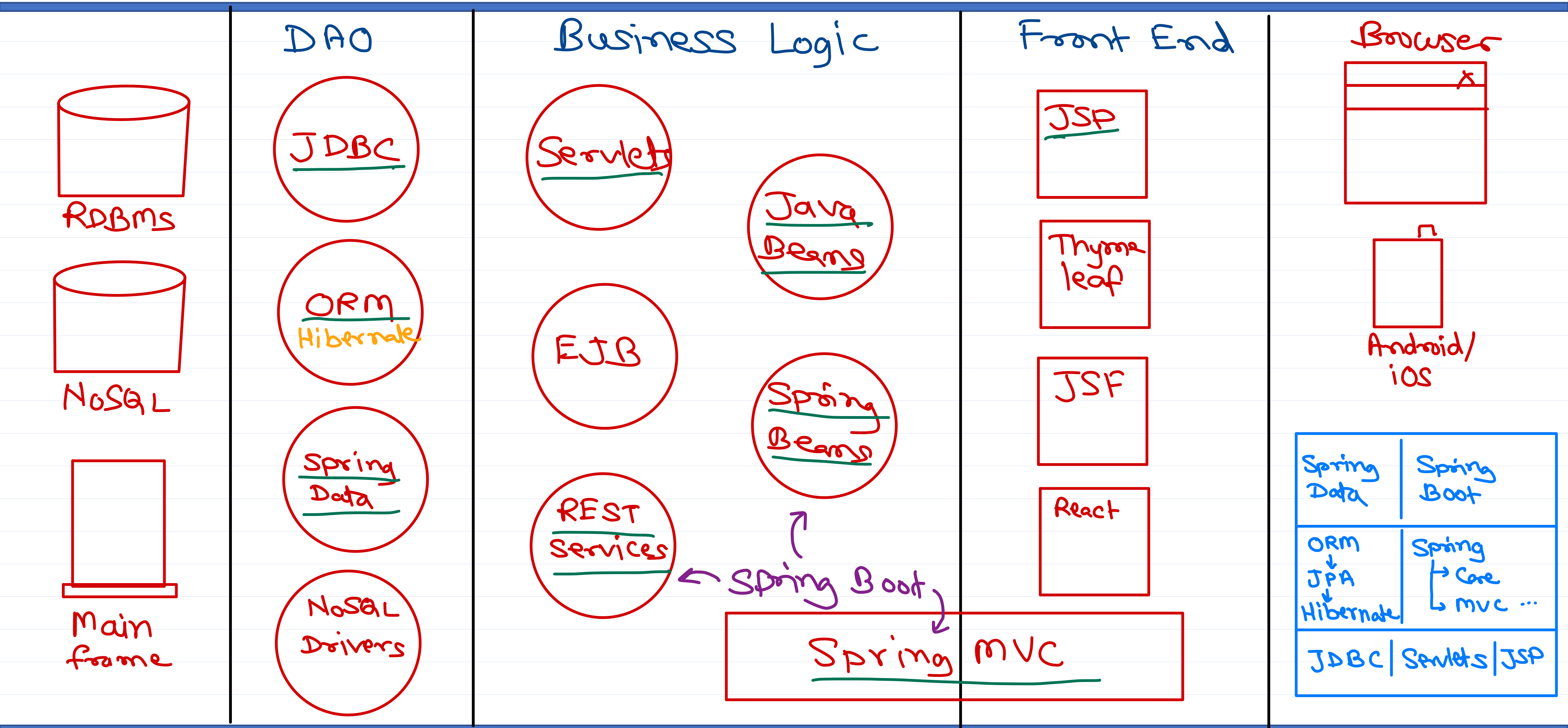
Advanced Java

Trainer: Nilesh Ghule

Java EE - Enterprise applications

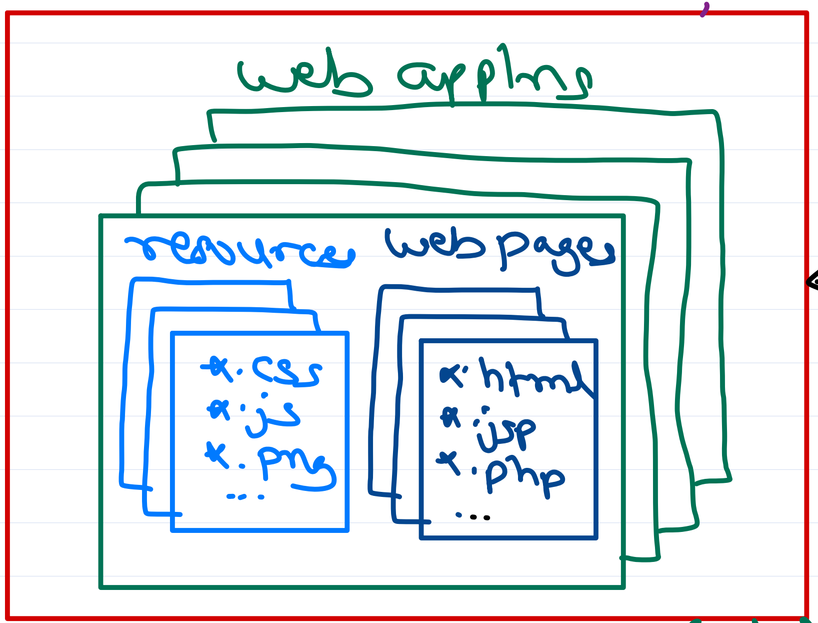
Java EE is set of specifications/standards.
e.g. JDBC, Servlet, JSP, JSF, EJB, JPA, ...

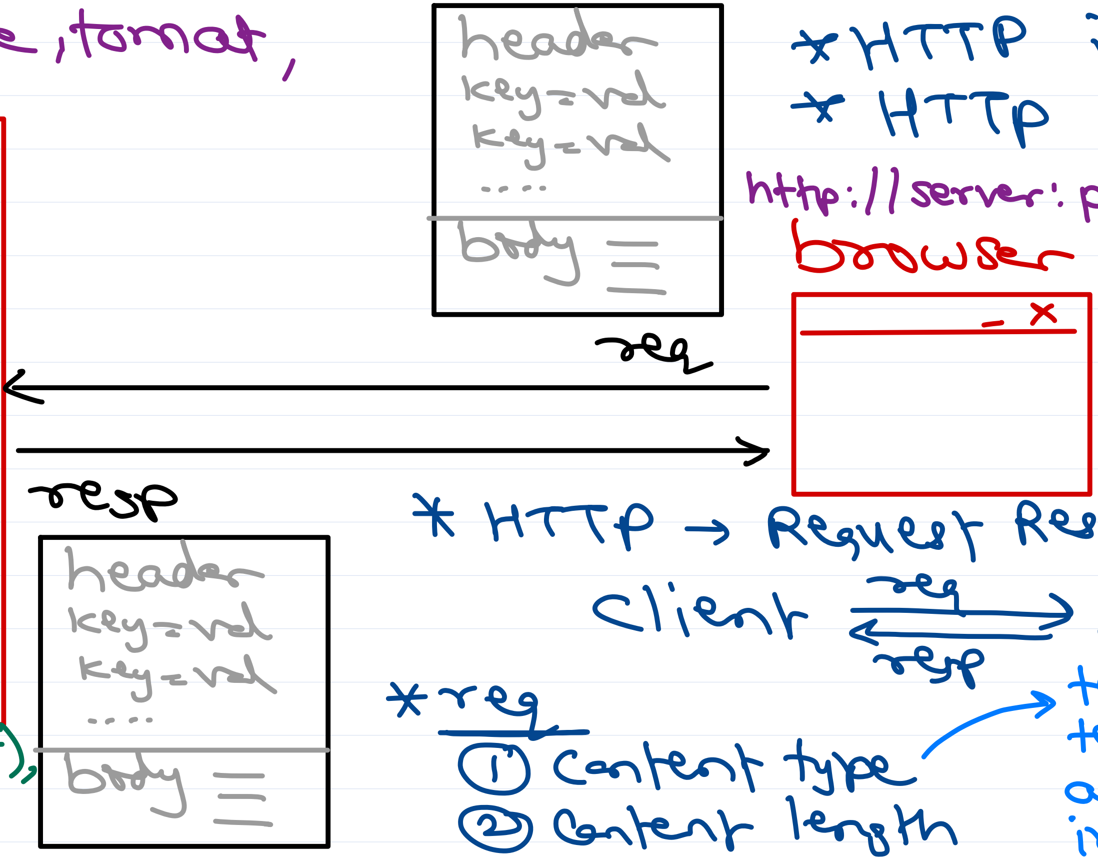


HTTP protocol

web server : web client | server : client
web appln
web page

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





* HTTP is connectionless.
* HTTP is stateless.

http://server:port$\underbrace{\text{/appln/resource}}_{\text{uri}}$?$\underbrace{\text{k=v}}_{\text{query string}}$

* HTTP → Request Response model (pull).

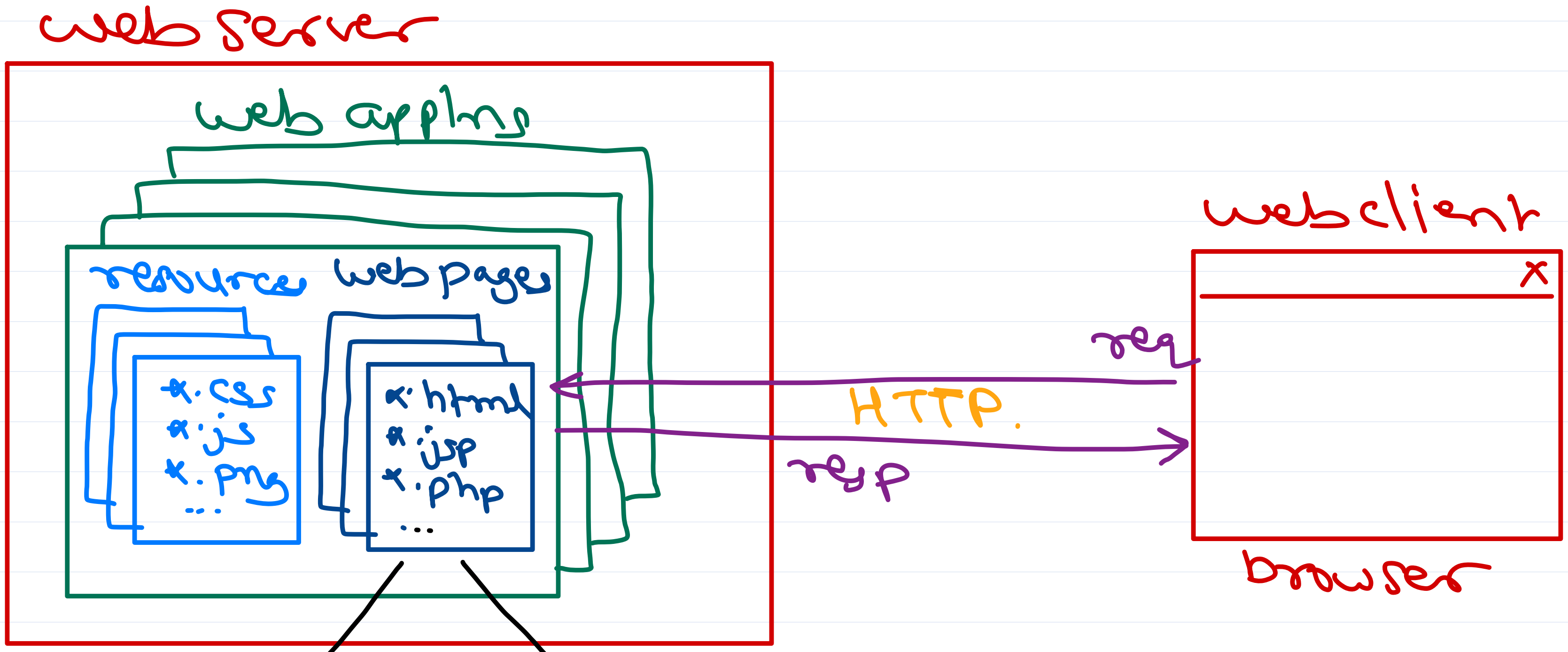
$\text{client} \underset{\text{resp}}{\overset{\text{req}}{\rightleftarrows}} \text{server}$

* 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

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).

Web Server



GET

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

POST

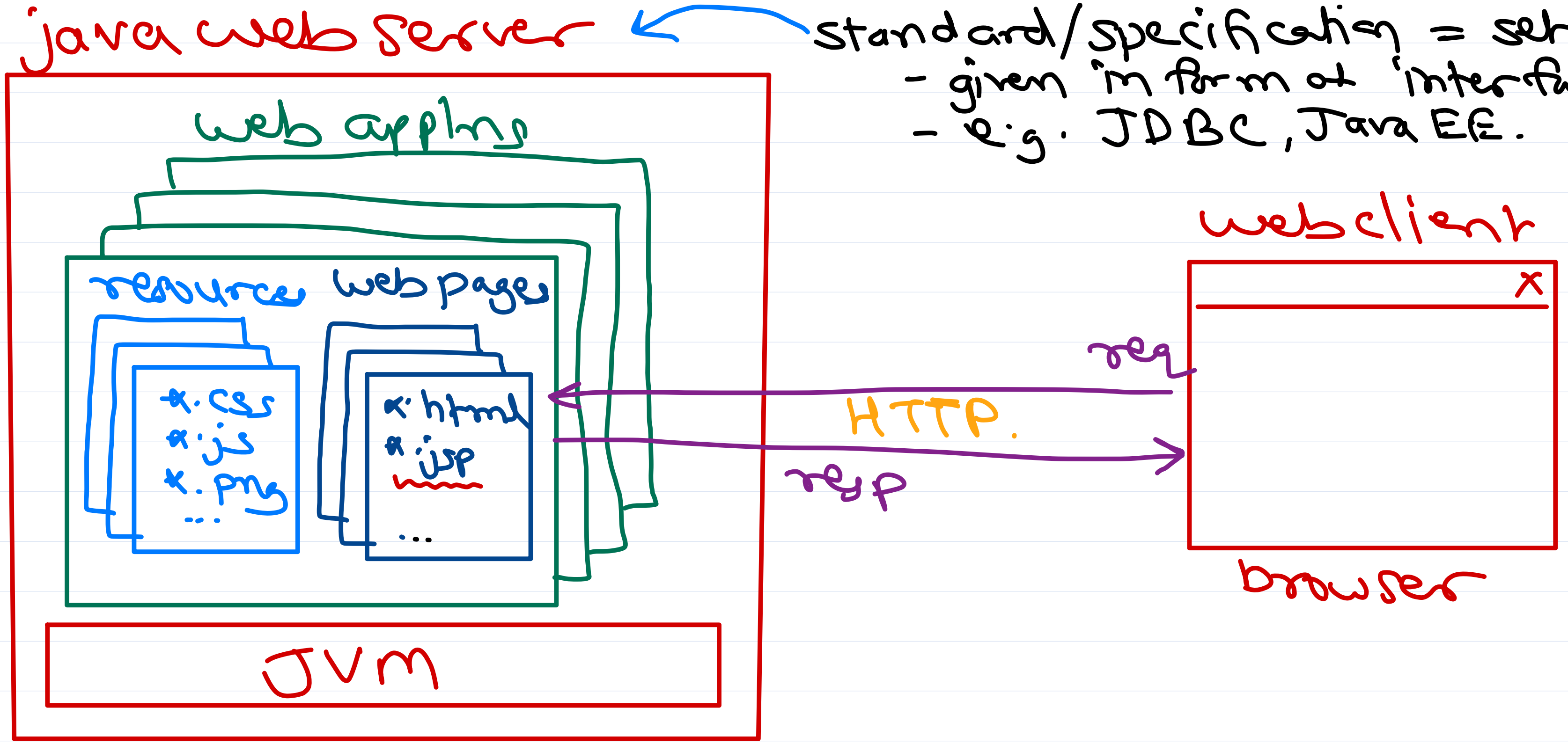
1. html form with method="post".

supported by all web servers { static pages
- *.htm
- *.html

dynamic pages
- *.php → apache (httpd)
- *.jsp → tomcat, web sphere, ...
- *.aspx → IIS
- ...

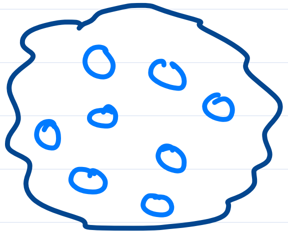
e.g. IIS, Apache, Tomcat, JBoss, ...

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).

Apache Tomcat

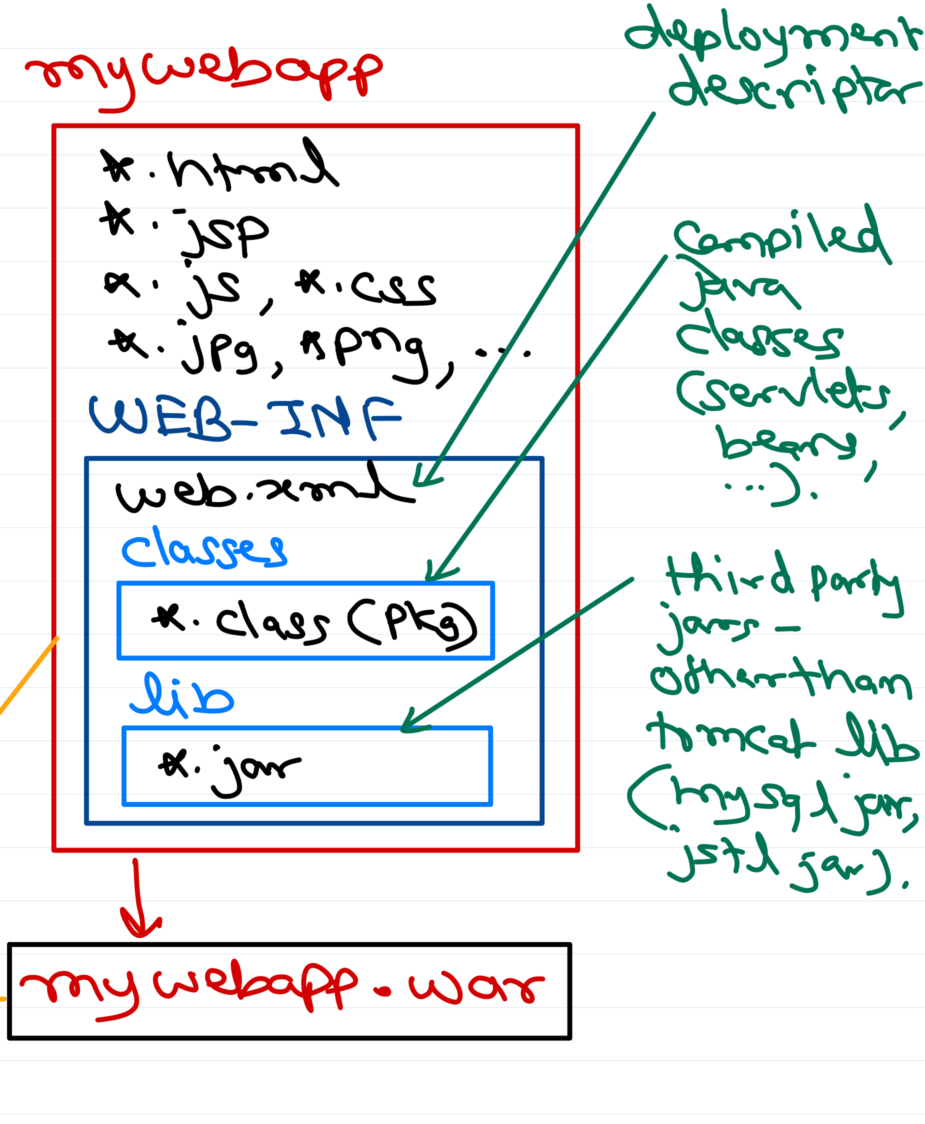
* To start/test tomcat on your system:

① download & extract tomcat.
② enviorment var → JAVA_HOME → path of jdk.
③ tomcat dir → bin → startup.bat ✓
④ browser → http://localhost:8080
⑤ tomcat dir → bin → shutdown.bat ×

* Tomcat directories

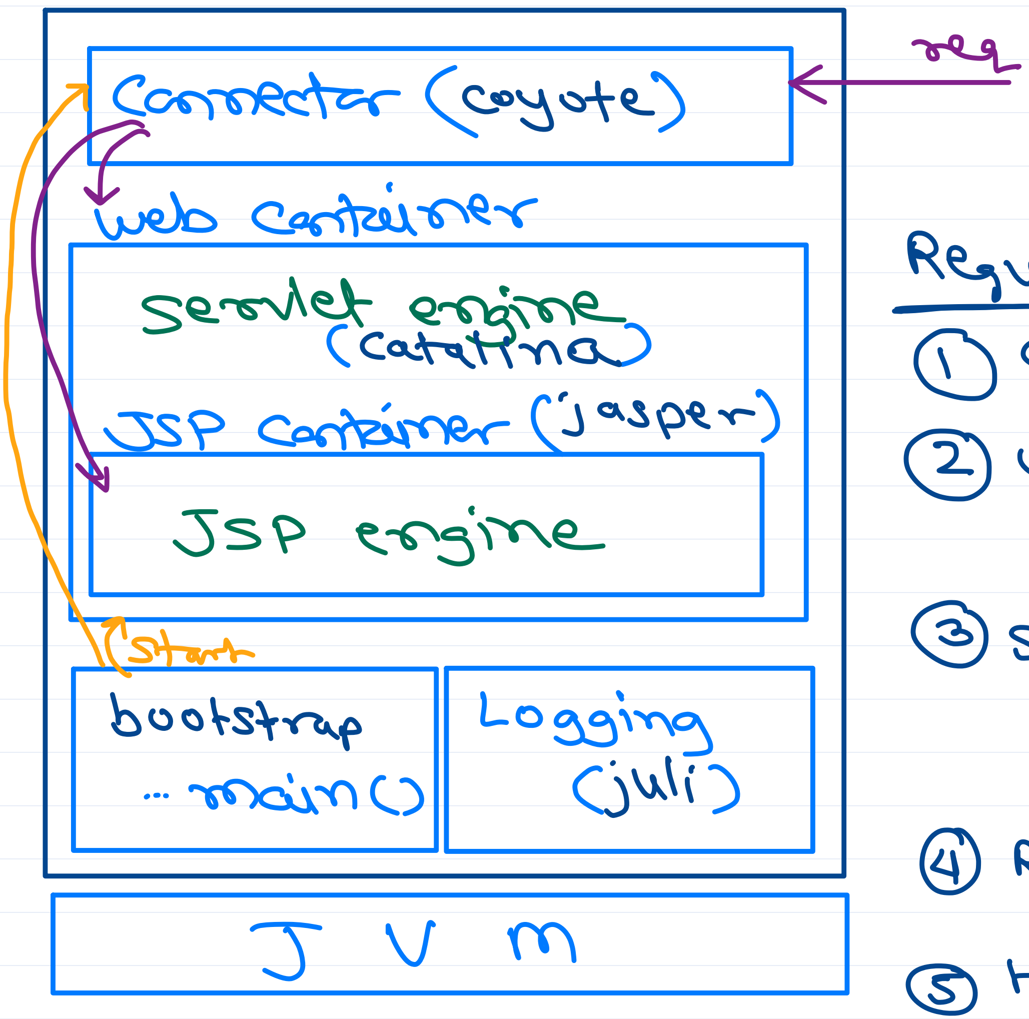
① bin : start/stop scripts, tomcat main executable.
② conf : config files of tomcat
  e.g. server.xml to set tomcat port (default: 8080)
③ lib : tomcat libs/jars (which implements java ee specs) e.g. servlet-api.jar, jsp-api.jar, ...
④ logs : web server execution logs (exceptions ...).
⑤ temp & work : temp files created at runtime for appln execution.
⑥ webapps : Hot deployment directory.
  - applns to be deployed are copied here
  - hot ⇒ applns copied while server is on are auto-deployed.

Java Web Application



Tomcat architecture

Tomcat architecture



Key parts:

1. Connector (accept reqs)
2. web Container (req processing)
3. web applns running in tomcat.

Request Processing

1. accept request
2. url mapping
   ↳ @WebServlet or web.xml
3. servlet object
   ↳ created on 1st req
   ↳ accessed from pool on next req
4. Response generation
   ↳ doXyz() method
5. HTTP response

Thread model

✓ tomcat creates a collection of threads during startup
- executor thread pool.

✓ on each req to a connector, a thread from pool is assigned to process that request.

✓ upon completion, thread returns back to pool.

✓ max num of threads in pool can be configured in server.xml.

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>