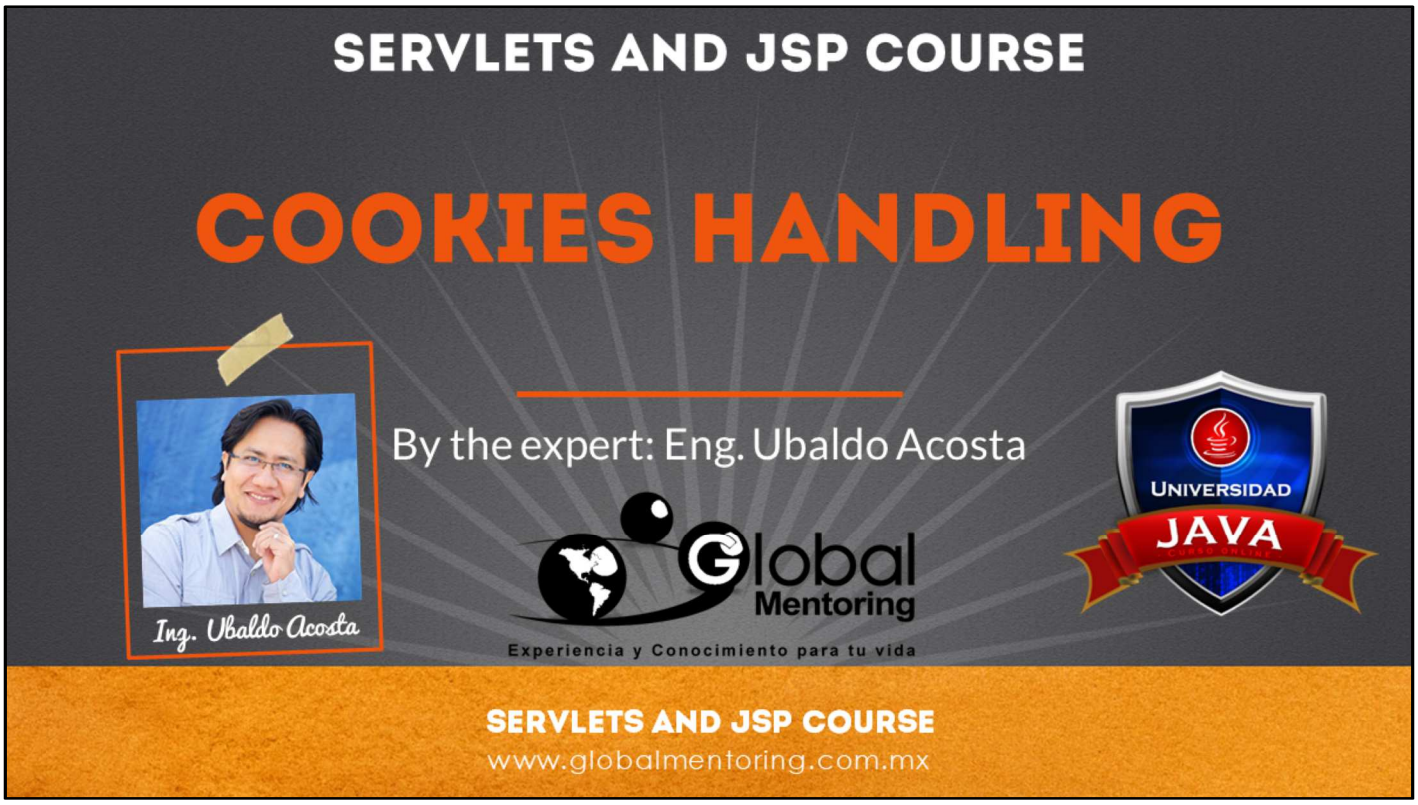
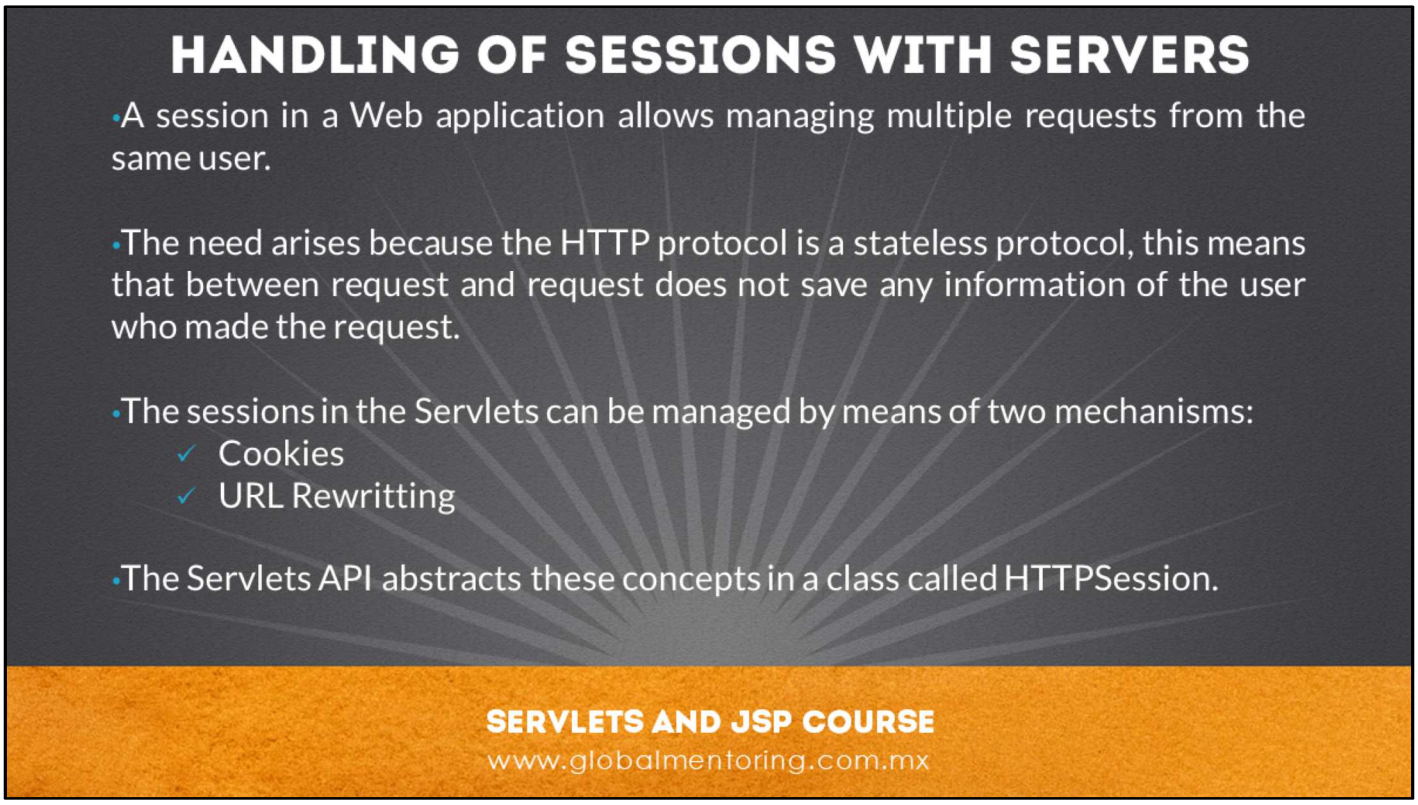


Hello, Ubaldo Acosta greets you again. I hope you're ready to start with this lesson.

We will study the topic of Handling Sessions and Cookies with Servlets.

Are you ready? Come on!



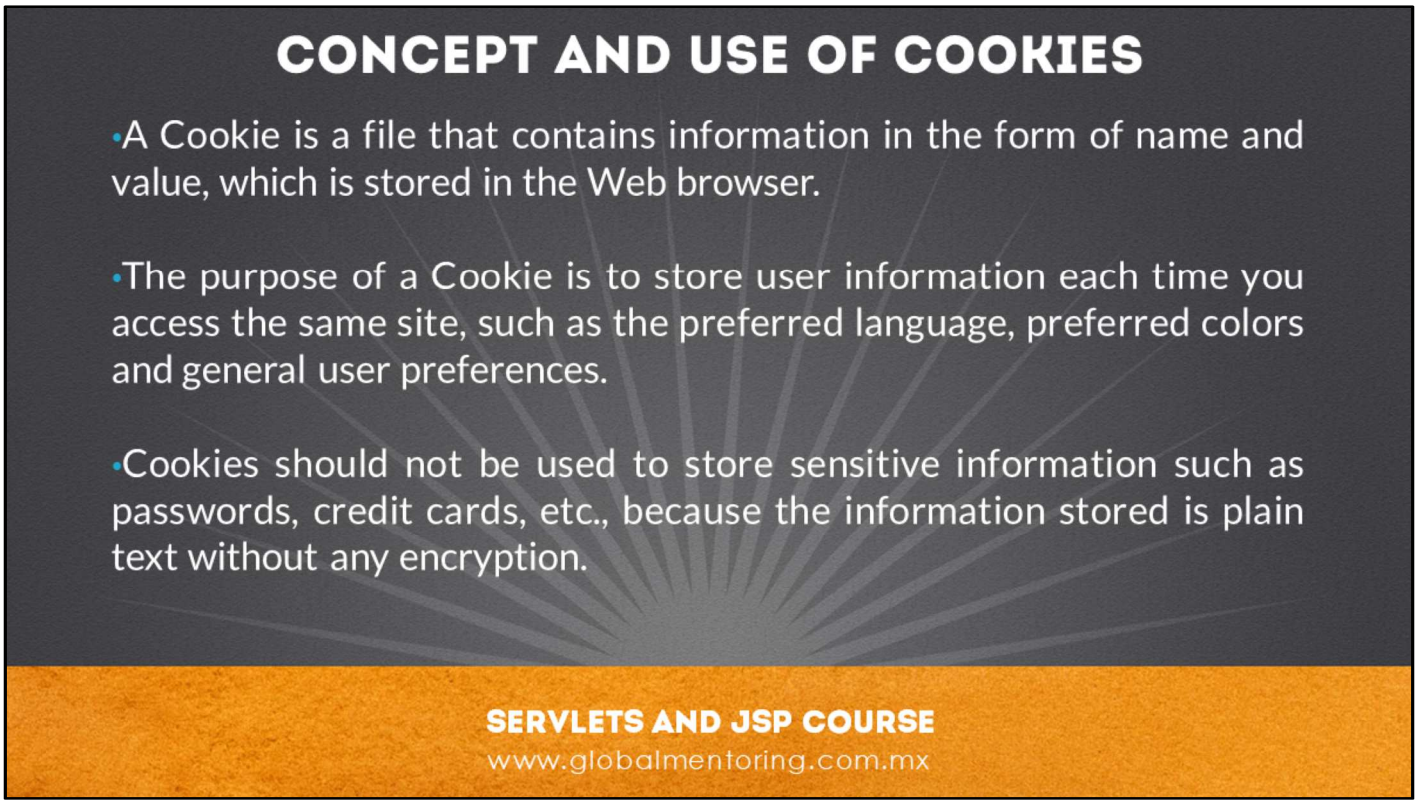
In this lesson we will review the topic of handling sessions with Servlets.

A session in a web application will allow us to manage a series of requests related to the same user, this need arises because the HTTP protocol is a protocol that does not remember information, it is a protocol without a state or in English it is known as a protocol of type stateless, this means that between request and request of certain user will not remember any information.

If we analyze the example of a shopping cart, a user can select a series of articles as he navigates through a web application, so to remember which items have already been added to the shopping cart, the web application needs to remember which elements you have already selected. the user.

The Servlets API will allow us to manage this concept through sessions and sessions we can manage them through two mechanisms: the concept of cookies and the concept of URL Rewritting.

In turn, the Servlets API abstracts these concepts in a class called HTTP Session. In the following topics we will detail each of these techniques and how to apply them to our web applications.



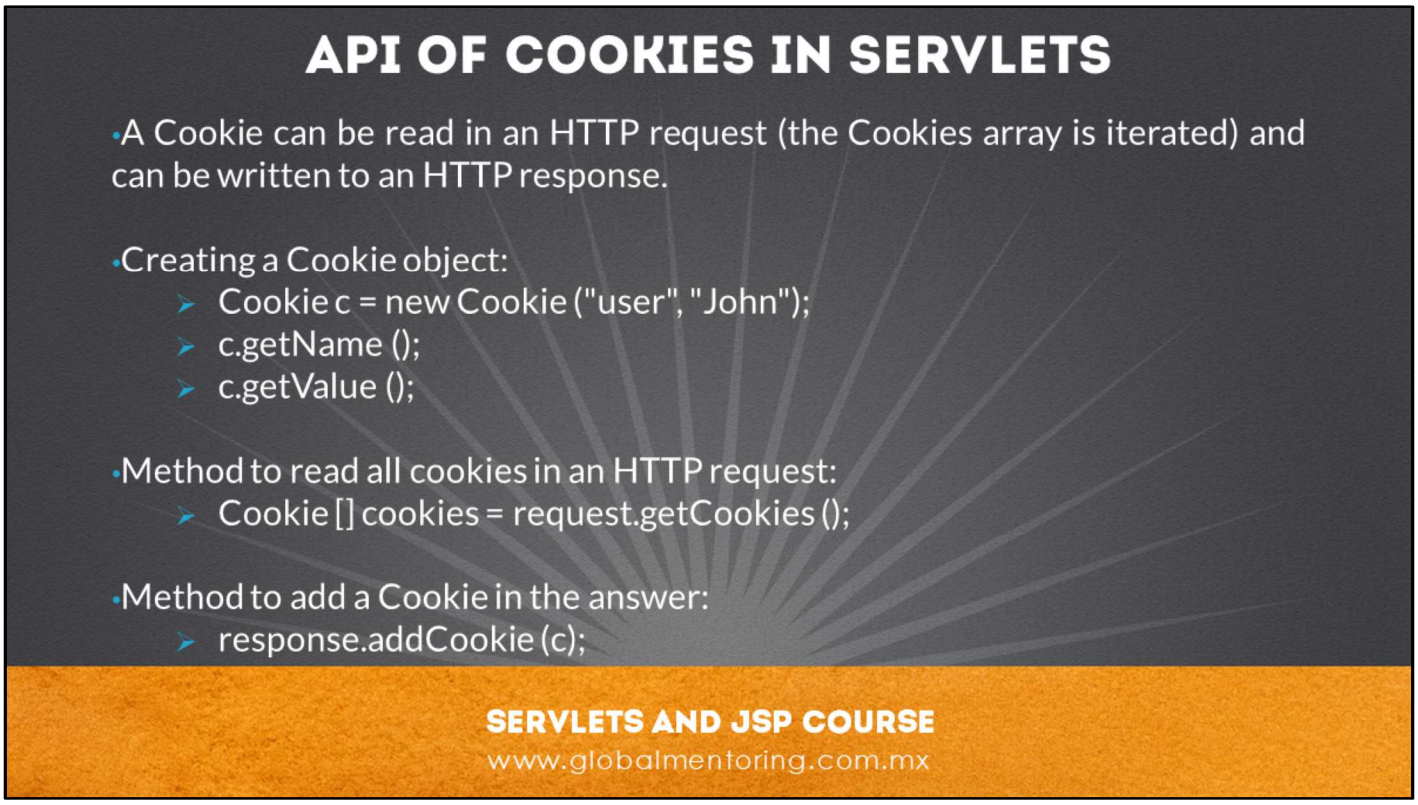
We use cookies to remember user information, this is because the HTTP protocol is a protocol without memory or stateless type. A cookie is information that is stored in our web browser and the way it is stored is in a text file. This text file stores information form name and a value associated with this name.

In Java the Servlets can create this type of objects and send the information to the web browser, in turn we can also read the Cookies that the browser sends to our web server through the Servlets API using the Cookie object. This Cookie object is in the javax.servlet.http package.

This cookie object will allow us to store user information, whether to read new information that is arriving or to send information to the web browser, all this must relate to the same website, ie a cookie is associated with an IP address or the domain of the Web site from which the HTTP request comes.

Cookies are also going to be used to specify the user's preferences, that is, when a user is browsing a web application, they can select their preferred language, their preferred colors and in general, we can include these types of concepts with the use of cookies or session management.

Cookies also have their cons, one of them is that we should not store sensitive user information such as passwords, credit card numbers and in general information that is sensitive, because cookies are stored without any encryption, so that any user can reach our team and verify the cookies that have been stored in our web browser, therefore this type of sensitive information is not advisable to store them by means of cookies.



We are going to review the API of Servlet Cookies. The Servlets will be able to read the cookies that we receive from an HTTP request and will also be able to create, modify and, once it has manipulated the cookies, send this information again by means of the HTTP response.

Through the Servlets API we can use the Cookie class and create a new cookie using this syntax:

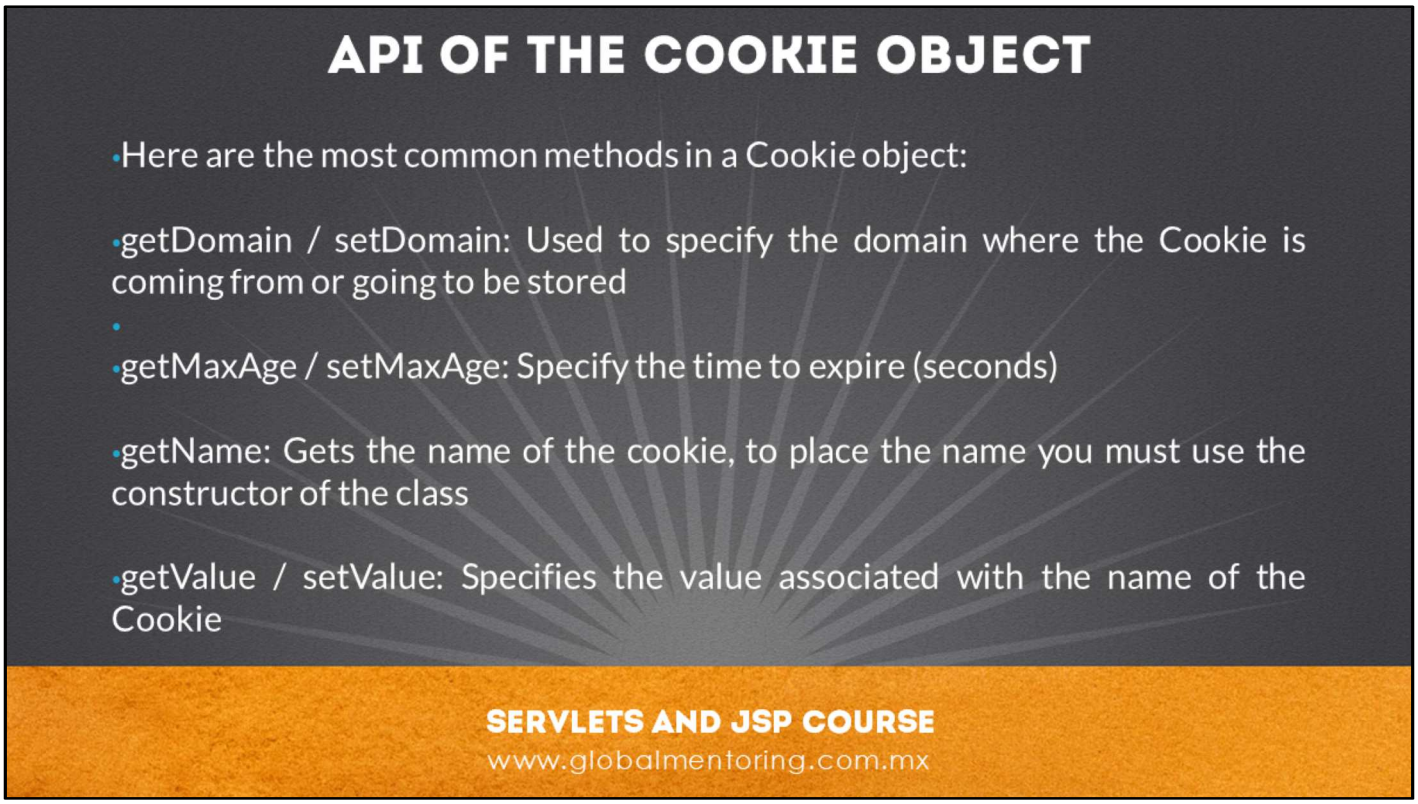
Cookie c = new Cookie ("user", "John"); What we are doing is adding the name of the cookie and the associated value to that name and so we can add N number of cookies in the response of our http request and once we have the cookie object we can retrieve information from the cookie by means of the following methods:

c.getName (); The getName method to get the name of the cookie that would be our first parameter.

c.getValue (); and we can also recover the value associated with that name, which would be the second parameter when we create the cookie object.

Now to read the cookies that we are receiving from part of an HTTP request we can use the request object and the getCookies () method. It should be noted that the Servlets API does not allow directly recover a specific cookie, but they need to recover the entire arrangement of cookies and manipulate it to recover the cookie we are looking for and once we have already manipulated or created our cookies we can add these cookies in the response of our server through the response object and using the addCookie (c) method; and as a parameter we are sending a cookie object that we created previously.

Then with these methods we will be able to interact in an HTTP request and in the response of our HTTP request.



We will review the cookie object in more detail. The cookie object has the following methods, remember that a cookie is associated with the IP address of the server, with one hour and with a specific user, this with the objective that each client or user that requests the Web resource is associate a single cookie. For this reason, in a cookie object getDomain / setDomain we can obtain the domain or we can also specify the domain from which the request is being made. This method is used to specify where the cookie comes from or where it is going to be stored.

We also have the getMaxAge / setMaxAge method. These methods are used to specify the duration of our cookie. The time is specified in seconds and what it means is that after these seconds, the cookie expires, so it will be removed from our web browser.

We also have the getName () method. This method returns the name of the cookie which was provided when creating our cookie object.

And finally when we create our cookie we can obtain the associated value again or we can also modify it using the setValue method. Then the getValue / setValue methods allow us to manipulate or obtain the value related to the name of a cookie.

Then the management of cookies will allow us to remember user information, and this is a way to manage the Servlet API sessions. Next, we will create an exercise to put the concept of cookies into practice in the Servlets API.