Mindtree

*Welcome to possible*

# Session Handling

© Mindtree limited 2012

# Objectives

- Define Stateless and stateful web apps
- Define Session tracking techniques
- HttpSession
- Adding, removing, reading from session
- Session expiry

# HTTP Protocol

- HTTP Protocol is a stateless protocol which means that each page request is considered independent of any other request.
  - A web server does not understand if a page request comes from someone who has already requested a page or if the person is visiting the page for the first time.
  - It treats each request in the same way.

- In some scenarios, e.g. ecommerce and banking web applications this is inconvenient and we need some way to tie every page request together as a session of a single visitor.

# Session Tracking

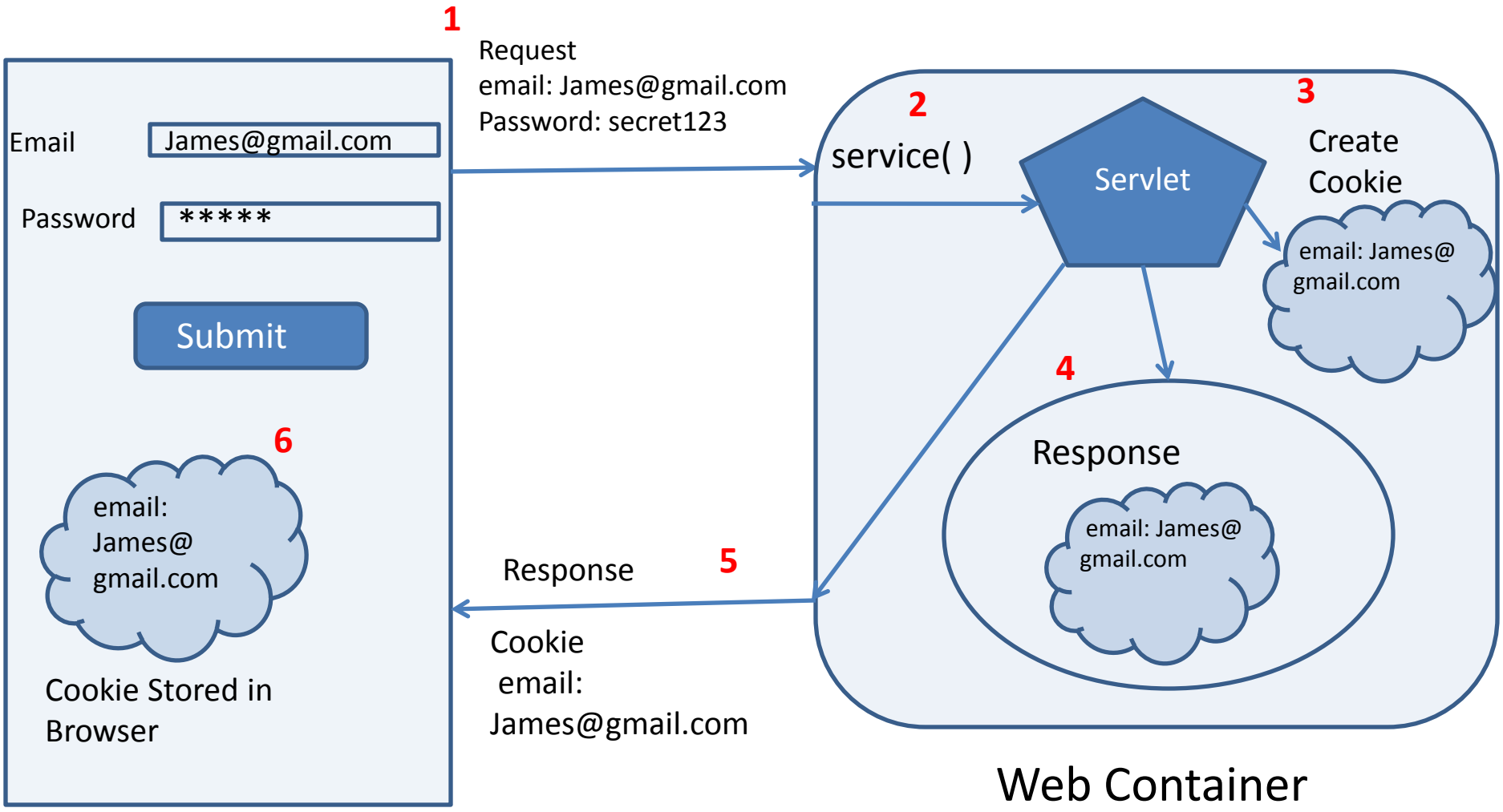
- Session tracking is a mechanism which helps the servers to maintain the conversational state of a client.

- Different mechanisms that can be used for session tracking:
  - Hidden Fields
  - URL Rewriting
  - Cookies
  - HttpSession API.

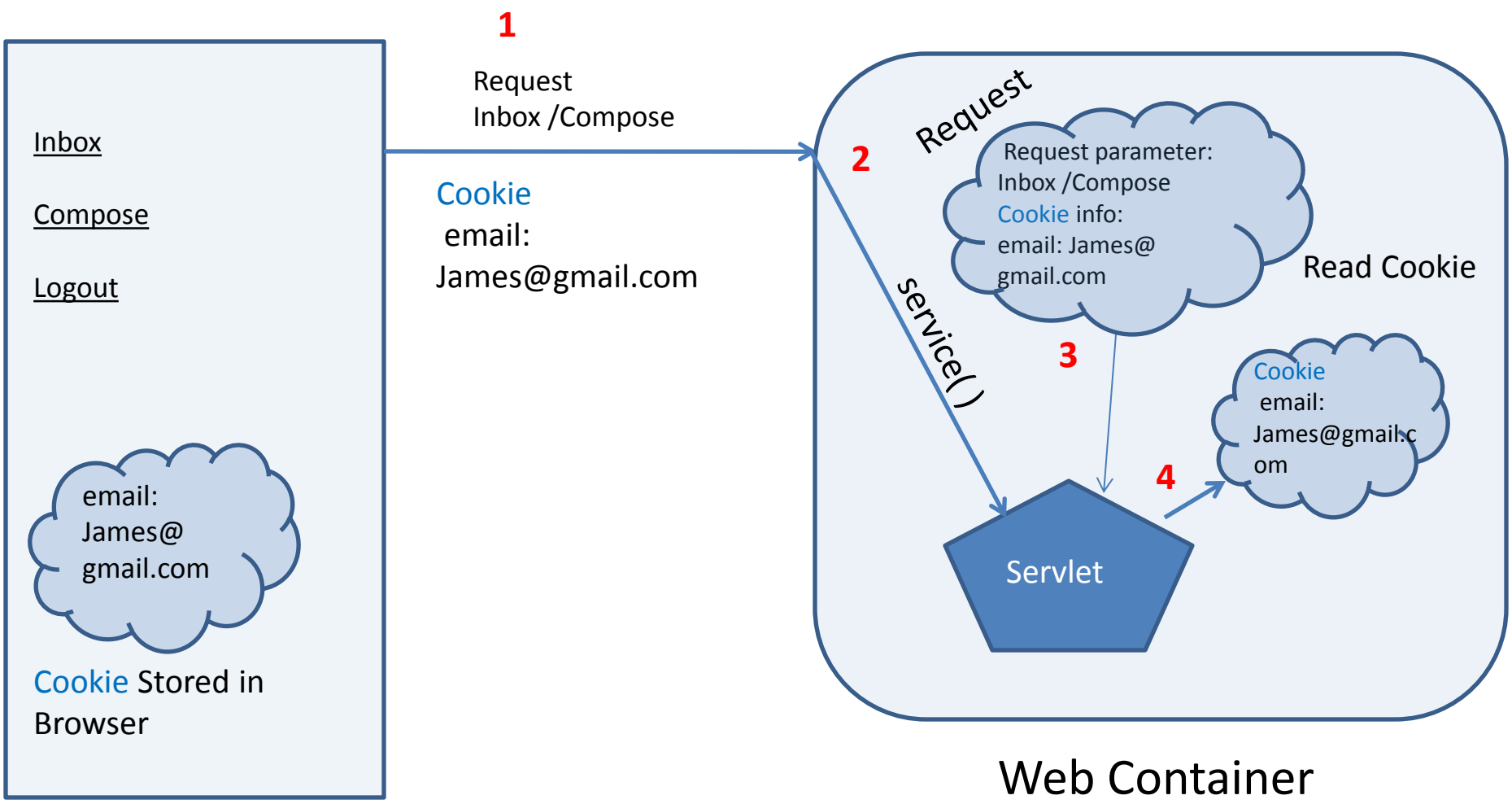
# Cookie

- Cookies can be used for session tracking.
- A *cookie* is a bit of information sent by a web server to a browser that can later be read back from that browser.
- Browser receives a cookie and it saves the cookie.
- Browser sends the cookie back to the server each time it accesses a page on that server

# How cookies work?



# How cookies work?



# Cookie and Servlet API

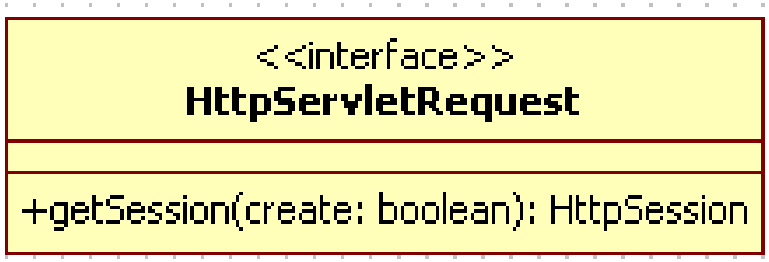
- Creating a Cookie using constructor:
  - public Cookie(String name, String value)
  - This creates a new cookie with an initial name and value

- Sending a cookie to the client:
  - public void addCookie(Cookie cookie) method of HttpServletResponse

- Retrieving all cookies sent by the web browser:
  - public Cookie[ ] getCookies() method of HttpServletRequest.

# HttpSession API

- HttpSession interface is used by the Web Container to create a session between the HTTP server and an HTTP client.
- A Servlet uses its request object's getSession() method to retrieve the current HttpSession object:

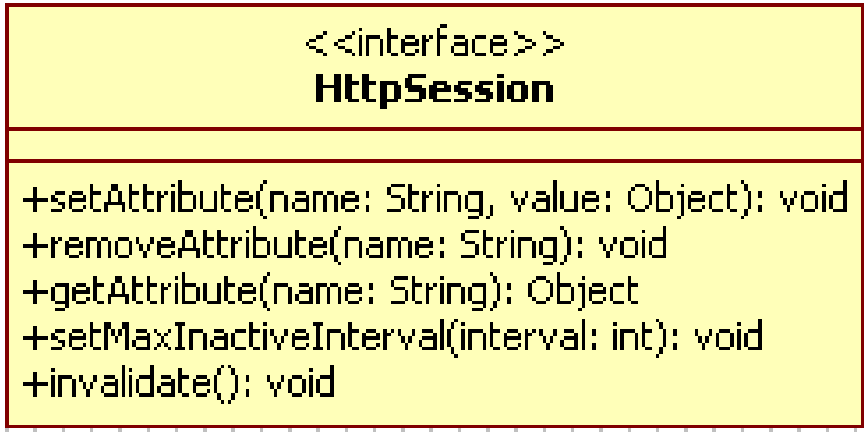



This method returns the current session associated with the user making the request.

If the user has no current valid session, this method creates one if **create** is true or returns null if **create** is false.

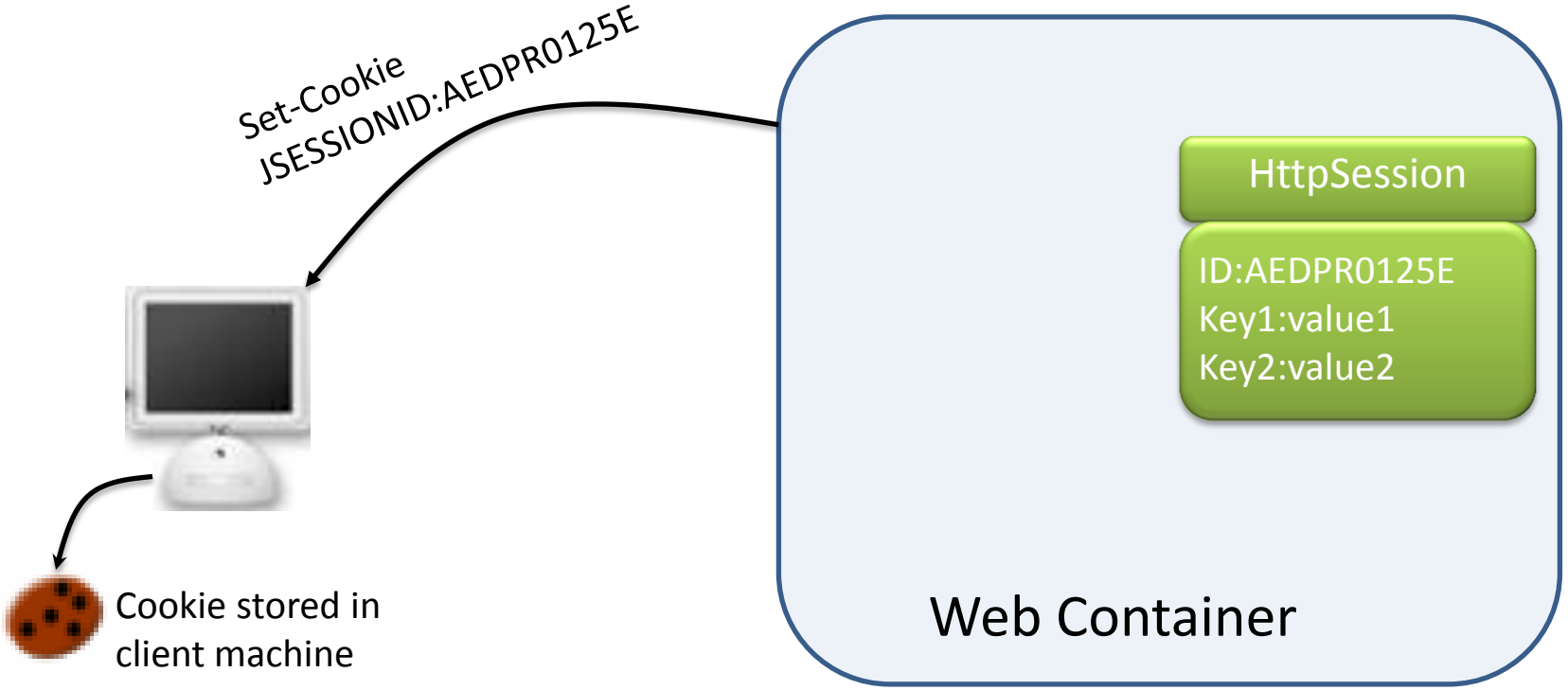
# HttpSession API

- void **setAttribute**(String name, Object value)
  - Binds an object to this session, using the name specified.
- void **removeAttribute**(String name)
  Removes the object bound with the specified name from this session.
- Object **getAttribute**(String name)
  Returns the object bound with the specified name in this session, or null if no object is bound under the name.
- void **setMaxInactiveInterval**(int interval)
  Specifies the time, in seconds, between client requests before the servlet container will invalidate this session



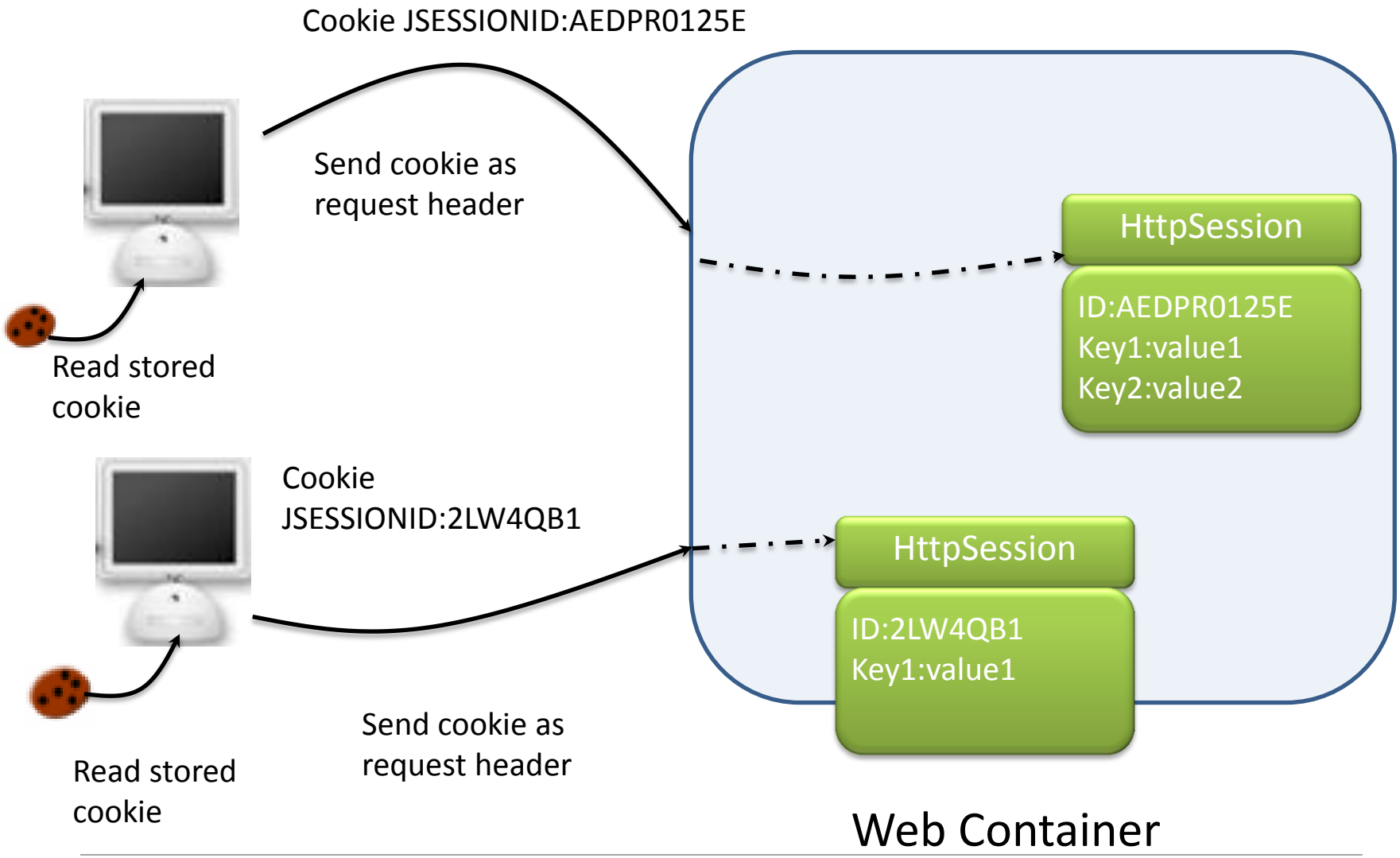

- void **invalidate**()
  Invalidates this session then unbinds any objects bound to it.

# HttpSession API

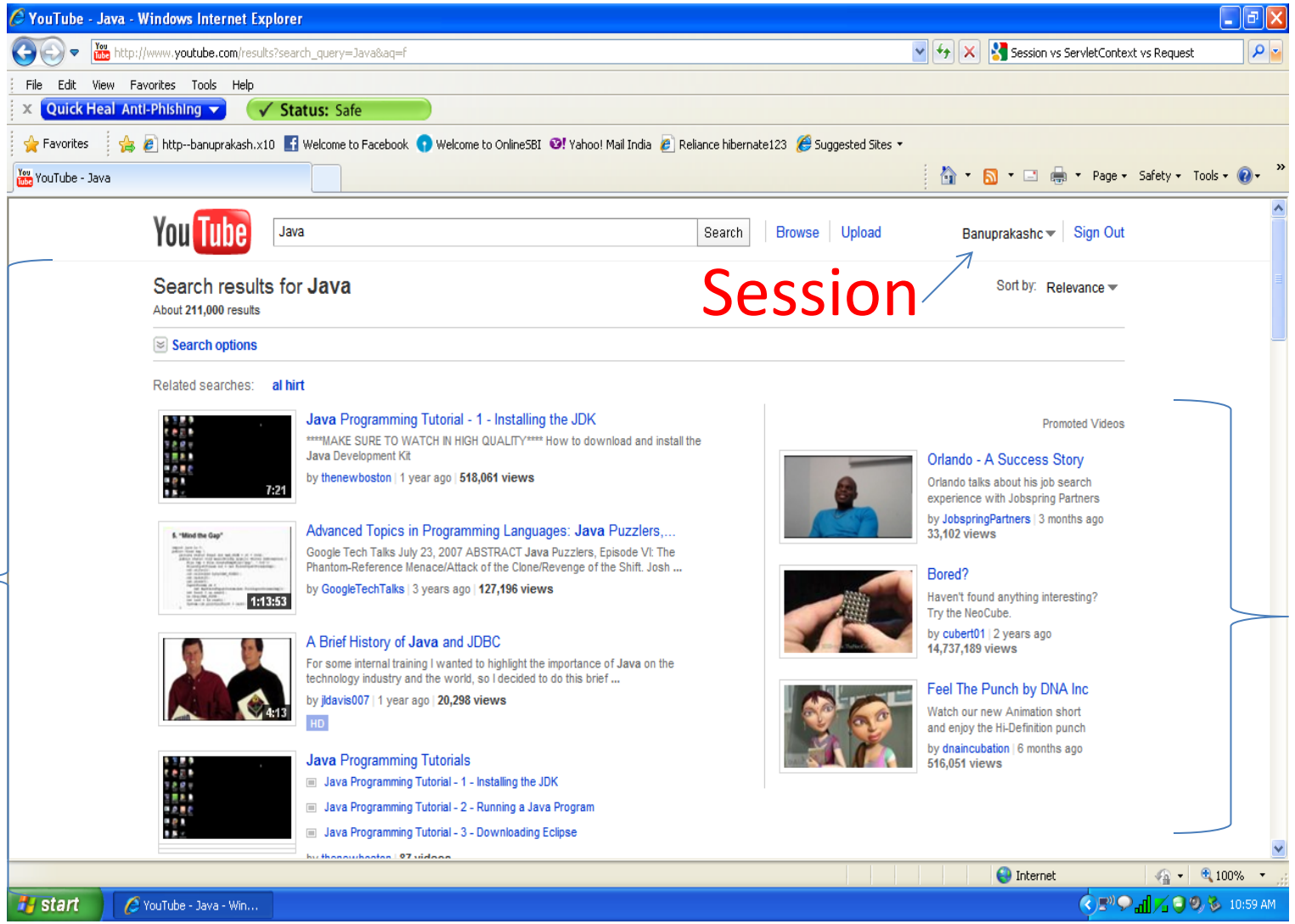



Every Session created for Client's will have an unique Session ID.
The Session ID will be sent to the client in the form of Cookie or URL-Rewriting.
Client browser stores cookies and for every new request cookie information will be sent to the server through request headers.

# HttpSession API



# Request, Session and ServletContext all in one




Mindtree

*Welcome to possible*

Name

Email

www.mindtree.com/social