# DOWNLOAD PDF EXPERIMENT 6.3: BUILD A WEB SERVER USING SOCKETS

## Chapter 1 : EF v The underlying Provider failed to open

*First step to create a web server is to create a network socket which can accept connection on certain TCP port. HTTP server usually listen on port 80 but we will use a different port for testing purpose.*

Network Python network sockets programming tutorial In this tutorial you will learn about in network programming. You will learn about the client-server model that is in use for the World Wide Web, E-mail and many other applications. Client server with email protocol The client server model is a model where there are n clients and one server. The server replies to those messages received. Python Network Programming â€" Part 1: Build 7 Python Apps socket server code This code will start a simple web server using sockets. It waits for a connection and if a connection is received it will output the bytes received. In a second screen, open a client with Telnet. If you use the same machine for the client and server use: If you use another machine as client, type the according IP address of that machine. You can find it with ifconfig. Everything you write from the client will arrive at the server. The server sends the received messages back. An example output below Click to enlarge: The client script below sends a message to the server. The server must be running! Limitations of the server code The server code above can only interact with one client. To let the server interact with multiple clients you need to use multi-threading. We rebuild the server script to accept multiple client connections: Every message can have a specific meaning in an application. This is known as the protocol. The meaning of these messages must be the same on both the sender and receiver side. The Internet Layer is the IPv4 protocol. All we have to define is the Application Layer. Below we modified the server to accept simple commands We use the non-threading server for simplicity. We changed the port to  Server code with a protocol:

# DOWNLOAD PDF EXPERIMENT 6.3: BUILD A WEB SERVER USING SOCKETS

## Chapter 2 : Multiple UDP sockets and multiple clients - Stack Overflow

*This article provides metrics for HTTP long-polling, HTTP short-polling, server-sent events, and WebSockets in the form of bandwidth per request. The primary audience of this article is a seasoned web developer or library author, however web developers of all skill levels may benefit from the following material.*

Compromised data can cost thousands of dollars to company. In the last section, we compiled LDAP authentication module into the Apache build to provide a Authentication mechanism. This creates a problem. HTTPS runs on port  If you are just going to use this server for DAV, then I will highly suggest that you close port  The following is a over-simplified structure of the layers involved in SSL. In this algorithm, encryption and decryption is performed using a pair of private and public keys. The Web-server holds the private Key, and sends the Public key to the client in the Certificate. The client checks to see if the certificate has expired. Then the client checks if the Certificate Authority that signed the certificate, is a trusted authority listed in the browser. This explains why we need to get a certificate from a a trusted CA. If everything is successful the SSL connection is initiated. Anything encrypted with Private Key can only be decrypted by using the Public Key. Similarly anything encrypted using the Public Key can only be decrypted using the Private Key. There is a common mis-conception that only the Public Key is used for encryption and Private Key is used for decryption. This is not case. However if one key is used for encryption then the other key must be used for decryption. A message can not encrypted and then decrypted using only the Public Key. Using Private Key to encrypt and a Public Key to decrypt ensures the integrity of the sender owner of the Private Key to the recipients. Using Public Key to encrypt and a Private Key to decrypt ensures that only the inteded recipient owner of the Private Key will have access to the data. Symmetric Cryptography - Actual transmission of data: In symmetric cryptography the data can be encrypted and decrypted using the same key. The Key for symmetric cryptography is exchanged during the initiation process, using Public Key Cryptography. This ensures the Authenticity of the sender. This will ensure the Authenticity of the Receiver i. This Digital Signature can be used by the receiver to ensure the Integrity of the message and authenticity of the Sender. If they are equal, the data was not modified during transmission, and the integrity of the Original "Clear Text" has been maintained 6. Test Certificates While compiling Apache we created a test certificate. We used the command: Certificates for Production use For production use you will need a certificate from a Certificate Authority hereafter CA. As mentioned in the Encryption Algorithms section, if the CA is not listed as a trusted authority, your user will get a warning message when trying to connect to a secure location. What you are about to enter is what is called a Distinguished Name or a DN. California Locality Name eg, city []: Seagate Organizational Unit Name eg, section []: In that case you can use the following command: Any file can be specified. If you put in anything else, it will NOT work. Remember the password that you use, for future reference. Once the process is complete, you will have private. You will need to submit the public. At this pointe the public. The Digital Certificate is in the format defined by X. The following shows the structure of a typical X v3 Digital Certificate Certificate.

# DOWNLOAD PDF EXPERIMENT 6.3: BUILD A WEB SERVER USING SOCKETS

## Chapter 3 : Responding to a Http Message from a Console Application

*For each client that contacts the listening socket, lets call it 0, I create a separate socket for, so I can send data to it without clogging socket 0. What I want to do is move all communication with a specific client to the new socket.*

An application connection fails through a dual-use socket that is redirected by using a WFP callout driver Content provided by Microsoft Applies to: Resolution Hotfix information A supported hotfix is available from Microsoft Support. However, this hotfix is intended to correct only the problem that is described in this article. Apply this hotfix only to systems that are experiencing the problem described in this article. This hotfix might receive additional testing. Therefore, if you are not severely affected by this problem, we recommend that you wait for the next software update that contains this hotfix. If the hotfix is available for download, there is a "Hotfix download available" section at the top of this Knowledge Base article. If this section does not appear, contact Microsoft Customer Service and Support to obtain the hotfix. Note If additional issues occur or if any troubleshooting is required, you might have to create a separate service request. The usual support costs will apply to additional support questions and issues that do not qualify for this specific hotfix. For a complete list of Microsoft Customer Service and Support telephone numbers or to create a separate service request, go to the following Microsoft website: Note The "Hotfix download available" form displays the languages for which the hotfix is available. If you do not see your language, it is because a hotfix is not available for that language. Prerequisites To apply this hotfix in Windows 8. There are no prerequisites for installing this hotfix on Windows 8 and Windows Server  Restart requirement You must restart the computer after you apply this hotfix. Hotfix replacement information File information The English United States version of this hotfix installs files that have the attributes that are listed in the following tables. The dates and the times for these files on your local computer are displayed in your local time together with your current daylight saving time DST bias. Additionally, the dates and the times may change when you perform certain operations on the files. However, only "Windows 8. To request the hotfix package that applies to one or both operating systems, select the hotfix that is listed under "Windows 8. Always refer to the "Applies To" section in articles to determine the actual operating system that each hotfix applies to.

# DOWNLOAD PDF EXPERIMENT 6.3: BUILD A WEB SERVER USING SOCKETS

## Chapter 4 : How to create HTTP Server in Java - ServerSocket Example

*You will learn about the client-server model that is in use for the World Wide Web, E-mail and many other applications. Client server (with email protocol) The client server model is a model where there are n clients and one server.*

Chrome, Firefox or Internet Explorer. Though there is no short of good open source library e. Book is very focused on practical and you will find lot of interesting example related to common networking task e. HTTP server usually listen on port 80 but we will use a different port for testing purpose. You can use ServerSocket class in Java to create a Server which can accept requests, as shown below import java. Now our server is ready and listening for incoming connection on port  If you connect to http: If your browser is smart and giving up after waiting for sometime then try telnet command. You should be able to connect to server and as soon as you stop your server telnet will show that "could not open connection to the host, on port  So now we have a server which is listening for connection on port but we are not doing anything with incoming connection but we are not rejecting them either. All of them are waiting to be served and stored inside server object. Do you see the while true loop? Any guess why we have that? This allows us to keep our program running, without this infinite loop our program will finish execution and server will be shutdown. In Java, you can accept incoming connection by blocking call to accept method, as shown below: As soon as a client connect it returns the Socket object which can be used to read client request and send response to client. Once you are done with client you should close this socket and get ready to accept new incoming connection by calling accept again. So basically, our HTTP server should work like this: This is endless cycle until server is stopped. When you connect to http: You can read the content of request using InputStream opened from the client socket. Listening for connection on port  So now our server is not only listening for connection, but accepting it and also reading HTTP request. Now only thing remaining is to send HTTP response back to the client. Sun Mar 29  It means our HTTP Server is working properly, it is listening on port , accepting connection, reading request and sending response. By using try-with-resource statement of Java 7 , we have also simplified our code, because socket will automatically closed by Java once you are done with response. Only limitation of this server is that it can serve one client at a time. If request processing takes longer time, which is not in our case, the other connection has to wait. This problem can be solved by using threads or Java NIO non blocking selectors and channels. This is a good example to learn network programming in Java. You have learned how to use ServerSocket and Socket class from this example. Remember, ServerSocket is used to receive connections in Server application and Socket is used to send and receive data from individual client.

# DOWNLOAD PDF EXPERIMENT 6.3: BUILD A WEB SERVER USING SOCKETS

## Chapter 5 : Unable to add Downstream server in Upstream server console.

*With sufficient hardware (RAM, processor speed, etc.), the same computer can serve as a web server, an ftp server, and mail server (pop, smtp, imap, or all of the above) all at the same time. Each service is associated with a port.*

Windows Create a new Windows user who is part of the Administrators group and has the privilege to act as part of the operating system. See Create a Windows user for WebSphere. Create a user by entering the following command in a command prompt: Linux and Solaris Create a shadow password file by entering pwconv with no parameters in the command prompt. If the shadow password file does not exist, an error occurs after enabling global security and configuring the user registry as Local OS. Add the user who you created in step 2 to the root group. Save and close the file. Right-click Users and select New User. Type a user name and password in the appropriate boxes, and type any other information you require in the remaining boxes. Click Users, right-click the user you just created and select Properties. Click the Member Of tab and then click Add. Click OK and then click OK again. Click Add User or Group. Under Administrative security, select Administrative user roles. Click Add and do the following: Click Administrator under roles. Add the newly created user to Mapped to role and map it to Administrator. Click OK and save your changes. Restart the WebSphere profile. Click Security Configuration Wizard. Ensure Enable Application Security checkbox is enabled. Select Federated Repositories and click Next. Specify the credentials you want to set and click Next. WebSphere will start using the default keystore and truststore. Enable SSL custom key and truststore Truststores and keystores can be created using ikeyman utility or admin console. To make ikeyman work properly, enure that the WebSphere installation path does not contain parentheses. Click Keystores and certificates under Related items. Type a logical name and description. Specify the path where you want your keystore to be created. If you have already created a keystore through ikeyman, specify the path to the keystore file. Specify and confirm the password. Choose the keystore type and click Apply. Save the master configuration. If you had added already created a keystore using ikeyman, your certificate will appear. Otherwise, you need to add a new self-signed certificate by performing the following steps: Specify appropriate values on the certificate form. Ensure that you keep Alias and common name as fully-qualified domain name of the machine. Repeat steps 2 through 10 for creating a truststore. Click Manage endpoint security configuration. The local topology map opens. Under Inbound, select direct child of nodes. Under Related items, select SSL configurations. From the truststore name and keystore name drop-down lists, select the custom truststore and keystore that you created.

# DOWNLOAD PDF EXPERIMENT 6.3: BUILD A WEB SERVER USING SOCKETS

## Chapter 6 : Python Tutorial: Network Programming - Server & Client A : Basics -

*(Windows) Create a new Windows user who is part of the Administrators group and has the privilege to act as part of the operating system. (See Create a Windows user for WebSphere.) (Linux, UNIX) The user can be a root user or another user who has root privileges.*

In this tutorial you will learn about in network programming. You will learn about the client-server model that is in use for the World Wide Web, E-mail and many other applications. Client server with email protocol The client server model is a model where there are n clients and one server. The server replies to those messages received. Python Network Programming â€" Part 1: Build 7 Python Apps socket server code This code will start a simple web server using sockets. It waits for a connection and if a connection is received it will output the bytes received. In a second screen, open a client with Telnet. If you use the same machine for the client and server use: If you use another machine as client, type the according IP address of that machine. You can find it with ifconfig. Everything you write from the client will arrive at the server. The server sends the received messages back. An example output below Click to enlarge: The client script below sends a message to the server. The server must be running! Limitations of the server code The server code above can only interact with one client. To let the server interact with multiple clients you need to use multi-threading. We rebuild the server script to accept multiple client connections: Every message can have a specific meaning in an application. This is known as the protocol. The meaning of these messages must be the same on both the sender and receiver side. The Internet Layer is the IPv4 protocol. All we have to define is the Application Layer. Below we modified the server to accept simple commands We use the non-threading server for simplicity. We changed the port to  Server code with a protocol: So how do you start and build one in Python, just for fun? If you are curious, you could read the entire protocol. To create a socket we use the command: The second argument tells the library to use stream sockets, which are traditionally implemented on the TCP protocol. We then must use the commands to authenticate with the server: USER botname botname botname: Summing up, we get this class save it as irc. This is a fun bot! We will keep our ro bot simple for explanatory purposes. Connect with a traditional irc client mirc,hexchat,irsii to the the channel and observe the experiment has worked! You can now extend it with any cool features you can imagine.

# DOWNLOAD PDF EXPERIMENT 6.3: BUILD A WEB SERVER USING SOCKETS

## Chapter 7 : Configuring SSL for Apache Tomcat

*I'm trying to build a simulation for multiple socket clients. My server has the following code to listen to multiple clients. My socket are from a very simple class drive from CAsyncSocket and my environment is windows MFC.*

More information How does a WebSocket basically work? WebSocket communications are managed by a server. Each client have to introduce itself by sending a handshake request via the WebSocket protocol ws. The server accepts or not to open a socket connection by sending a handshake response. Only after the handshake acceptance, both sides can communicate. Many clients can be connected to the server. For some performance reasons, the limit of clients can be set before creating a new socket. The server works like a daemon and is executed by the following command: The browser introduces itself by sending HTTP headers, something like: Moreover, server responses are not the same regarding versions. In the following code, it deals with the version  Sent messages have to be formatted in a special way. In fact, if you try to display them, they will be encrypted, so they have to be unmasked to be human readable. The RFC describes the way to unmask data. An implementation of that description in PHP would be this: If a sent text is wrongly encoded, the client might close the connection or not correctly receive it. This is a basic implementation without encryption: This method below have to be called after the handshake. The server will treat the message and will send a kill request to the process. However, we can make it in another way: The client side This part is really easy. Nothing special to figure out other than the WebSocket initialization and events which handle WebSockets: We will see the onmessage event in the receiving data section. This is a basic client side implementation: When there is something new, the server will send automatically to the concerned client. That new data is handled by the onmessage event. This message is directly handled by the server.

# DOWNLOAD PDF EXPERIMENT 6.3: BUILD A WEB SERVER USING SOCKETS

## Chapter 8 : Build a real-time application using HTML5 WebSockets - Thoughts and Experiments

*blog.quintoapp.com(): Create a new socket using the given address family, socket type and protocol number. blog.quintoapp.com(address): Bind the socket to address. blog.quintoapp.com(backlog): Listen for connections made to the socket.*

Specifies information to communicate with a site. For example, a Web site binding includes the IP address or unspecified IP addresses , the port number, and an optional host header used to communicate with the site. Only one certificate can be bound to a combination of IP address and the port. A value of "1" specifies that the secure connection be made using the port number and the host name obtained by using Server Name Indication SNI. A value of "2" specifies that the secure connection be made using the centralized SSL certificate store without requiring a Server Name Indicator. A value of "3" specifies that the secure connection be made using the centralized SSL certificate store while requiring Server Name Indicator Centralized SSL certificate support enables you to create a centralized certificate store that can contain multiple certificate files. You can name the certificate files to correspond to the host names that they contain. When a request comes in, IIS matches the port, determines the host name from the request, and searches the centralized certificate store for a certificate file with a matching name. It uses that certificate. This is especially useful for SSL connections that host multiple servers on a single network address. For more information, see IIS 8. The sslFlags attribute is only set when the protocol is https. The default value is 0. Configuration Sample The following example defines a site named Contoso with two bindings. The first binding is for a hostname of "www. This commits the configuration settings to the appropriate location section in the ApplicationHost. C using System; using System. Add bindingElement1 ; serverManager. CreateNewElement "binding" ; bindingElement. CreateNewElement "binding" ; bindingElement1. AddElement bindingElement1 ; adminManager. Item i ; if element. Echo "Element not found! Item i If element.

# DOWNLOAD PDF EXPERIMENT 6.3: BUILD A WEB SERVER USING SOCKETS

## Chapter 9 : C++ Creating multiple socket clients - Stack Overflow

*Well we are building a simple server and a client where server will open a socket and wait for clients to connect. Once client is connected he can send a message and server will process the message and reply back the same message but converted into upper case to demonstrate the server side processing and transmission.*

Simple Socket In the following code, the server sends the current time string to the client: Create a new socket using the given address family, socket type and protocol number. Bind the socket to address. Listen for connections made to the socket. The backlog argument specifies the maximum number of queued connections and should be at least 0; the maximum value is system-dependent usually 5 , the minimum value is forced to 0. The return value is a pair conn, address where conn is a new socket object usable to send and receive data on the connection, and address is the address bound to the socket on the other end of the connection. At accept , a new socket is created that is distinct from the named socket. This new socket is used solely for communication with this particular client. For TCP servers, the socket object used to receive connections is not the same socket used to perform subsequent communication with the client. This allows a server to manage connections from a large number of clients simultaneously. Send data to the socket. The socket must be connected to a remote socket. Returns the number of bytes sent. Applications are responsible for checking that all data has been sent; if only some of the data was transmitted, the application needs to attempt delivery of the remaining data. Mark the socket closed. The remote end will receive no more data after queued data is flushed. Sockets are automatically closed when they are garbage-collected, but it is recommended to close them explicitly. It just produces client sockets. A simple protocol based around shared memory and locks or semaphores is by far the fastest technique. On most platforms, this will take a shortcut around a couple of layers of network code and be quite a bit faster. For more info, visit Character Encoding. So, if any kind of text string is to be sent across the network, it needs to be encoded. Likewise, when a client receives network data, that data is first received as raw unencoded bytes. Instead, we need to decode it first. Echo Server This is an echo server: