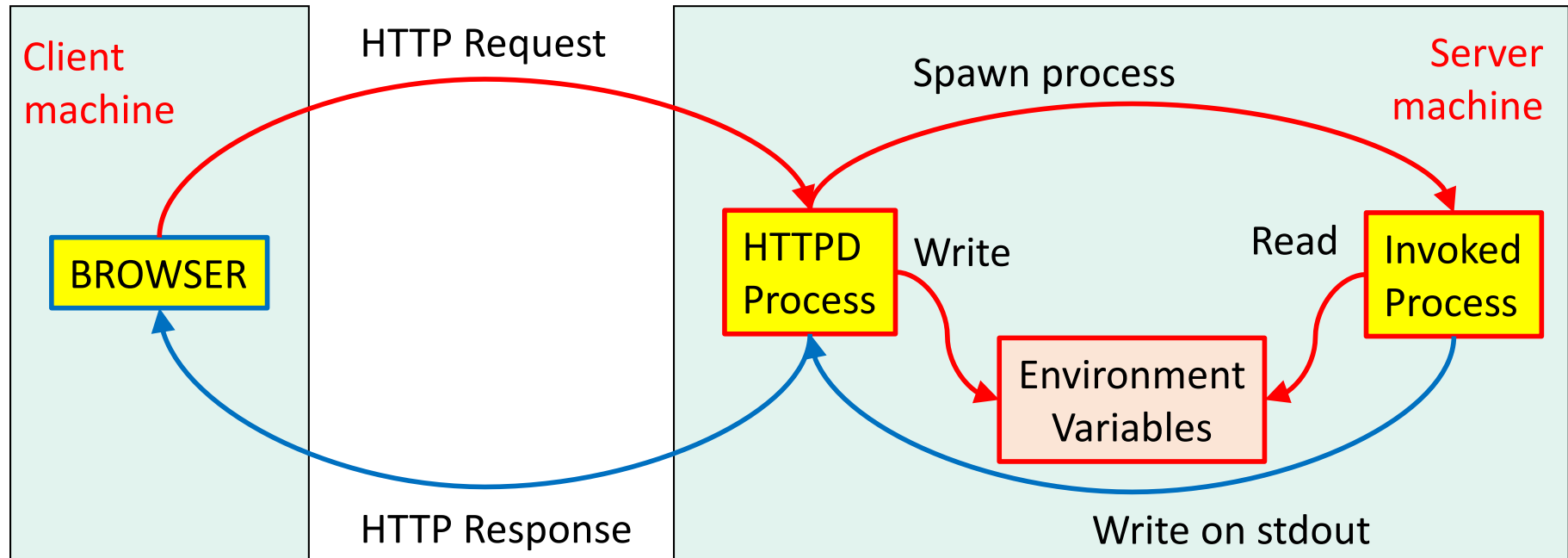
# The original web architecture: dynamic pages



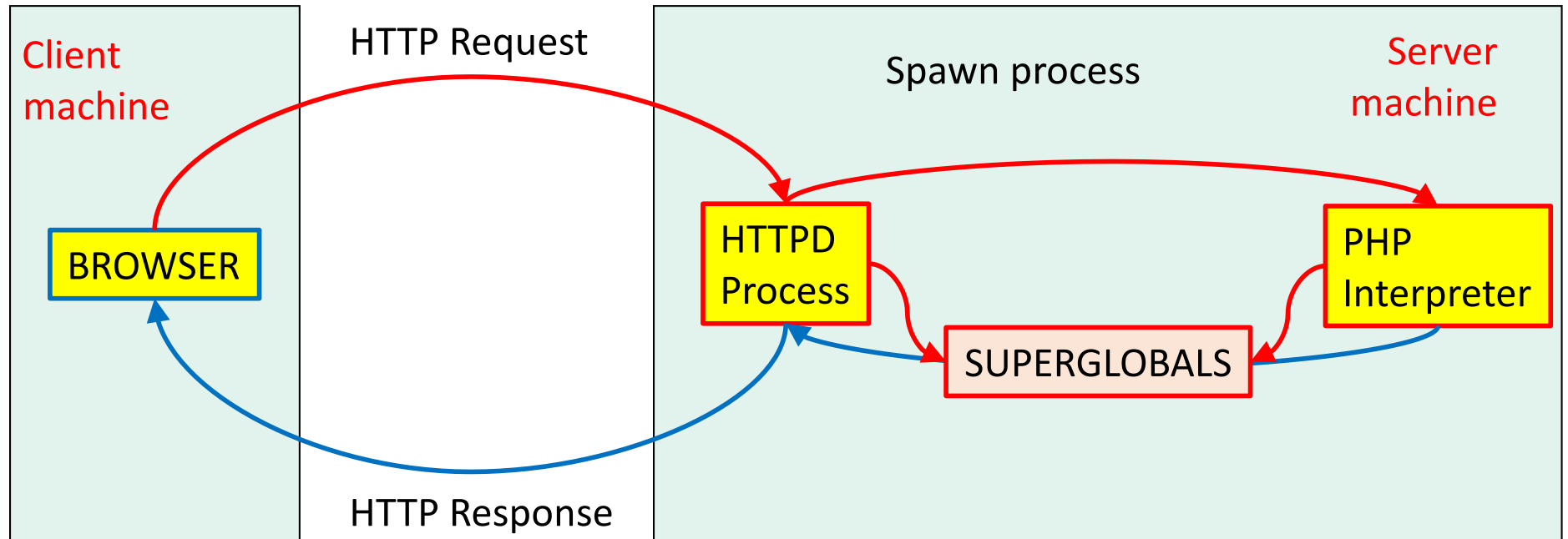
**Evolution 1: dynamically create (interlinked) documents**



# Getting info about the request

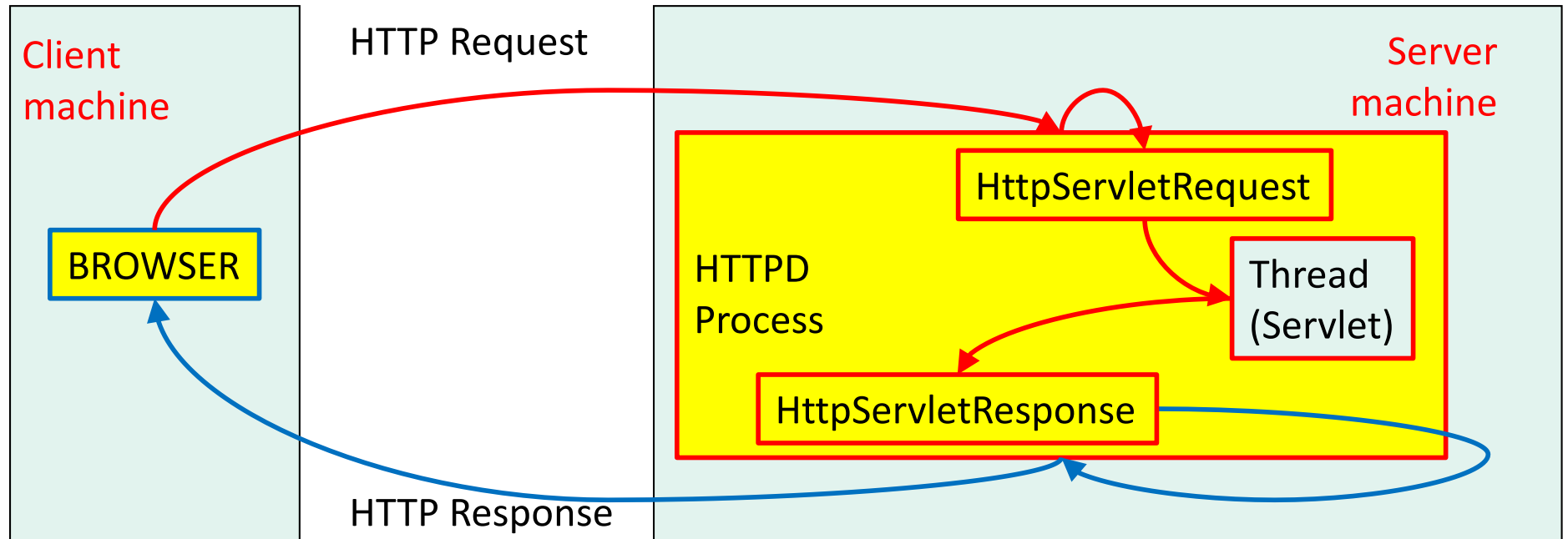


# Getting info about the request





# Getting info about the request



- Apache **Tomcat** is a **Java servlet container**, and is run on a **Java Virtual Machine**, or **JVM**.
- **Tomcat** utilizes the Java servlet specification to execute servlets generated by requests, often with the help of JSP pages, allowing dynamic content to be generated much more efficiently than with a CGI script.

# Apache Tomcat

<https://tomcat.apache.org/whichversion.html>

| Apache Tomcat Versions |          |         |                |                             |                       |                         |  |
|------------------------|----------|---------|----------------|-----------------------------|-----------------------|-------------------------|--|
| Servlet Spec           | JSP Spec | EL Spec | WebSocket Spec | Authentication (JASIC) Spec | Apache Tomcat Version | Latest Released Version | Supported Java Versions                    |
| 6.0                    | 3.1      | 5.0     | TBD            | TBD                         | 10.1.x                | 10.1.0-M5 (alpha)       | 11 and later                               |
| 5.0                    | 3.0      | 4.0     | 2.0            | 2.0                         | 10.0.x                | 10.0.11                 | 8 and later                                |
| 4.0                    | 2.3      | 3.0     | 1.1            | 1.1                         | 9.0.x                 | 9.0.53                  | 8 and later                                |
| 3.1                    | 2.3      | 3.0     | 1.1            | 1.1                         | 8.5.x                 | 8.5.71                  | 7 and later                                |
| 3.1                    | 2.3      | 3.0     | 1.1            | N/A                         | 8.0.x (superseded)    | 8.0.53 (superseded)     | 7 and later                                |
| 3.0                    | 2.2      | 2.2     | 1.1            | N/A                         | 7.0.x (archived)      | 7.0.109 (archived)      | 6 and later<br>(7 and later for WebSocket) |
| 2.5                    | 2.1      | 2.1     | N/A            | N/A                         | 6.0.x (archived)      | 6.0.53 (archived)       | 5 and later                                |
| 2.4                    | 2.0      | N/A     | N/A            | N/A                         | 5.5.x (archived)      | 5.5.36 (archived)       | 1.4 and later                              |
| 2.3                    | 1.2      | N/A     | N/A            | N/A                         | 4.1.x (archived)      | 4.1.40 (archived)       | 1.3 and later                              |
| 2.2                    | 1.1      | N/A     | N/A            | N/A                         | 3.3.x (archived)      | 3.3.2 (archived)        | 1.1 and later                              |

**SELECT VERSION 9 (Why?)**



# GET



Version: 2021.3.3  
Build: 213.7172.25  
17 March 2022

[Release notes](#)

[System requirements](#)

[Installation instructions](#)

[Other versions](#)

[Third-party software](#)

<https://www.jetbrains.com/idea/download>

FREE EDUCATIONAL  
LICENSE

<https://www.jetbrains.com/community/education/#students>

11

## Download IntelliJ IDEA

[Windows](#) [macOS](#) [Linux](#)

### Ultimate

For web and enterprise development

[Download](#)

Free 30-day trial

### Community

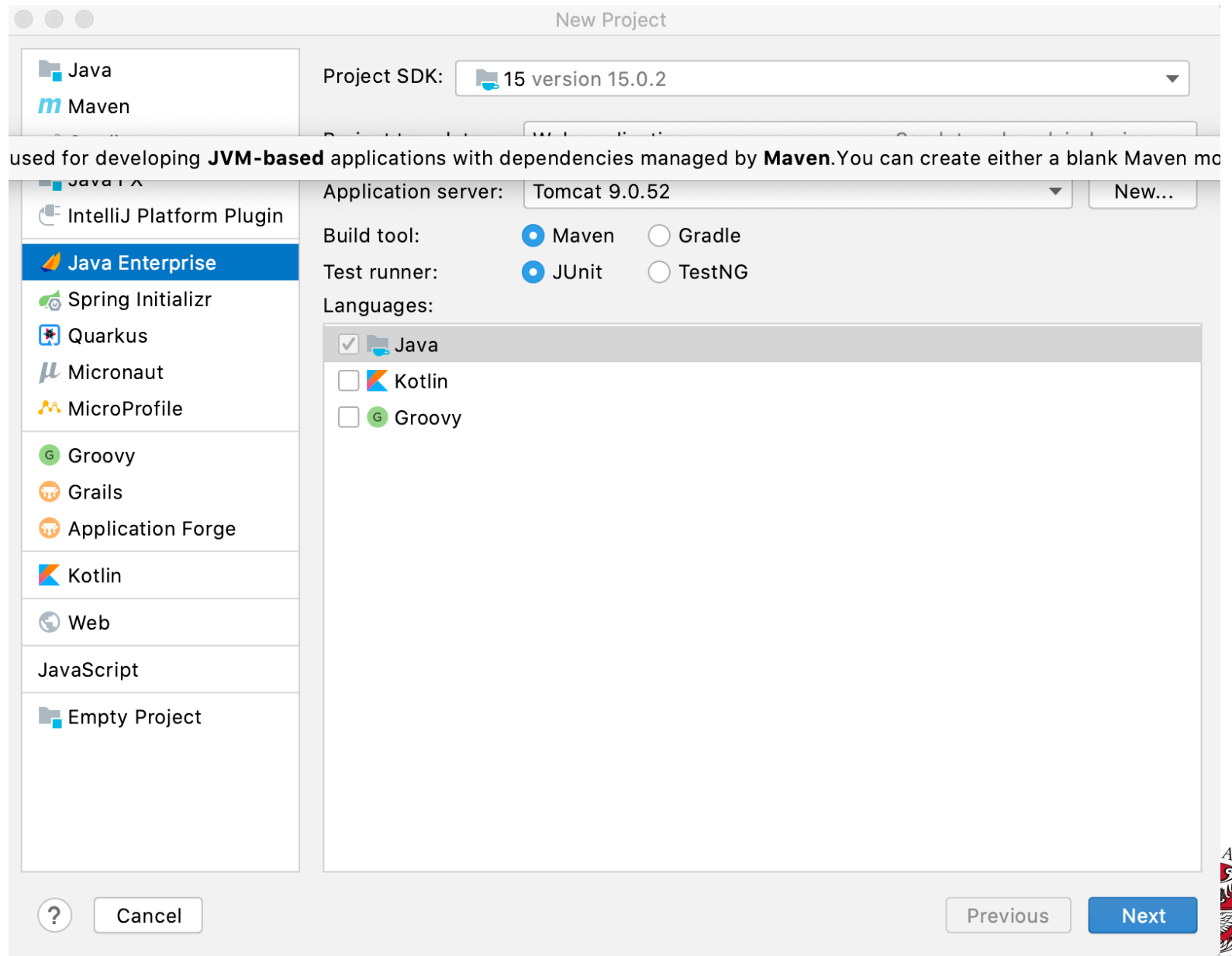
For JVM and Android development

[Download](#)

Free, built on open source

|   | IntelliJ IDEA Ultimate | IntelliJ IDEA Community Edition <span>?</span> |
|---|------------------------|--|
| Java, Kotlin, Groovy, Scala   | ✓                      | ✓  |
| Maven, Gradle, sbt  | ✓                      | ✓  |
| Git, GitHub, SVN, Mercurial, Perforce   | ✓                      | ✓  |
| Debugger  | ✓                      | ✓  |
| Docker  | ✓                      | ✓  |
| Profiling tools <span>?</span>  | ✓                      |  |
| Spring, Jakarta EE, Java EE, Micronaut, Quarkus, Helidon, and more <span>?</span> | ✓                      |  |
| HTTP Client   | ✓                      |  |
| JavaScript, TypeScript, HTML, CSS, Node.js, Angular, React, Vue.js                | ✓                      |  |
| Database Tools: SQL   | ✓                      |  |

# Create new JEE project



# Choose Java (not Jakarta)

New Project

Version:

Libraries and Frameworks:

- Specifications
  - Full Platform (8.0.1)
  - Web Profile (8.0.1)
  - Batch (1.0.1)
  - Bean Validation (2.0.1.Final)
  - Contexts and Dependency Injection (CDI) (2.0)
  - Concurrency Utils (1.1)
  - Connector Architecture (JCA) (1.7.1)
  - Enterprise Java Beans (EJB) (3.2.2)
  - JSON Binding (JSON-B) (1.0)
  - JSON Processing (JSON-P) (1.1.4)
  - Message Service (JMS) (2.0.1)
  - Model View Controller (MVC) (1.0.0)
  - Persistence (JPA) (2.2)

**Web Profile**  
Includes Servlet, CDI, JPA, Bean Validation, JSF, JAX-RS, WebSocket, JSON-P, JSON-B, Security, EJB and JTA specifications.

New Project

Name:

Location:

Artifact Coordinates

Group:   
The name of the artifact group, usually a company domain

Artifact:   
The name of the artifact within the group, usually a project name

Version:

13



MyWebProjects – HelloApplication.java

MyWebProjects > src > main > java > com > example > MyWebProjects > HelloApplication

```
1 package com.example.MyWebProjects;
2
3 import ...
4
5
6 @ApplicationPath("/api")
7 public class HelloApplication extends Application {
8
9 }
```

Throw away these files

Services

- Tomcat Server
  - Not Started
    - Tomcat 9.0.52 [local]
      - MyWebProjects:war exploded

Here you can manage Tomcat

Starting and stopping Tomcat  
Undeploying WebApps



The screenshot shows the IntelliJ IDEA IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The main editor window displays the file path `MyWebProjects > src > main > java > com > example > MyWebProjects` and the content of `index.jsp`. The code is as follows:

```
1 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <title>JSP - Hello World</title>
```

The 'New' menu is open, showing various options. The 'Servlet' option is highlighted. Other options include Java Class, Kotlin Class/File, File, Scratch File, Package, package-info.java, HTML File, Stylesheet, JavaScript File, TypeScript File, package.json File, Kotlin Script, Kotlin Worksheet, OpenAPI Specification, Swing UI Designer, EditorConfig File, Resource Bundle, XML Configuration File, Diagram, Data Source, DDL Data Source, Data Source from URL, Data Source from Path, Data Source in Path, Driver and Data Source, Driver, Web Filter, Web Listener, HTTP Request, RESTful Web Service, and RESTful Web Service Client.



```

package it.unitn.disi.ronchet.myservlets;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(name = "ReadPost", urlPatterns = {"/ReadPost"})
public class ReadPost extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
                        HttpServletResponse response)
                        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request,
                        HttpServletResponse response)
                        throws ServletException, IOException {
        processRequest(request, response);
    }
}

```





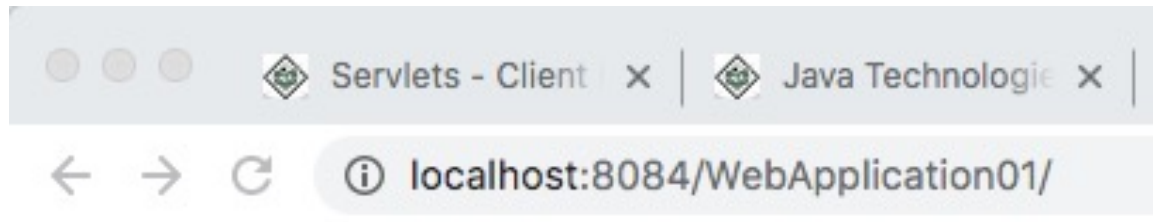
```

/**
 * Processes requests for both HTTP GET and POST
 * methods.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
protected void processRequest(HttpServletRequest request,
                               HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html; charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html><head>");
        out.println("<title>Servlet ReadPost</title>");
        out.println("</head><body>");
        out.println("<h1>Servlet ReadPost at "+request.getContextPath()+"</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}

```



# Output

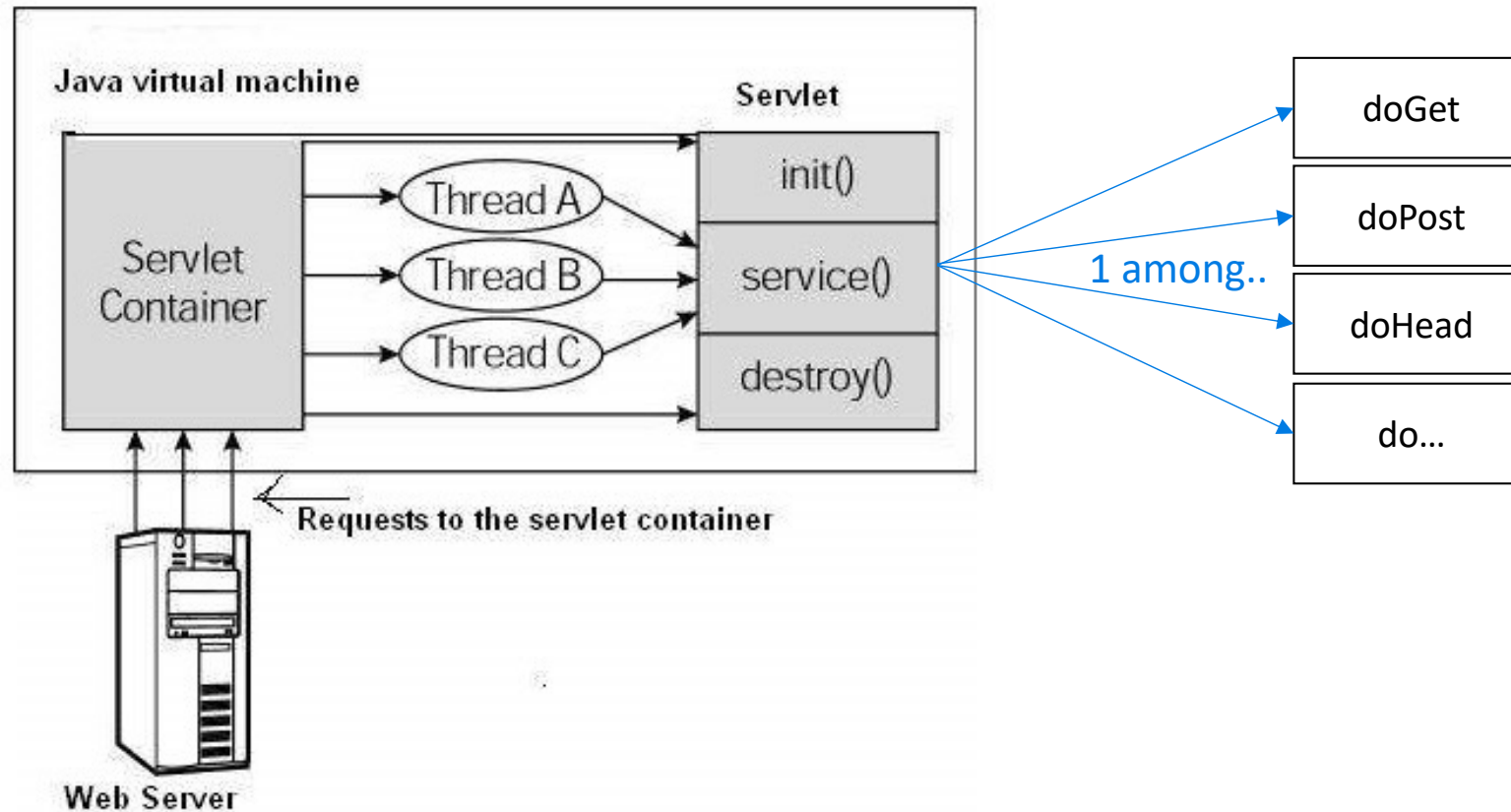


TODO write content



## Servlet ReadPost at /WebApplication01

# Servlet lifecycle



See <https://www.tutorialspoint.com/servlets/servlets-life-cycle.htm>

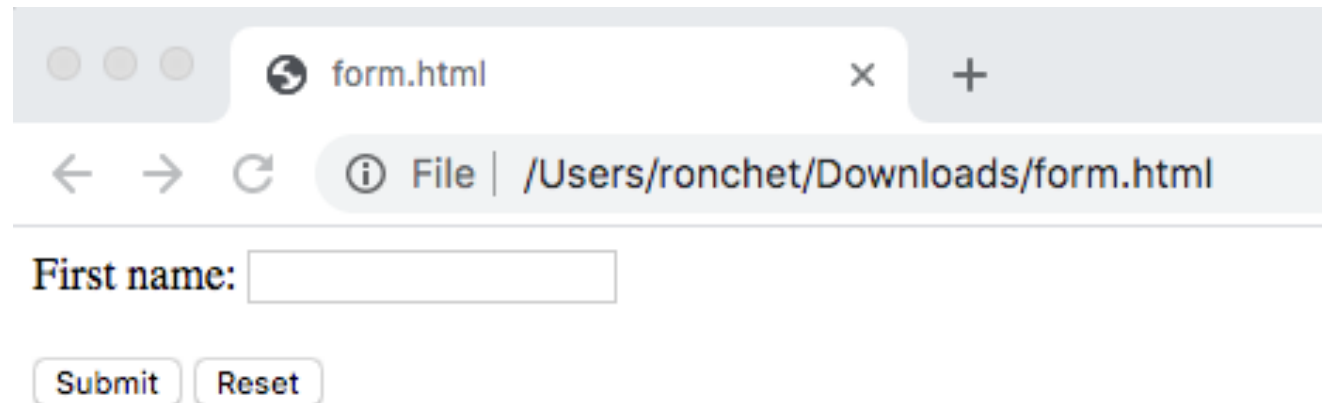
# Step 3: let's pass parameters in our request

## HTML Forms



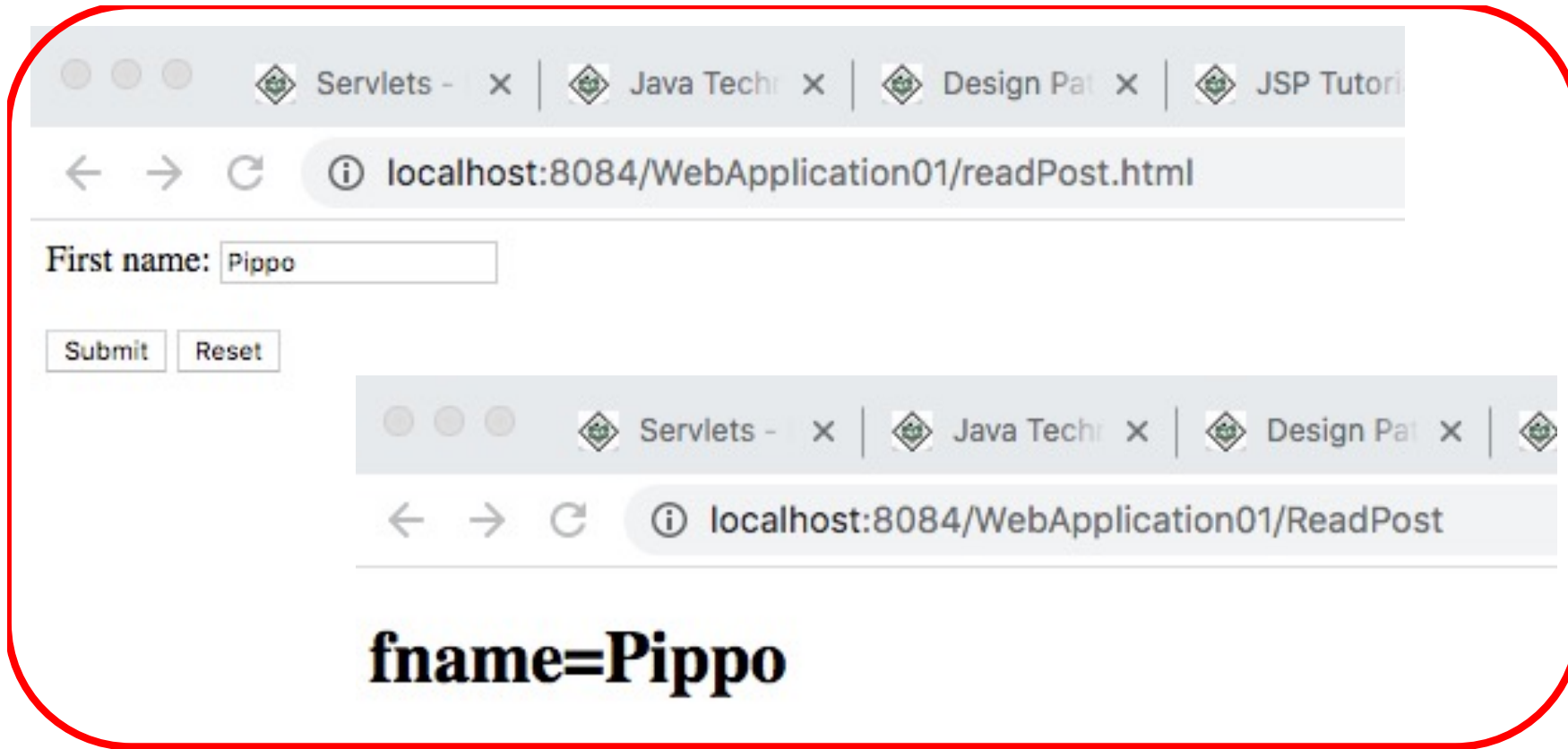
# Parameters passing

```
<html>
<body>
<form action="http://localhost:8084/WebApplication01/ReadPost" method="post">
  <label for="fname">First name:</label>
  <input type="text" name="fname"><br><br>
  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>
</body>
</html>
```



```
protected void processRequest(HttpServletRequest request,
                             HttpServletResponse response)
    throws ServletException, IOException {
    String name = request.getParameter("fname");
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html><head>");
        out.println("<title>Servlet ReadPost</title>");
        out.println("</head><body>");
        out.println("<h1> fname="+name+"</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```





Post



Servlets - x | Java Techn x | Design Pat x | JSP Tutori

localhost:8084/WebApplication01/readPost.html

First name:

Submit Reset

localhost:8084/WebApplication01/ReadPost

**fname=Pippo**

Post

Servlets - x | Java Techn x | Design Pat x | JSP Tutori x

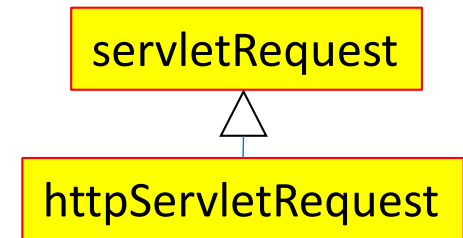
localhost:8084/WebApplication01/ReadPost?fname=Minnie

**fname=Minnie**

Get



# Getting parameters from `HttpServletRequest`:



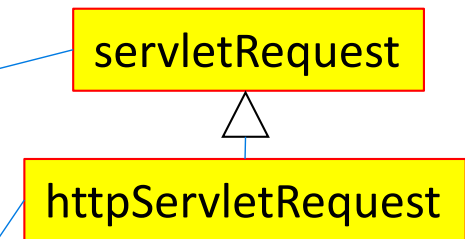
|  |   |
|--|---|
| <code>String</code>                      | <code>getParameter(String name)</code><br>Returns the value of a request parameter as a <code>String</code> , or <code>null</code> if the parameter does not exist.   |
| <code>Map&lt;String, String[]&gt;</code> | <code>getParameterMap()</code><br>Returns a <code>java.util.Map</code> of the parameters of this request.   |
| <code>Enumeration&lt;String&gt;</code>   | <code>getParameterNames()</code><br>Returns an <code>Enumeration</code> of <code>String</code> objects containing the names of the parameters contained in this request.  |
| <code>String[]</code>                    | <code>getParameterValues(String name)</code><br>Returns an array of <code>String</code> objects containing all of the values the given request parameter has, or <code>null</code> if the parameter does not exist. |

See <https://docs.oracle.com/javaee/7/api/javax/servlet/ServletRequest.html>

# What else can I obtain from `HttpServletRequest`?

- Character encoding
- Content length
- Content Type
- Locale
- Protocol
- Local/remote name, address, port
- Server name
- Is Secure ?

- Headers
- Method
- Path Info
- Query String
- Session
- Cookie

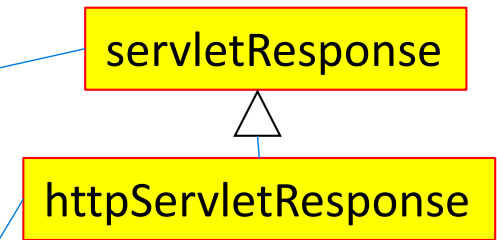


See <https://docs.oracle.com/javaee/7/api/javax/servlet/ServletRequest.html>

# What can use `HttpServletResponse` for?

- `getWriter`
- `setCharacterEncoding`
- `setContentLength`
- `setContentType`
- `setLocale`

- `addHeader`
- `sendError`
- `sendRedirect`
- `setStatus`
- `Session`
- `addCookie`



See <https://docs.oracle.com/javaee/7/api/javax/servlet/ServletResponse.html>

# Readings

<https://www.tutorialspoint.com/servlets/index.htm>

A screenshot of a web page titled 'Servlets Tutorial'. The title is in a red header bar. Below the header is a list of ten topics, each preceded by a small icon of a book. The topic 'Servlets - Form Data' is highlighted with a grey background. The topics are: Servlets - Home, Servlets - Overview, Servlets - Environment Setup, Servlets - Life Cycle, Servlets - Examples, Servlets - Form Data, Servlets - Client Request, Servlets - Server Response, and Servlets - Http Codes.

- Servlets - Home
- Servlets - Overview
- Servlets - Environment Setup
- Servlets - Life Cycle
- Servlets - Examples
- Servlets - Form Data
- Servlets - Client Request
- Servlets - Server Response
- Servlets - Http Codes

Stop here!

