

U-3 (Server Side Technologies)

Page No.

Date

dynamic generation of web servers served up by web servers as opposed to static web pages.

~~Server Side Scripting -~~

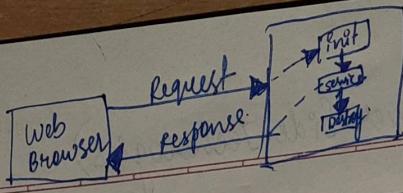
- all calculations are done by server your website is hosted on.
- script interpreted by supported languages parser, like PHP or ASP.

When to use server side scripts?

- password protection
- Browser sniffing / customization
- Form processing.
- Building & Displaying page.

- Adv →
- Browser-independent
 - maintaining code - easy
 - global variables → available
 - customize webpages dynamically
 - secured scripts
 - provide live data
 - control → security of data
 - Device mapping
 - communication (other programs)
 - retrieval of data
 - authentication, authorization & session.

- Disadv →
- no debugging tool
 - no control over UI
 - more difficult
 - hard to acknowledge when session ends.



Page No.

Date



Servlet - Java programming class that is used to extend

capabilities of servers that host applications.

programs that run on Web or Application server & act as middle layer between a request

coming from a web browser or other HTTP client & databases or application on HTTP server.

- **Functionality of Servlet**

- 1) Read explicit data from client -
- 2) Read implicit HTTP request & data sent by browser.
- 3) Generate results
- 4) Send implicit HTTP response data.
- 5) Send explicit data (document) to client

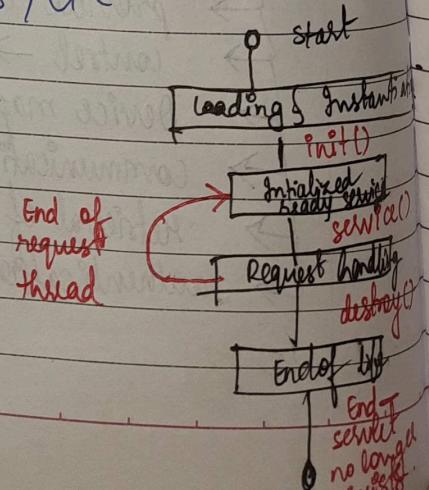
in comparison

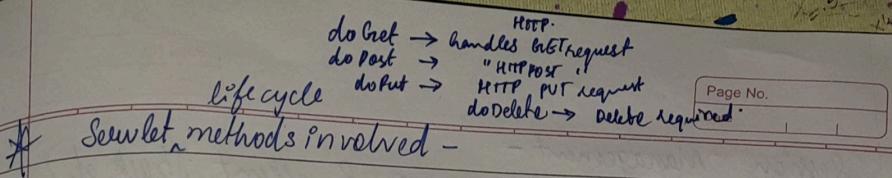
- **Advantages of Servlets with CGI -**

- Better Performance
- creation of separate process isn't need
- platform-independent → written in Java
- security manager → set of restrictions → protect resources
- Full functionality → Java Class Libraries → available to serve
- communicate with applets, databases, etc

- **Servlet Life Cycle -**

- Server Class is loaded
- Servlet instance is created
- init method is invoked
- service method is invoked
- destroy method is invoked





- **init()**
 - call only once
 - called → server first created
 - one time initialization.
 - single instance of each servlet is created
 - creates or loads some data that will be used throughout life of servlet

- **service()**
 - perform actual task.
 - servlet container calls service method to handle request from client.
 - writes formatted response back to client.
 - serves → new thread & calls services.
 - service method → checks HTTP ~~get~~ request type.
(GET, POST, DELETE, etc)

- **destroy()**
 - called once → end of life cycle of servlet.
 - close database connections
 - halt bg threads, write cookie lists or hit counts to disk.
 - after this method, servlet object is marked for garbage collection.

* Parameter Data-

- **Servlet Request.get Parameter Values()** → string array with all values present
- **Servlet Request.get Parameter()** → first value for given parameter
- **Servlet Request.get Parameter Names()** → returns enumeration of string objects presenting names.
- **Servlet Request.get Parameter Map()** → returns a Java Util Map object
keys in map are type string.

* Session Management -
 ↳ conversion b/w server & client (back of)
 ↳ series of continuous request & response.

* Session tracking → capability of a server to maintain a current state of single client.
 ↳ techniques

- 1) User Authorization
- 2) Hidden fields
- 3) URL rewriting
- 4) Cookies
- 5) Session tracking API

getSession() → create sessions (HTTPSessionRequest)
 getAttribute() → y HTTPsession class.
 setAttribute()
 removeAttribute()

GET

POST

get data

send data

limited → data in header

unlimited → data in body

bookmarked

cannot bookmark.

first
second

is non-imperative

cachable

not cachable

VOL → data in header

data in body.

not secure

secure

Used more

Used less

only string

String, numeric, binary



990366312484523

MRP ₹ 130/-

Inclusive of all taxes

For Educational
Official Us

cookies

- web server → unique session ID → cookie (web client)
- subsequent requests from client → received code.
- some little information → left on computer by other computer → access an internet
- info left → advertising agency on internet.
- info stored in cookies → advertising agency → keep track of internet usage, linking of users.
- name / email → remembered → cookies (in system).
- cookies → dangerous → malicious data → local disk.
- on you → maintain privacy & security.
- cookie class → create cookies in servlet.
- syntax → cookie()

Cookie (String name, String value)

public String getName() → return name of cookie.
 .. . getValue() → return value of cookie.
 " " set Name() → set or changes name of cookie.
 " " set Value() → set or changes value of cookie.

public void addCookie (Cookie c) → cookie added → response object of
 HttpServletResponse interface

public Cookie[] get Cookies() → cookies can be returned → method
 help of HttpServletRequest Interface

- in servlet → step used to support cookies
 - 1) creation of cookies → (addCookie method)
 - 2) Reading cookies → (getName() & getValue())

* Data storage -

efficient use of file system or database management system.
is required by every web application.

- commonly used techniques for data storage.

1) Java Object Serialization

↳ serialization → mechanism of writing the state of an object
into data stream (byte).

2) JDBC → Java Database connectivity.

↳ API (appln Programming Interface).

↳ classes, interfaces, exceptions → Java appln → sent SQL
statement to database.

↳ connectivity with RDBMS packages.

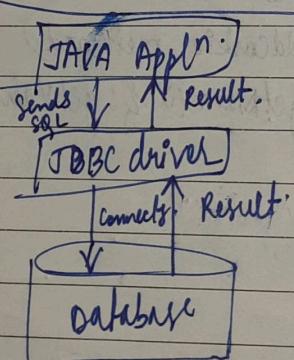
↳ JDBC API → classes, interfaces, exceptions → establish connection
with data source.

↳ JDBC API → java.sql & javax.sql packages.

Driver Manager → Java appln → connection to database involves driverManager.

↳ connection → connection with data source.

↳ statement → pto representing SQL statements.



↳ USES (JDBC)

- Helps client to store & retrieve data from dB

- client → update database

* MySQL → SQL commands.

- creating database → CREATE DATABASE mydb;
- displaying all the databases → SHOW DATABASES;
- selecting particular database → USE MYDB;
- creating table → CREATE TABLE my-table (id INT(4) name VARCHAR(20));
- displaying table → SHOW TABLES;
- displaying table fields → DESCRIBE my-table;
- inserting values into table → INSERT INTO my-table;
- displaying contents of table → SELECT * FROM my-table;
- updating record → UPDATE my-table
SET name = 'PRIYA'
WHERE id = 4;
- deleting record → DELETE from my-table
WHERE id = 3;
- deleting table → drop table my-table;
- inner join → SELECT customer.Name, customer.city, order.Order_No.
FROM customer
INNER JOIN order
ON customer.Cust_Id = order.Order_Id
- order by clause → SELECT * FROM my-table ORDER BY name;

XML Document

- extensible Markup Language.
- similar to HTML → scripting language various tags
- tags → not predefined
- user define own tags.
- HTML → representation of data
XML → transport or to store data.

Uses →

- display meta contents
- exchange data b/w applications.
- extract data from database → diff appln → diff task on data.

Advantages →

- Human readable & edit it on simple text editors.
- language neutral → Java program → generate XML doc.
- free structure → complex data systematically.
- independent of OS.

Goals →

- user → define & use tag → restrict use of set of tags
- build own library of tag → web requirement
- formatting rules → user defined tags.
- XML → storage or support data.

Feature → Transport or store data

- define & use own tag.
- decide how to display data.
- create tags → any type of info.
- searching, sorting, rendering or manipulation.
- XML doc → validate → external tools.

Limitations →

- redundant & verbose
- less readable.
- syntax → higher complexity
- array (no support)
- large size

MRP ₹ 130/-

Inclusive of all taxes

For Education
Official

* Document Type Definition (DTD)

- MOD XML

- define basic building block of any XML doc.
- DTD → specify element type, attribute & relationship.
- DTD → specify set of rules → structuring data → XML file.

- elements -
- attribute
- cdata → character data → parsed by parser
- pCDATA → parsed character data (i.e. text).

Merits -

- define structural component of → XML doc.
- simple & compact.
- defined inline & directed embedded.

Demerits -

- basic → not for complex document (specific)
- various framework by XML → not supported by DTD.
- can't define type of data.
- XML processor → don't understand DTDs.
- DTD's → not aware of datatype namespace concept

XML

- ## * Schema
- structure of XML document.
 - alternative to XML-DTD.
 - XML Schema Definition (XSD) language
 - defines elements, attributes, elements having child elements.
 - use datatypes.

Datatypes → • string • date • numeric

Adv → • specific

- schema file → aware of namespaces
- support datatypes
- W3C recommendation
- built-in types

• boolean.

Disadv → complex to design

- hard to learn
- doc → schema file
- large & complex operation
- slow down

ECOFRIENDLY



Page No.

Date

* XML DOM -

- document object model → set of platform independent & language neutral
 - API → access & manipulate info stored in XML or HTML doc.
 - Loading
Accessing
Deleting
Changing
- Y elements of XML document.

* XSLT → extensible stylesheet Language

- W3C recommendation
- define XML transformations & presentations.
- transforming XML docs. → XSLT
- navigating in XML docs. → XPath
- displaying it in desired manner → XSL-FO

AJAX -

- It's asynchronous Java Script & XML.
 - ↓
execution of script doesn't disturb user's work
 - ↓
use of Javascripting actual work
 - Javascript with XML
↳ support to task in AJAX.
- web docs → adopts certain standards
- exchange data with server & updates part of web document

Merits -

- faster → increase performance & speed
- asynchronous HTTP request
- form validation
- fetch data from database & storing

Demerits -

- browser compatibility issues
- bookmark AJAX → impossible
- search engine → can't crawl → AJAX generated content