HTTP protocol

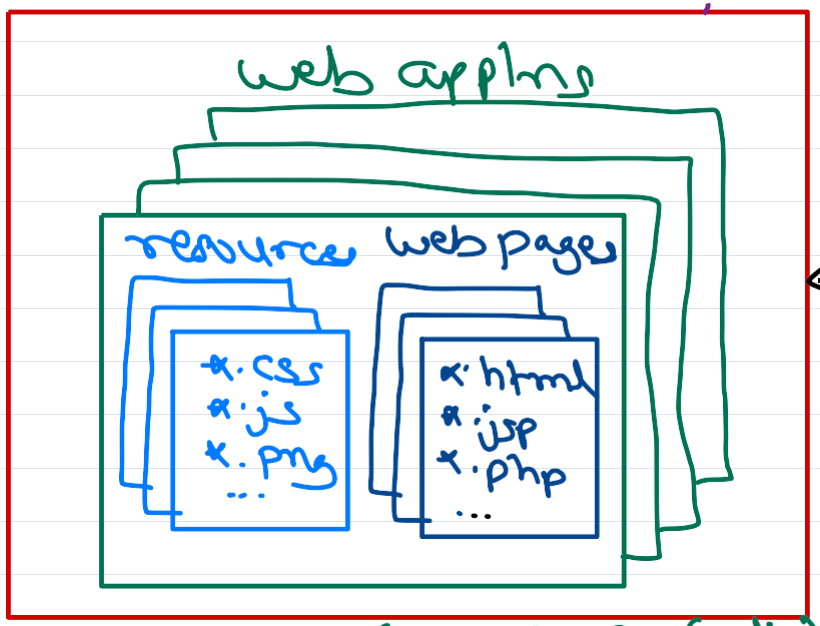
web server
web appln
web page

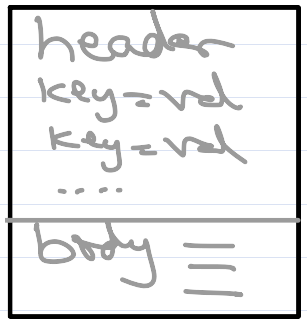
web client

server

client

web server e.g. apache, tomat, IIS, ...





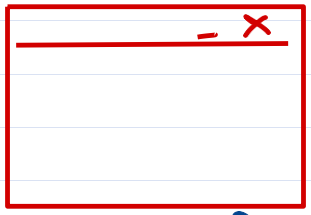
* HTTP is connectionless.
* HTTP is stateless.

http://server:port/appln/resource?k=v

↳ uri

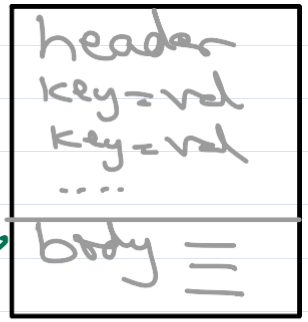
query string

browser



req

resp



* HTTP → Request Response model.

client $\underset{\text{resp}}{\overset{\text{req}}{\rightleftarrows}}$ server (pull).

resp

1. status → 2xx (success), 3xx (redir), 4xx (client error), 5xx (server error).
2. content type
3. content length
4. server info
5. cookie (also in req).

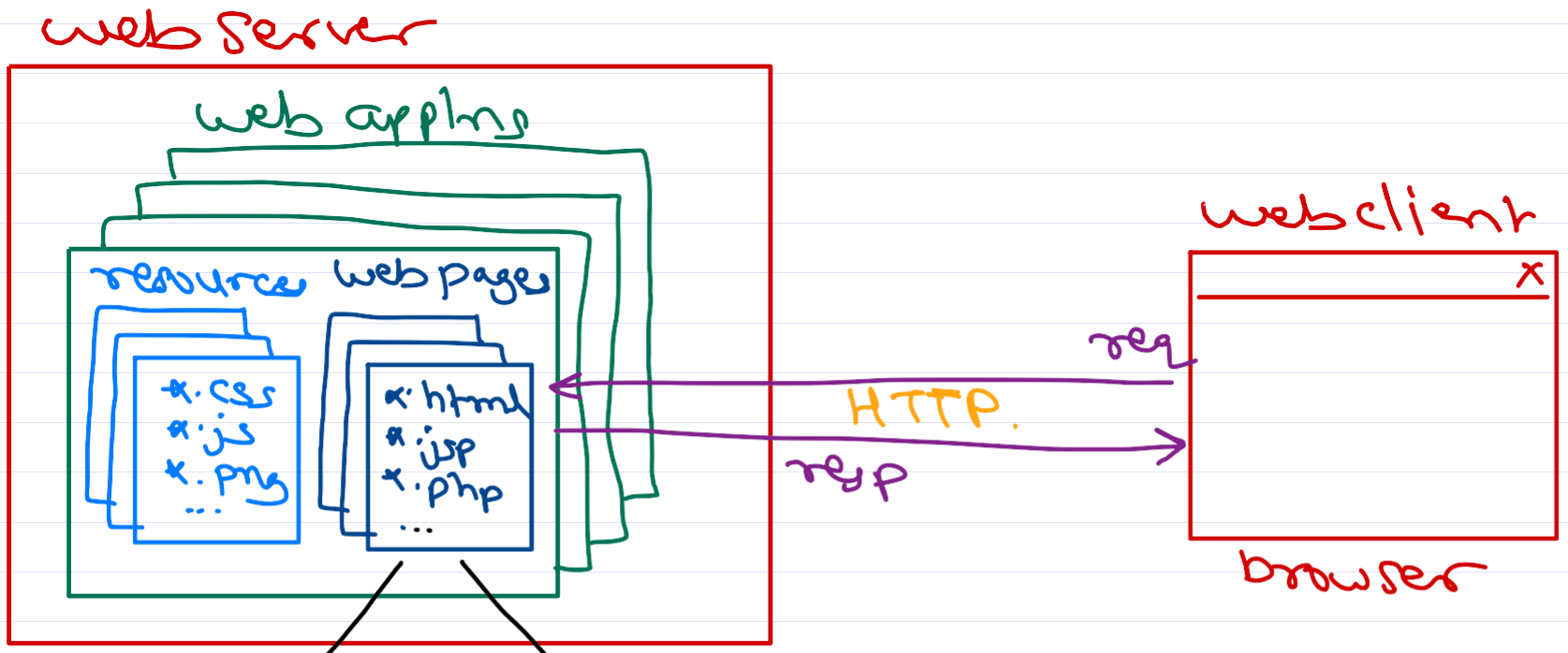
*req

1. content type → text/html*, text/xml, application/json*, image/jpeg, audio/mpeg, application/form-data*
2. content length
3. authorization ...
4. client info.
5. server+port + uri
6. req method → get, post, put, delete, head, trace, options

GET req: 1. browser url enter 2. hyperlink click 3. form get method 4. after redir.

POST req: 1. form post method

Web Server



GET

1. go to url in browser.
2. click on hyperlink.
3. http redirection
4. html form with method="get".

POST

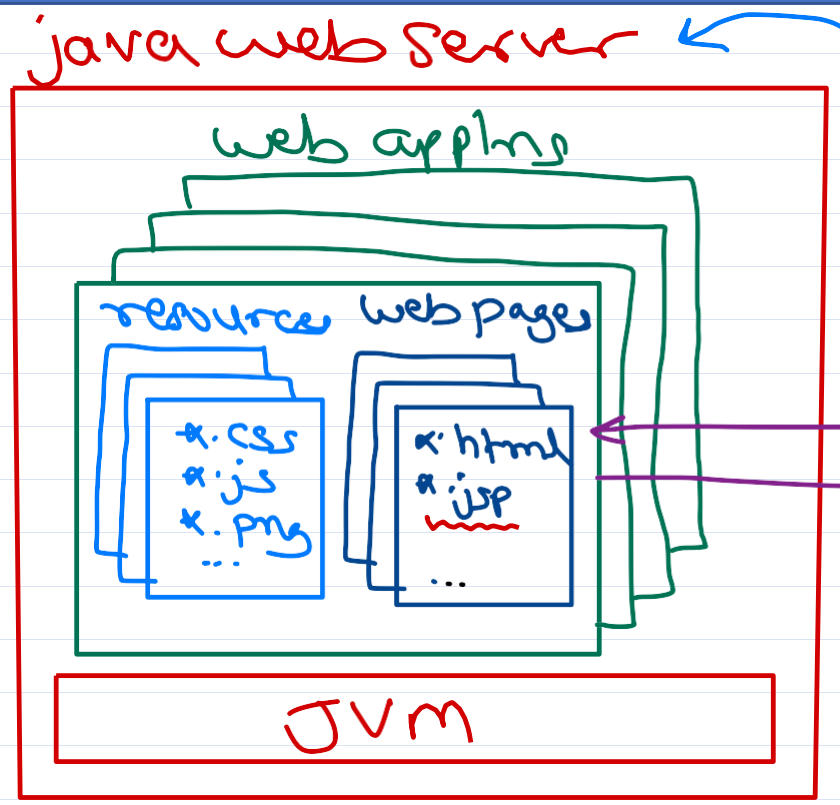
1. html form with method="post".

Supported by all web servers

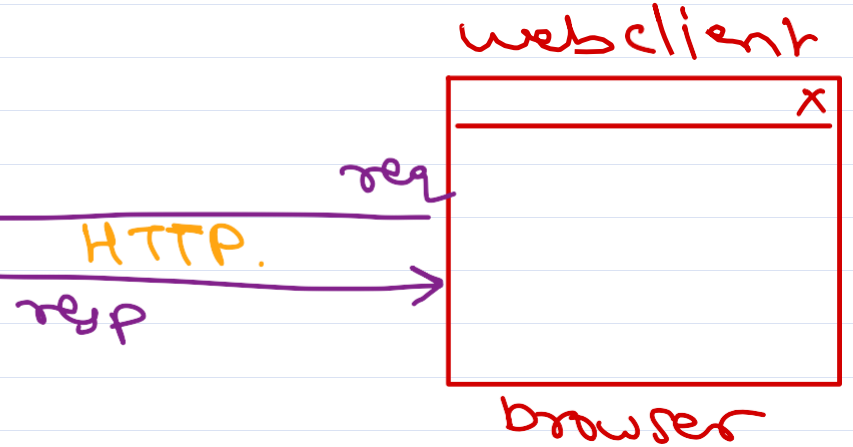
static pages	dynamic pages
*.htm	*.php → Apache (httpd)
*.html	*.jsp → tomcat, websphere, ...
	*.aspx → IIS
	...

e.g. IIS, Apache, Tomcat, JBoss, ...

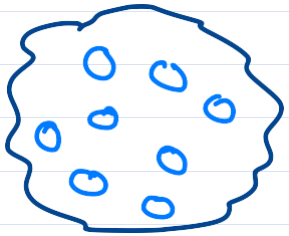
Java Web Server



standard/specification = set of rules
- given in form of interfaces, classes & ...
- e.g. JDBC, Java EE.



Java EE is set of specs → servlet + jsp + jsf + jpa + jndi + ejb + ...



④ ...
③ JNDI
② Connection pool
① SSL config

Java Web Server = web container + other services
↳ servlets + jsps.
e.g. Tomcat, Jetty, Lotus, ...

Java App Server = web container + other services + ejb container
↳ servlets + jsps. ↳ ejb
e.g. JBoss, web logic, websphere, ...
Glassfish (educational).