AJAX: asynchronous javascript and xml

Based on any event u can make an asynchronous call to the server an get back the data without having to reload the XmlHttpRequest

Javascript:

Event propagation:

<div onclick=”alert(‘div clicked!’)”><h2>This is a heading></h2>

<input type=”button” onclick=”alert(‘hello’);”>

</div>

$(document).ready(function() {

Servlets:

Dynamic web pages run on web container

Web server: Tomcat, wildfly, Jetty

Http Server+ web container. Web container can manage servlets

1. Lifecycle of a servlet (init, destroy, service), object of servlet
2. Dynamic parameter passing (config, context)

Http Server:

Storing static pages. (html pages). Request comes in for a page, just returns back the page as the response.

Application Server: Glassfish, Weblogic, Websphere

It is more advanced compared to your web server and was used mostly for EJB applications

1. Configure JMS objects over there (Inbuilt broker)
2. Configure connection Pool over here (database details)

Application becomes dependent on Application server.

Limitation is lot of manual configuration to be done, if application has to be deployed on another application server

Q) What are the ways of creating a servlet?

a) Servlet interface

1) init

2) service

3) destroy

4) getConfig()

5) getServletInfo(): return string giving some info about ur servlet

b) Generic Servlet: abstract class with 1 abstract method i.e. service

c) HttpServlet: abstract class with no abstract method.

Service method implementation is provided by default and it is internally calling your doGet, doPost, doPut, doHead, doOptions, doPatch and other methods based on the request method.

Q) Can a servlet return json data? How?

response.setContentType(“application/json”);

response.println(employee);

Gson/ Jackson/jaxb (included in jdk: 6,7,8) on build path to serialize it

Q) Is HTTP a stateless protocol?

Yes

Q) How do you create sessions?

a) cookie:

Request -----(username, password )-----🡪 server

Create cookies to store this info

Cookies stored on ur browser<--------------(cookie)Response <--------------

When another req is made from ur browser to the server this cookie is sent back to the server, hence server can get this info with other requests. Hence session is created.

Login page: username, password ------🡪 server (store it in form of cookies)-🡪 sends back response as login successful page

If u make another request to get employee information, since cookies are being sent automatically, no need to pass this username, password again with ur request.

Cookies -> expiration date.

Privacy, browser if cookies are disables, no session can be obtained

b) session object: Most preferred way

c) url rewriting: passing the client information in the url

d) hidden form fields: form, u would have to add an extra hidden field and manually would have to set the value on every page

Q) If u r using session object, how do u implement the functionality of logout?

session.invalidate()

Q) Diff between config and context parameters?

A) Context parameters (context-param) are set for all the servlets whereas config params (init-param) belongs to a particular servlet

Q) What are filters?

A) Filter is an interface . 3 methods: init, doFilter, destroy. Generally used for service code (logging, transaction, security, performance…)

Class AuthenticationFilter implements Filter{

Void doFilter(FilterChain chain, request, response){

String username=request.getParamter(“username”);

//password

//If user exist in db: then request should be forwarded else dropped{

Chain.doFilter(req,resp); //some other filter/ servlet

//This would be executed when we get back the response

}}

Req🡪 filter1 -> filter2 -🡪 servlet -> filter2 -> filter1 -> response

Listener:

Listening to the events

ServletContextListener: context is created

ServletConfigListener

}

Q) Are servlets thread-safe?

Class MyServlet extends HttpServlet{

Int x;

Service(req, resp){ int y=5; y++;

Synchronized(this){

X++;

}

}

}