Servlet-api

1. HttpSessionTracking
2. CookieSessionTracking
3. URLReWriting Tracking

HttpSessionTracking

1. Session id will stored in the form of cookie and in client side if browser disables the cookie HttpSessionTracking mechanism won't work.
2. Maintainence of Session object is at the server side which would be burden to the server.

eg: jSESSIONID = ...DBEFH

CookieSessionTracking

1. Cookies will be stored in the client side and if browser disables the cookies then we can't keep track of client data.
2. Since cookies are used to store the client data,it might result in "Security Breach".

URLRe-Writing Mechanism

=> This mechanism is same as "HttpSessionTracking" mechanism,but sessionid won't be stored inside cookie rather the sessionId will be appended to the url everytime when the request-response happens b/w client and server.

```
eg: <form method ="get" action='"+response.encodeURL('./second')
+"'>
        </form>
```

./second?JSESSIONID = ...

Note: In Realtime project we don't use technologies directly,we use framework to improve the productivity,so by default Framework support SessionTracking Mechanism through "URL-ReWriting" only.

Invented by Developer

1. HiddenFormField

Not used in realtime as it increases the lines of code.

URLPatternTypes

As per Servlet specification, we have 4 different ways of mapping the URLPattern

a. Exact match URL Pattern => eg: /test
b. Longest Path URL Pattern => eg: /controller/servlet/*
c. URL Pattern by extension => eg: *.do
d. Default URL Pattern => eg: /

Which of the following are valid url patterns?

1. /test (valid)
2. /test/*/test(invalid)
3. /test/test/*(valid)
4. *.test(valid)
5. /(valid)

6. /test/test/*.do (valid)

Patterns for Servlet
================
/   => FourthServlet
/test   => FirstServlet
/test/test/* => SecondServlet
*.do   => ThirdServlet

http://localhost:9999/URLPatternTypesApp/test/test/navindReddy.do => SS
http://localhost:9999/URLPatternTypesApp/test/test/navinReddy   => SS
http://localhost:9999/URLPatternTypesApp/
=> DS
http://localhost:9999/URLPatternTypesApp/navinReddy.do   => TS
http://localhost:9999/URLPatternTypesApp/test/hyder   =>
DS

Note: When none of the Servlet is getting mapped, then default Servlet will get a chance once again.
Webcontainer will always gives preference in the following order
      a. Exact match UrlPattern
      b. Longest Path prefix UrlPattern
      c. UrlPattern by Extension
      d. Default UrlPattern

In realtime projects which type of url pattern is preferred?
      URLPattern by extension is prefered(*.do).
      In SpringMVC for inbuilt servlet called "DispatcherServlet(FC)", we configure it through "URL-pattern by extension".

Configuring welcome pages
=======================
  It is highly recomended to configure the welcome-page/landingpage for our webapplication.
  Advantage
    1. It increses the easyness of the use of the webapplication for the end user.

Configuration can be done in XML only for html files, jsp files
==
```
<web-app>
      <welcome-file-list>
            <welcome-file>home.jsp</welcome-file>
            <welcome-file>welcome.jsp</welcome-file>
            <welcome-file>index.jsp</welcome-file>
      </welcome-file-list>
</web-app>
```

Note:
1. The order of searching a landingpage would be from top to bottom
2. In any webapplication index.html acts as a default welcome file,if index.html is not avaialable then index.jsp acts a default
    welcome file.
eg:
  http://localhost:9999/FirstApp/ ===========> search for index.html/index.jsp and load as the response to the client.

As per JEE specification, when we are configuring jsp pages inside <welcome-file>, we need to just specify the file name not

with "/".

eg: <welcome-file>index.jsp</welcome-file>(valid)
<welcome-file>/index.jsp</welcome-file>(error)