

Servlets

Presented by Hosein Zare
Twitter: @zare88r

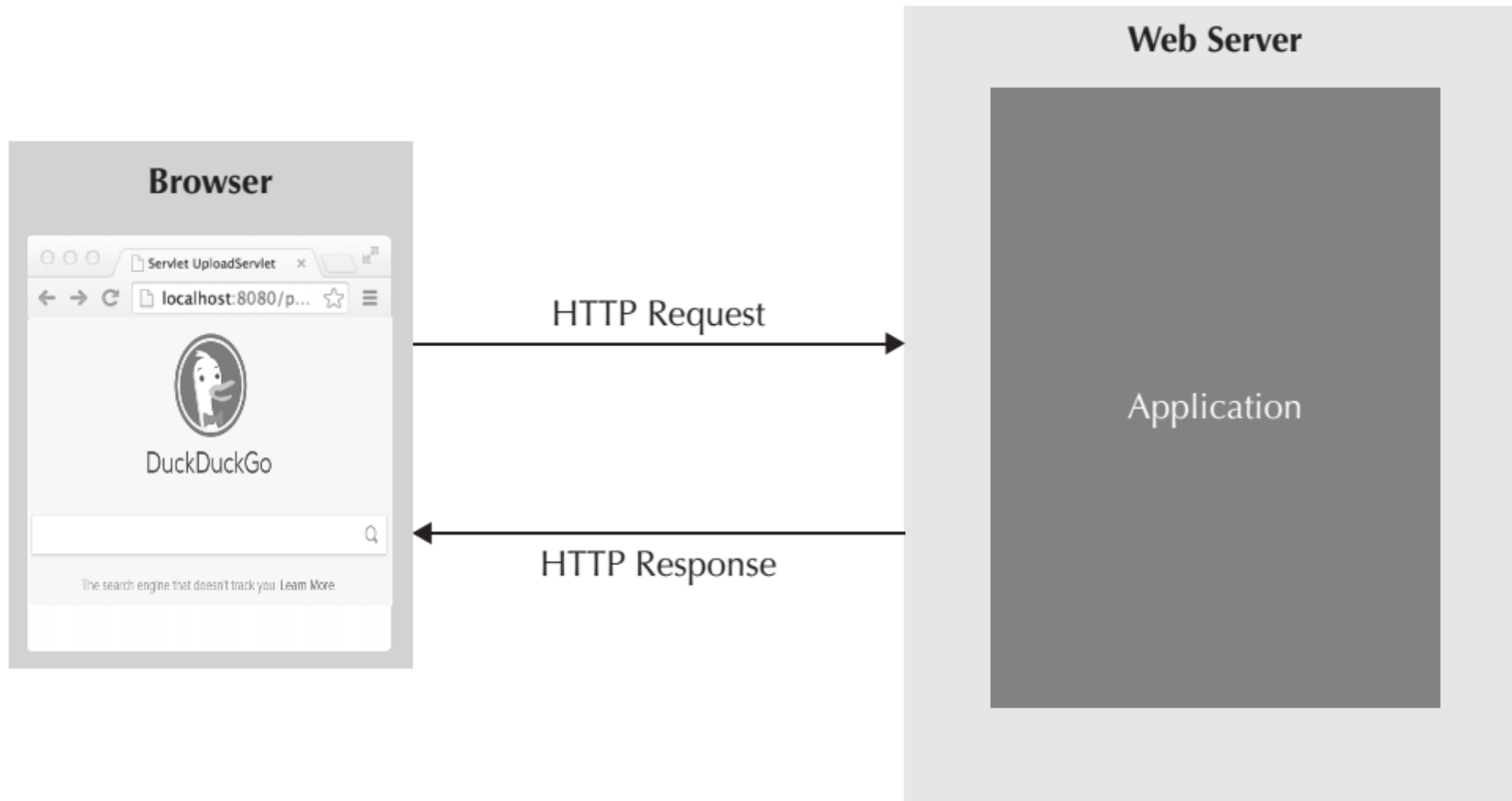




Http Protocol

- Http یک پروتکل کلاینت و سرور بر مبنای TCP میباشد.
- کلاینت یک درخواست Http به سرور ارسال میکند و منتظر پاسخ میماند.
- سرور درخواست را دریافت کرده ، آن را پردازش میکند و به کلاینت پاسخ ارسال میکند

Http Protocol





URI(Uniform resource identifier)

- URI به نام host وب سرور شما اشاره دارد (میتوان به IP هم آدرسی دهی کرد)

- مثال :

- `http://www.searchengine.com/images.html`
- `http://192.168.25.226:9000/hr/projects?id=project1`



URI(Uniform resource identifier)

Http Request

Headers

GET /myphotos.html HTTP/1.1
host : photoserver.com
accept : text/html, application/xml
user-agent : Mozilla/5.0
accept-encoding : gzip
accept-language : en-US

Body

<empty>

Http Response

Headers

HTTP/1.1 200 OK
server : GlassFish Server Open Source
Edition 4.0
content-type : text/html; charset=UTF-8

Body

```
<html>
<head>
  <title>My Photos</title>
</head>
<body>
  <h3 align='center'>My Photos</h3>
  <table align='center'><td>
    ....
  </td>
</table>
</body>
</html>
```



Http Methods

- GET
- POST
- HEAD (Like GET but only headers)
- PUT
- PATCH (apply partial modifications to a resource)
- DELETE
- TRACE
- OPTION (HTTP methods that the server supports)
- CONNECT (establishes a tunnel to the server)



Java Servlets

- جاوا سرولت ها اشیائی هستند که درخواست های HTTP را در سمت سرور پردازش میکنند
- اقداماتی که یک سرولت انجام میدهد عبارتند از :
 - بررسی Header های درخواست
 - خواندن body درخواست
 - تولید دیتا برای پاسخ به کلاینت
 - مشخص کردن Header پاسخ
 - نوشتن body پاسخ

```

@WebServlet(name = "TemplateServlet",
            urlPatterns = {"/TemplateServletURI"})
public class TemplateServlet extends HttpServlet {

    @Override
    protected void service(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {

        for (Enumeration e = request.getHeaderNames();
             e.hasMoreElements(); ) {
            String nextRequestHeaderName = (String) e.nextElement();
            String nextRequestHeaderValue =
                request.getHeader(nextRequestHeaderName);
            // 1. Examine request headers
        }

        // 2. Read request body
        InputStream is = request.getInputStream();
        // or
        Reader reader = request.getReader();

        // 3. Gather data to send back

        // 4. Set response headers
        response.setHeader(myResponseHeaderName,
                           myResponseHeaderValue);

        // 5. Write response body
        PrintWriter writer = response.getWriter();
        // or
        OutputStream output = response.getOutputStream();

    }
}

```




متود های Java Servlet

HttpServlet method	Purpose
<code>public void service(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException</code>	Handle all HTTP requests
<code>protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException</code>	Handle only HTTP GET requests
<code>protected void doHead(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException</code>	Handle only HTTP HEAD requests
<code>protected void doOptions(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException</code>	Handle only HTTP OPTIONS requests
<code>protected void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException</code>	Handle only HTTP POST requests
<code>protected void doPut(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException</code>	Handle only HTTP PUT requests
<code>protected void doTrace(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException</code>	Handle only HTTP TRACE requests



چرخه حیات Servlet ها

MyServlet

void init(ServletConfig config)

void doXXX(HttpServletRequest req,
HttpServletResponse res)

void destroy ()



HttpServletRequest

- این کلاس نماینده درخواست های Http میباشد
- این url را فرض کنید:

`http://127.0.0.1:7001/hr/employee$list?dep_name=fin^com`

Encoded :

`http%3A%2F%2F127.0.0.1%3A7001%2Fhr%2Femployee%24list%3Fdep_name%3Dfin%5Ecom%0A`

HttpServletRequest

Method	URL-Decoded	Result
<code>getContextPath()</code>	no	/hr
<code>getLocalAddr()</code>		0:0:0:0:0:0:0:1
<code>getLocalName()</code>		0:0:0:0:0:0:0:1
<code>getLocalPort()</code>		7001
<code>getMethod()</code>		GET
<code>getPathInfo()</code>	yes	employee\$list
<code>getProtocol()</code>		HTTP/1.1
<code>getQueryString()</code>	no	dep_name%3Dfin%5Ecom%0A
<code>getRequestedSessionId()</code>	no	wsWtBXstJK1GRXmw9HIC3Qe1vfX0G_OcD1VaKFd55tgQn8h1bEme!644890759!1519026273069
<code>getRequestURI()</code>	no	hr%2Femployee%24list%3Fdep_name%3Dfin%5Ecom%0A
<code>getRequestURL()</code>	no	http%3A%2F%2F127.0.0.1%3A7001%2Fhr%2Femployee%24list%3Fdep_name%3Dfin%5Ecom%0A
<code>getScheme()</code>		http
<code>getServerName()</code>		127.0.0.1
<code>getServerPort()</code>		7001
<code>getServletPath()</code>	yes	/hr
<code>getParameterNames()</code>	yes	[dep_name]
<code>getParameter("dep_name")</code>	yes	fin^com



HttpServletRequest

Method	Result
<code>BufferedReader getReader()</code>	این متود <code>body</code> درخواست را میخواند (بر مبنای کاراکتر)
<code>ServletInputStream getInputStream()</code>	این متود <code>body</code> درخواست را میخواند (بر مبنای بایت ها)
<code>Enumeration<String> getHeaderNames()</code>	دریافت لیست تمامی <code>header</code> های دریافتی
<code>String getHeader("Host")</code>	دریافت مقدار یک <code>header</code> خاص
<code>String getRemoteAddr()</code>	اطلاعات آدرس سیستم کلاینت
<code>String getRemoteHost()</code>	اطلاعات هاست سیستم کلاینت
<code>int getRemotePort()</code>	اطلاعات پورت سیستم کلاینت



HttpServletResponse

Method	Result
<code>PrintWriter getWriter()</code>	یک شی کاراکتر محور برای نوشتن <code>body</code> پاسخ
<code>ServletOutputStream getOutputStream()</code>	یک شی بایت محور برای نوشتن <code>body</code> پاسخ
<code>addHeader(String name, String value)</code>	اضافه کردن <code>header</code> به پاسخ
<code>addDateHeader(String name, long date)</code>	اضافه کردن <code>header</code> از جنس تاریخ به پاسخ
<code>void sendRedirect(String location)</code>	درخواست به کلاینت برای عوض کردن آدرس وب