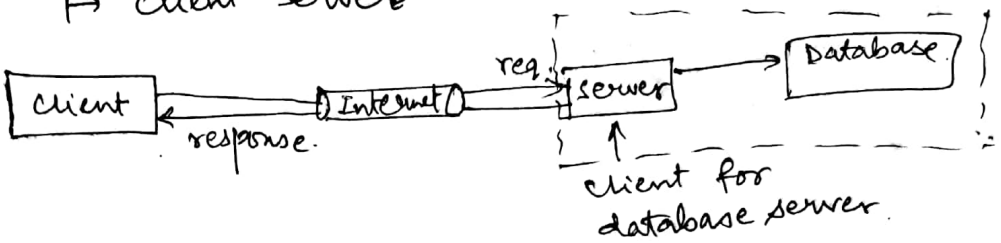
06/08/19 Miss

Internet Technologies.

- Web Applications (def:-)
- Design Patterns : Abstract design of software. → MVC
  It is a reusable solution to a design problem and it involves a set of components that interact to solve a general design problem within a particular context.
  - ↳ Architectural Design Patterns
    - ↳ client Server



- ↳ 3-tier, 6-tier, in-tier } Each tier addresses a separate concern.

presentation Application (logic) Data tier

- Each tier is independent. Separation of concern.
- Framework – Scaffolding of web application.
  A software application framework is a universal reusable software environment that provides particular functionality as part of a larger software platform to facilitate development of software applications, products and solutions.
  Framework → suitable if frozen spots >> hot spots
  - Frozen spots
  - Hot spots

* Web Applications :
WordPress – content management system
Web Application Framework : An application framework designed to support development of web applications that generally includes
→ Database support (schema change)
→ Templating framework for generating dynamic web content.

→ HTTP session management with middleware support
→ Built-in testing framework.

When a user is logged in session id is created.
Sends session id to client, → ① time
→ ② security services
→ ③ Data.

session id stored in browser.

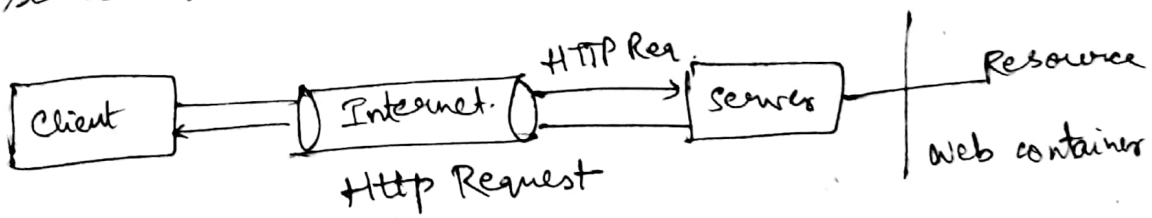
When next request goes to server browser sends session id,

Intelligent web.

1. Ruby on Rails (Web Framework).
   → Sinatra
2. React
   Angular JS [Node JS]
3. ~~Symphony~~ Symfony (PHP).
4. Play (Java/scala).
5. Django (Python).
6. Spring (Java/JSP/servlet)
7. ~~Hibernate~~. ASP.NET

*HTTP:

- Gives unified interface at server side for resource/services
- Reuse web infrastructure.
- Efficient in distributing requests to different servers.



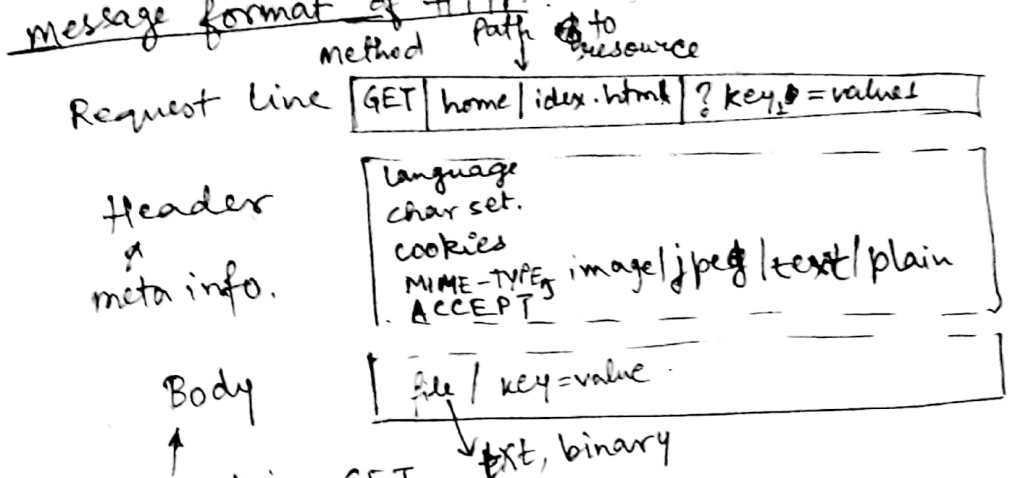
Http Request

method (type of req.)

Idempotent → GET
Not naturally idempotent → POST → create new file
Idempotent → PUT → server will store → update file
Idempotent → DELETE

path + resource → container to actual filepath.

Message format of HTTP:



URL: Way resources are identified in the server side

http://<hostname : port no> / <path + resource> ? key_1 = $value_1$ & key_2 = $value_2$ ...

MIME Type: Type on server may not be same as type on client. It is an identifier for a particular type or format of information.

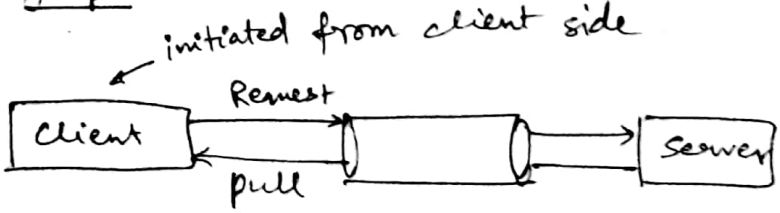
Instead of ~~types~~ MIME-TYPE, 'ACCEPT' is used.

Formats clients will accept.

For files multi-part is used. Part-separator.

- <boundary> - -
- <boundary> - -

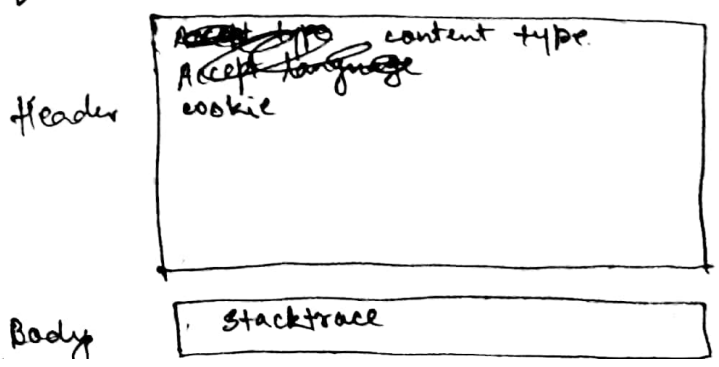
13/08/19 ~~Http~~



~~Http~~ response

status line: http/1.1 / <status code> <phrase> — 200 OK

req. line: method /< resource>



status code
- → 1xx – information, continuing process.
- → 2xx – successful
- → 3xx – redirection
- → 4xx – client side error
- → 5xx – server side error.

Server → web container → Servlet → Java Class POJO ← domain model

Content type is significant if status is OK.

web services | WSDL ← Web Service defn. Language

SOAP ↑ Simple Object Access Protocol. ↑ object

REST ← HTTP

REST ↓ Representational State transfer.

→ Treats objects as resource

PUT POST DELETE GET URL

① Interfacing HTTP
② Encoding of parameters.

http:// www.example.com :80/VideoSharing/ ~~all videos~~ video18 (GET) ← video1/duration.

List of videos. PUT ← identifier already given

http ...... :80/VideoSharing/video1

:80/VideoSharing/ (POST) upload all videos.

DELETE delete video.
:80/ ——

4 mechanisms for server to let client know about server update.

1. Event driven. ← Let client sync manually – when clientside window is opened sync.
2. ~~Event driven~~
3. Periodically – http polling update every 3s. client sends request.

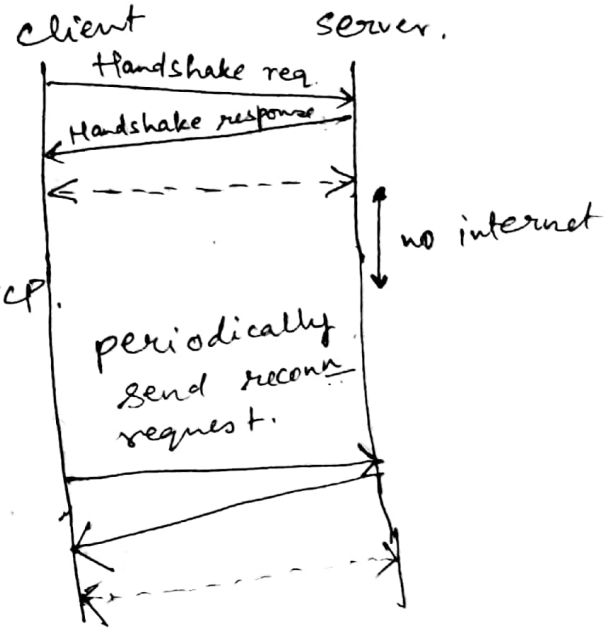
Binary Exponential backoff. Periodically. Adaptive Polling

4. Web socket ⟷ Http

① Persistence
② Bidirectional
③ Secure

→ push packets by server
→ Does not close the TCP.
→ security key.

GET /chat HTTP/1.1
Host: . . . . .
Upgrade: Web Socket
Connection: upgrade
Web-Socket key: security key.



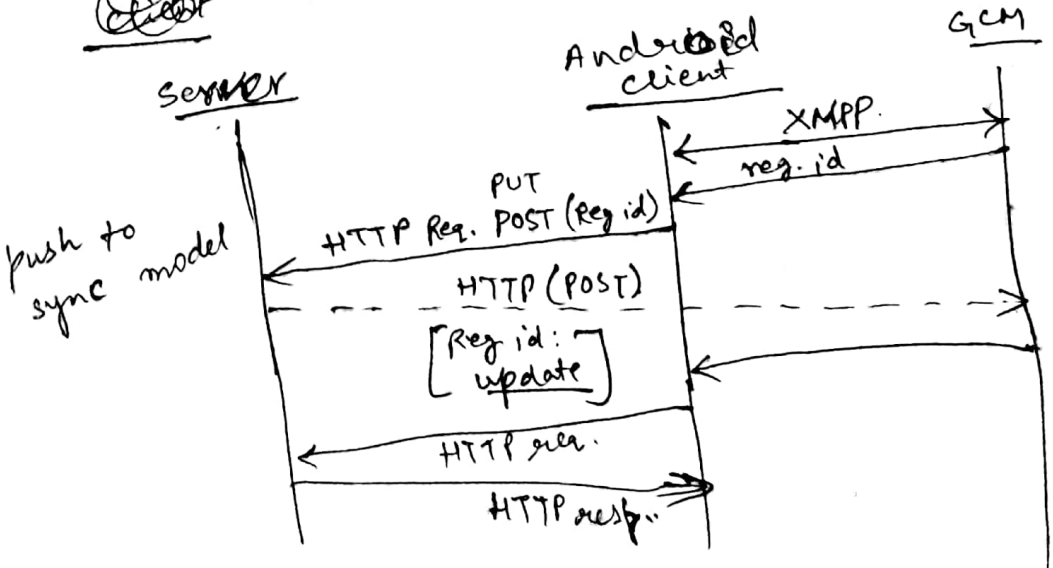
synchronization is difficult with bidirectional connection.

HTTP 1.1 | 101 switching protocol.
connection: Web Socket
upgrade: Web Socket
Sec-Web Socket accept: < >

Web-Socket → 80 Http.
Web-socket Security → 443 Https.

Server has to close the connection specifically. for web socket.

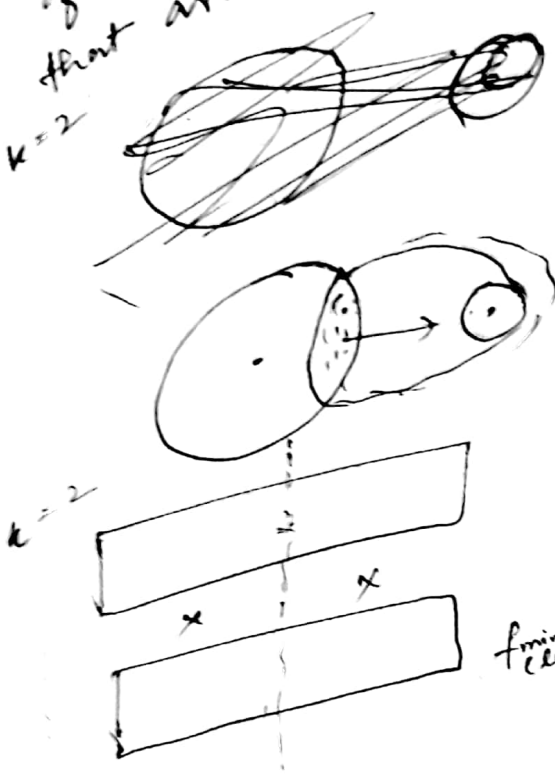
① < 4KB → send to GCM
② Security

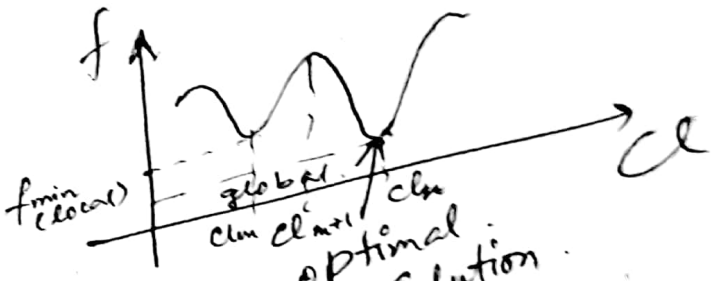


K-means is an efficient ... where data points ... of the clusters exhibit ... that are relatively far from ...

Non-overlapping

K-means algorithm can stop ... at local minimum which is not necessarily the global one





Internet Tech.

27/09/19

Deployment descriptor

<web-app .... >

<servlet>
<servlet-name> HelloWorldExample </servlet-name>
<servlet-class> Hello... </servlet-class>
</servlet>

<servlet-mapping>
<servlet-name> HelloWorldExample </servlet-name>
<url-pattern> /Hello/HelloWorld.do </url-pattern>
</servlet-mapping>

↖ .class file.

↑ what to show the client

</webapp>

1. FQ class name 2. URL name.
3. Deployer defined name.

1. Declarative security → can also specify which methods are accessible.

We can also write attributes to request object.

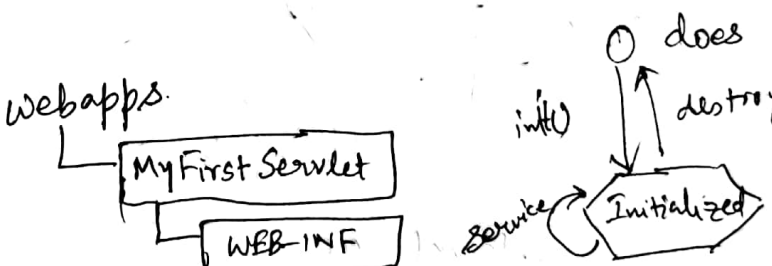
↓ Extracting parameters.
↕ Error checking.
↕ Conversion.
↕ Business Logic

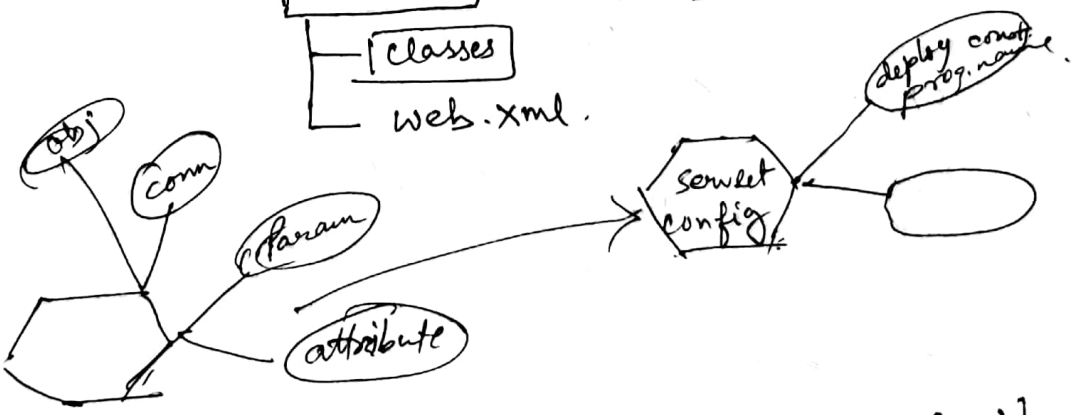
Web Container

- Communications support
- Lifecycle Management
- JSP Support.
- Multithreading Support
- Declarative Security

Scope
request, session, context / application.
↓ ↓ ↓
parameter, attribute attribute attribute.

D3 09/19





System.out.println(getServletConfig().getInitParameter(---));

Generic Servlet → init(), service(), destroy().

↓

Http Servlet

↓

MyFirstServlet

init() may be initialised if we need runtime objects.
Ex. db connection.
(say JDBC driver)

public void doGet (HttpServletRequest req, HttpServletResponse resp)

- ServletContext
- Listener. → Listening to an event.
  ↳ notification for event.

~~data~~

ServletContextListener.

```
class MyServletContextListener implements ServletContext
{
    p. void contextInitialized (servletContextRequest e)
    {
        Database obj = new Database (~~dbstring, dbvalue~~)
                    (e.getServletContext.(~~dbstring~~)
                    getInitParameter('dbstring'));
        e.getServletContext().set Attribute("con$",
                    obj);
    }
}
```

If servlet is changed.
response.sendRedirect("————") ← client side change. and again request.

```
View = request.getRequestDispatcher ("x.jsp");
View.forward(req, resp);
```

To send jar file:

```
p. v. doGet(.. req,.. resp){
    resp.setContentType("application/jar");
    ServletContext ctx = get ServletContext();
    InputStream is = ctx.getResourceStream("/xyz.jar");
    int read = 0;
    byte[] bytes = new bytes [1024];
    OutputStream os = response.getOutputStream();
    while(- — -) {
        os.write(- -);
    }
```

alised objects

}

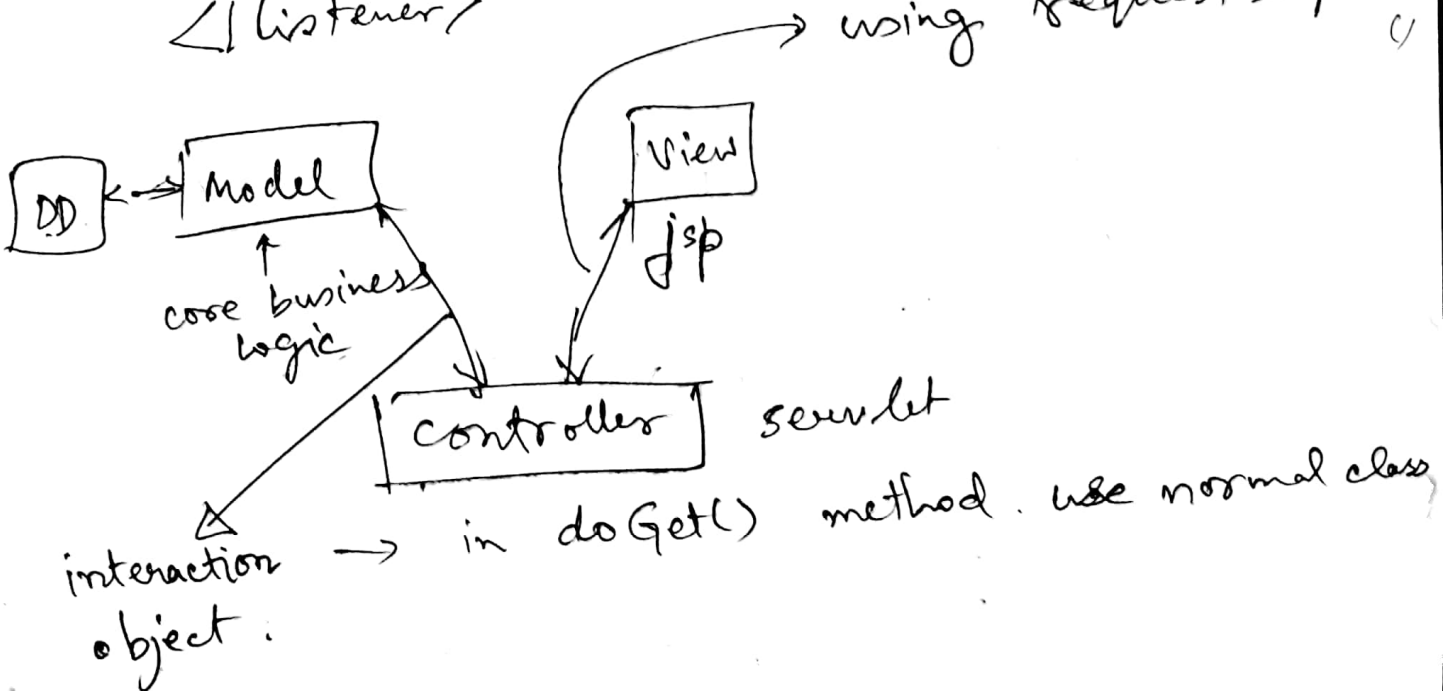
```
    os.flush();
    os.close();
}
```

Use listener to count no. of users accessing the servlet.

Container starts working

1. Loads servlet context
2. " " Listeners.
3. ~~Both (constructor) for servlets constructor~~
3. Servlet Config.
4. Configuration parameters.
5. Servlet instance
6. servlets.

```
<listener>
<listener-class> - - </!- - >
</!listener>
```



17/09/19

IT

Customised session

HTTP. as HTTP is stateless.

TCP

IP

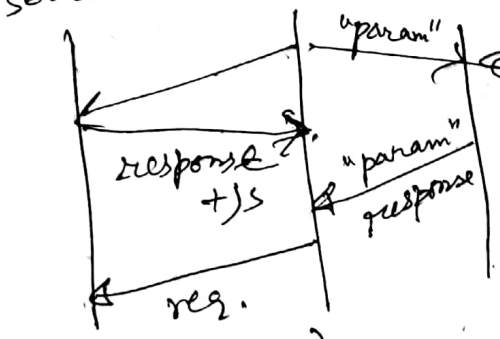
Session object is shared by the same client.
session ID cookie is created for every session object created is sent back to the client

cookie — name string
value string

alive for only a time interval.

Use both URL rewriting and cookies.

HTTP Session Listener.



Injection attack.

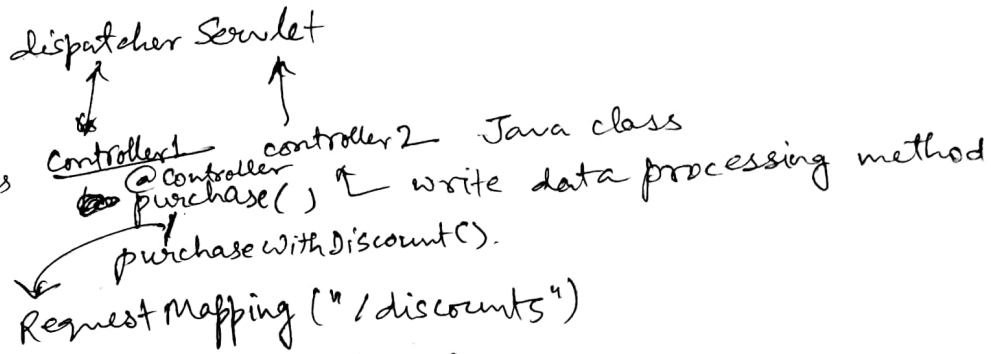
Servlet:

1. Extraction of parameters
2. error checking.
3. data type conversion.

overhead.

4. data processing.
5. writing the response.

Spring framework : does 1, 2, 3.



JSP has findAttribute(—).