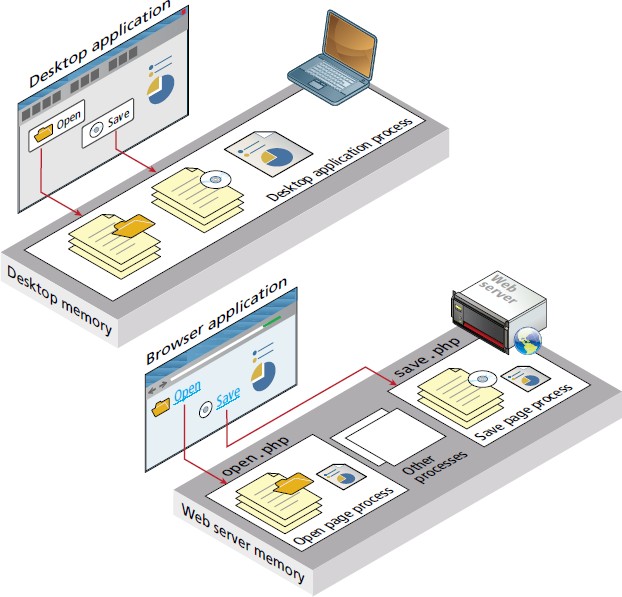
# Why is a state problem for web application? Explain

## The Problem of State in Web Applications:

At first glance this problem does not seem especially formidable. Single-user desktop applications do not have this challenge at all because the program information for the user is stored in memory (or in external storage) and can thus be easily accessed throughout the application. Yet one must always remember that web applications differ from desktop applications in a fundamental way. Unlike the unified single process that is the typical desktop application, a web application consists of a series of disconnected HTTP requests to a web server where each request for a server page is essentially a request to run a separate program, as shown in Figure 5.1.

# What are HTTP cookies? How do you handle them in php?

# Or What are cookies What is the purpose of it? Demonstrate

Cookies are a client-side approach for persisting state information. They are name=value pairs that are saved within one or more text files that are managed by the browser. These pairs accompany both server requests and responses within the HTTP header. While cookies cannot contain viruses, third-party tracking cookies have been a source of concern for privacy advocates.

Cookies were intended to be a long-term state mechanism. They provide website authors with a mechanism for persisting user-related information that can be stored on the user’s computer and be managed by the user’s browser.

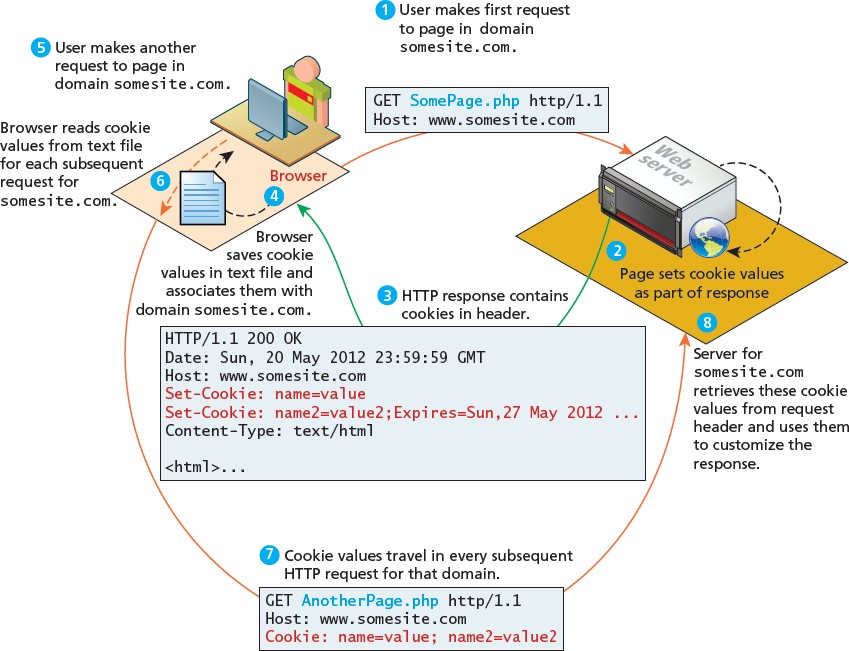


Figure 5.3 Cookies at work

### Using Cookies

Listing 5.1 illustrates the writing of a persistent cookie in PHP. It is important to note that cookies must be written before any other page output.

<?php

// add 1 day to the current time for expiry time

$expiryTime = time()+60\*60\*24;

// create a persistent cookie

$name = "Username";

$value = "Ricardo";

setcookie($name, $value, $expiryTime);

?>

Listing 5.1 Writing a cookie

The setcookie() function also supports several more parameters, which further customize the new cookie. You can examine the online official PHP documentation for more information. Listing 5.2 illustrates the reading of cookie values. Notice that when we read a cookie, we must also check to ensure that the cookie exists. In PHP, if the cookie has expired (or never existed in the first place), then the client’s browser would not send anything, and so the $\_COOKIE array would be blank.

<?php

if( !isset($\_COOKIE['Username']) ) {

//no valid cookie found

}

else {

echo "The username retrieved from the cookie is:"; echo $\_COOKIE['Username'];

}

?>

# Discuss jquery selectors in detail?

# What does $() short and stand for in Jquery?Explain any 3 Jquery selectors?

# With suitable examples ,explain any 4 basic jquery selectors?

### jQuery Selectors

The relationship between DOM objects and selectors is so important in JavaScript programming that the pseudo-class bearing the name of the framework, jQuery(), lets programmers easily access DOM objects using selectors passed as parameters. Because it is used so frequently, it has a shortcut notation and can be written as $(). This $() syntax can be confusing to PHP developers at first, since in PHP the $ symbol indicates a variable**.**

### Basic Selectors

The four basic selectors were defined back in Chapter 3, and include the universal selector, class selectors, id selectors, and elements selectors. To review:

* $("\*") Universal selector matches all elements (and is slow).
* $("tag") Element selector matches all elements with the given element name.
* $(".class") Class selector matches all elements with the given CSS class.
* $("#id") Id selector matches all elements with a given HTML id attribute. For example, to select the single <div> element with id="grab" you would write:

var singleElement = $(“#grab”);

### Attribute Selector

An attribute selector provides a way to select elements by either the presence of an element attribute or by the value of an attribute.

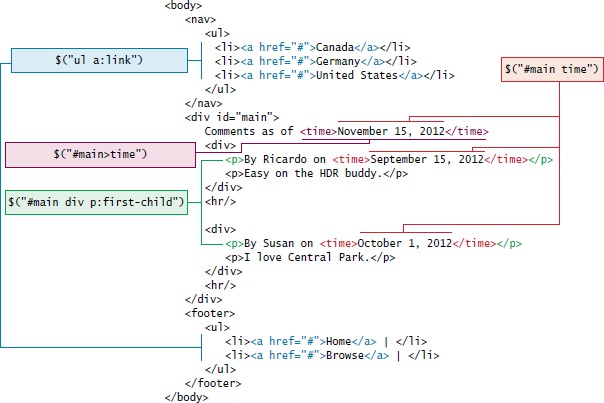


Figure 5.8 illustration of jQuery selectors and the HTML

# Discuss Session cookies , persistent cookies ,session state

# ?

### Session State

# Session state is a server-based state mechanism that lets web applications store and retrieve objects of any type for each unique user session. That is, each browser session has its own session state stored as a serialized file on the server, which is deserialized and loaded into memory as needed for each request, as shown in Figure

# With neat diagram explain SOAP and RESET web services?

## Overview of Web Services

Web services are the most common example of a computing paradigm commonly referred to as service-oriented computing (SOC), which utilizes something called “services” as a key element in the development and operation of software applications.

A service is a piece of software with a platform-independent interface that can be dynamically located and invoked. Web services are a relatively standardized mechanism by which one software application can connect to and communicate with another software application using web protocols. Web services make use of the HTTP protocol so that they can be used by any computer with Internet connectivity. As well, web services typically use XML or JSON (which will be covered shortly) to encode data within HTTP transmissions so that almost any platform should be able to encode or retrieve the data contained within a web service.

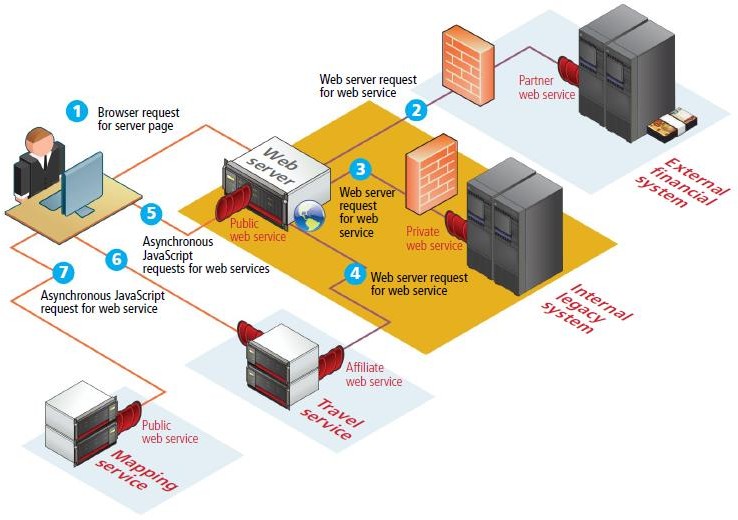


Figure 5.16 Overview of Web Services

The services are

1. SOAP Services
2. REST Services

### Identifying and Authenticating Service Requests

The previous section illustrated a sample request to a REST-based web service and its XML response. That particular service was openly available to any request (though its term of service license limited how the response data could be used).

Most web services are not open in the same way. Instead, they typically employ one of the following techniques:

* Identity. Each web service request must identify who is making the request.
* Authentication. Each web service request must provide additional evidence that they are who they say they are.

Many web services are not providing information that is especially private or proprietary. For instance, the Flickr web service, which provides URLs to publicly available photos on their site in response to search criteria, is in some ways simply an XML version of the main site’s already existing search facility. Since no private user data is being requested, it only expects each web service request to include one or more API keys to identity who is making the request.