DOWNLOAD PDF TCP/IP SOCKETS IN JAVA, SECOND EDITION

Chapter 1 : Practical TCP/IP Sockets in Java, Second Edition

*TCP/IP Sockets in Java: Practical Guide for Programmers (The Practical Guides) [Kenneth L. Calvert, Michael J. Donahoo] on racedaydvl.com *FREE* shipping on qualifying offers. The networking capabilities of the Java platform have been extended considerably since the first edition of the book.*

Download Microservices for Java Developers: A hands-on introduction to frameworks and containers. Brought to you in partnership with Red Hat. First of all, I wish you a very happy new year, guys. Being a Java developer and the author of a Java blog , I frequently receive requests from Java programmers from all over the world asking how they can improve themselves. You can take inspiration from these ideas to create your own goals and resolutions. Learn Java Performance Tuning In the last a couple of years, I have taken more than 50 interviews for senior Java developers , and one skill that I clearly see lacking is knowledge and understanding about JVM internals, GC behavior, and Java performance tuning. I have been reading it for years and still refer it whenever I get time. My goal is to read it again this year. Everyday Coding for 2 hours Another thing I noticed last yea r is that as your experience grows, you spend your time on coordination, replying to emails, being a catalyst, troubleshooting, mentoring, and generally being a project manager kind of person. What you left behind is coding , which is the single most important skill for a programmer. If you feel that you are not doing enough coding, then make a resolution to code every day. At least, write something, be it on your project, an open source framework, a library, or a utility. Since starting is the most difficult thing, I suggest that any time you feel resistance, start reading and refactoring code for fun. You will enjoy that and, in the process, you will also write code. Profile Your Java application Once a Month This resoluti on is attached with the first resolution about reading a good book on JVM internals and performance tuning. Just reading the book will not be enough. You have to apply that knowledge in your live project. I suggest you profile your Java application , at least once a month and spend a good amount of time understanding and analyzing the results. You can also take a heap dump of your Java process or, if you had a recent crash, then take that heap dump and find out which object is taking most of your memory. Is there a memory leak in your Java application? What is the cause? What will happen if other K new clients access your application? If you can answer all these questions comfortably, then you are in good shape. Participate in Coding Challenges This goal is, again, s omewhat related to our second resolution â€" write code every day for 2 hours. If you are starving for challenging code, then there is nothing better than participating in programming and coding challenge. There are many websites on the internet that host programming challenges and give you the opportunity to test your skills, but TopCoder is simply the best. If you are looking for some tough programming challenges, then you can also check out my list of good websites to practice coding. I have asked many questions from my list of Java networking programming questions, but most developers fail to answer most of them. Some of them even struggle to outline the key differences between TCP and UDP , which I thought was too basic to ask any Java developer of years of experience. Java 9 The year saw a couple of big releases, and one of them was JDK 9. I have yet to start with JDK 9, but this is the first thing I am going to look into in If you want to learn new features of Java 9, e. With lots of exciting features, e. Beginner to Guru is a good course to start with. The most interesting part is the OAuth 2. Unfortunately, there are not many resources to learn Spring Security 5. Unit Testing If you want to become a better developer in , then you should work on your unit testing skills. And not just unit testing, but automated testing in general. This also includes integration testing. You can learn JUnit 5 and other advance unit testing libraries like Mockito, PowerMock, Cucumber, and Robot to take your unit testing skill to next level. Mockito is really powerful and allows you to write a unit test for complex classes by mocking dependencies and just focusing on the objects under test. If you a beginner in unit testing and want to learn it in , then the JUnit and Mockito Crash Course from Udemy is a good starting point. If you are a Java programmer with a couple of years of experience, you can also take inspiration from this list to set your goal. Some other things you can add to this list are learning Android , Docker , and Spark , as those are essential for any Java programmers. I have purposefully kept this simple and achievable because I personally believe that small successes lead to big successes. Setting small

goals and achieving them is better than setting big, impractical goals and failing before kick-off. So what are you waiting for? Write down your resolutions for the new year and share them with us. At the end of the year, you can come back here and tell us about how much you achieved. Asynchronous and Event-Based Application Design. Read More From DZone.

Chapter 2 : TCP/IP Sockets in C, 2nd Edition [Book]

Focused, tutorial-based instruction in key sockets programming techniques allows reader to quickly come up to speed on Java applications. Concise and up-to-date coverage of the most recent platform () for Java applications in networking technology.

Sockets, protocols, and ports. As you proceed, you will encounter several ways for a socket to become bound to an address. Note that a single socket abstraction can be referenced by multiple application programs. Each program that has a reference to a particular socket can communicate through that socket. In practice, separate programs that access the same socket would usually belong to the same application e. How many support two-way transport? TCP hides all of this, providing a reliable service that takes and delivers an unbroken stream of bytes. We begin by demonstrating how Java applications identify network hosts using the InetAddress and SocketAddress abstractions. In the latter case the name must be resolved to a numerical address before it can be used for communication. For each Java networking class described in this text, we include only the most important and commonly used methods, omitting those that are deprecated or beyond the use of our target audience. However, this is something of a moving target. For example, the number of methods provided by the Socket class grew from 23 to 42 between version 1. Basic Sockets The InetAddress abstraction represents a network destination, encapsulating both names and numerical address information. The class has two subclasses, Inet4Address and Inet6Address, representing the two versions in use. Instances of InetAddress are immutable: To get the addresses of the local host, the program takes advantage of the Network Interface abstraction. Recall that IP addresses are actually assigned to the connection between a host and a network and not to the host itself. This is extremely useful, for example when a program needs to inform another program of its address. Check for empty list: Get and print address es of each interface in the list: At this time the only subtypes of InetAddress are those listed, but conceivably there might be others someday. The 12 Chapter 2: Get names and addresses for each command-line argument: Unable to find address for blah. Link-local IPv6 addresses begin with fe8. You may also have noticed a delay when resolving blah. Your resolver looks in several places before giving up on resolving a name. When the name service is not available for some reasonâ€”say, the program is running on a machine that is not connected to any networkâ€” attempting to identify a host by name may fail. It is, therefore, good to know that you can always refer to a host using the IP address in dottedquad notation. If you can ping a host using one of its names e. If your ping test fails or 2. See also the isReachable method of InetAddress, discussed below. For numeric IPv6 addresses, the shorthand forms described in Chapter 1 may be used. A name may be associated with more than one numeric address; the getAllByName method returns an instance for each address associated with a name. The getAddress method returns the binary form of the address as a byte array of appropriate length. If the instance is of Inet4Address, the array is four bytes in length; if of Inet6Address, it is 16 bytes. As we have seen, an InetAddress instance may be converted to a String representation in several ways. The numeric representation of the address only is returned by getHostAddress. For an IPv6 address, the string representation always includes the full eight groups i. Both methods return the numerical form of the address if resolution cannot be completed. Also, both check permission with the security manager before sending any messages. They all work for both IPv4 and IPv6 addresses. The fourth method checks whether it is a multicast address see Section 4. The scope determines, roughly, how far packets addressed to that destination can travel from their origin. Note that, unlike the other methods, which involve simple syntactic checks, these methods cause the networking system to take action, namely 2. The TTL limits the distance a packet can travel through the network. The NetworkInterface class provides a large number of methods, many of which are beyond the scope of this book. We describe here the most useful ones for our purposes. Similarly, the list of addresses may contain linklocal addresses that also are not globally reachable. The getName methods return the name of the interface not the host. This generally consists of an alphabetic string followed by a numeric part, for example eth0. The loopback interface is named lo0 on many systems. An instance of Socket represents one end of a TCP connection. An instance of ServerSocket listens for TCP connection requests and creates a new

Socket instance to handle each incoming connection. Thus, servers handle both ServerSocket and Socket instances, while clients use only Socket. We begin by examining an example of a simple client. The typical TCP client goes through three steps: Construct an instance of Socket: Close the connection using the close method of Socket. An echo server simply repeats whatever it receives back to the client. The string to be echoed is provided as a command-line argument to our client. Some systems include an echo server for debugging and testing purposes. You may be able to use a program such as telnet to test if the standard echo server is running on your system e. Application setup and parameter parsing: The getBytes method of String returns a byte array representation of the string. If we specify a third parameter, Integer. Note that the underlying TCP 18 Chapter 2: Basic Sockets deals only with IP addresses; if a name is given, the implementation resolves it to the corresponding address. If the connection attempt fails for any reason, the constructor throws an IOException. Get socket input and output streams: We send data over the socket by writing bytes to the OutputStream just as we would any other stream, and we receive by reading from the InputStream. Send the string to echo server: Receive the reply from the echo server: This particular form of read takes three parameters: For the client, this indicates that the server prematurely closed the socket. Why not just a single read? TCP does not preserve read and write message boundaries. That is, even though we sent the echo string with a single write , the echo server may receive it in multiple chunks. Even if the echo string is handled in one chunk by the echo server, the reply may still be broken into pieces by TCP. One of the most common errors for beginners is the assumption that data sent by a single write will always be received in a single read. We can communicate with an echo server named server. Specifying the local address may be useful on a host with multiple interfaces. String arguments that specify destinations can be in the same formats that are accepted by the InetAddress creation methods. The last constructor creates an unconnected socket, which must be explicitly connected via the connect method, see below before it can be used for communication. Any subsequent attempt to read from the socket will cause an exception to be thrown. By default, Socket is implemented on top of a TCP connection; however, in Java, you can actually change the underlying implementation of Socket. The Socket class actually has a large number of other associated attributes referred to as socket options. Because they are not necessary for writing basic applications, we postpone introduction of them until Section 4. The constructor that takes a string hostname attempts to resolve the name to an IP address; the 2. The isUnresolved method returns true if the instance was created this way, or if the resolution attempt in the constructor failed. If the InetSocketAddress is unresolved, only the String with which it was created precedes the colon. The typical TCP server goes through two steps: Construct a ServerSocket instance, specifying the local port. Call the accept method of ServerSocket to get the next incoming client connection. Upon establishment of a new client connection, an instance of Socket for the new connection is created and returned by accept. The server is very simple. It runs forever, repeatedly accepting a connection, receiving and echoing bytes until the connection is closed by the client, and then closing the client socket. Basic Sockets if args. We are done with this client!

Chapter 3 : TCP/IP Sockets in Java: Practical Guide for Programmers, Second Edition

Rent TCP/IP Sockets in Java 2nd edition () today, or search our site for other textbooks by Kenneth L. Calvert. Every textbook comes with a day "Any Reason" guarantee. Published by Morgan Kaufmann.

Mostly, it happens when the brand new readers quit using the eBooks as they are unable to use all of them with the appropriate and effectual fashion of reading these books. There present number of reasons behind it due to which the readers stop reading the eBooks at their first most attempt to use them. Yet, there exist some techniques that could help the readers to really have a nice and effective reading encounter. A person should fix the proper brightness of display before reading the eBook. As a result of this they have problems with eye sores and headaches. The best solution to overcome this severe difficulty would be to decrease the brightness of the displays of eBook by making particular changes in the settings. You can also adjust the brightness of display determined by the type of system you are using as there exists lot of the means to correct the brightness. It is proposed to keep the brightness to potential minimal amount as this can help you to raise the time which you can spend in reading and provide you great relaxation onto your eyes while reading. An excellent eBook reader should be installed. You may also use complimentary software that may offer the readers with many functions to the reader than simply a simple platform to read the desirable eBooks. Aside from offering a place to save all your precious eBooks, the eBook reader software even give you a lot of characteristics as a way to boost your eBook reading experience than the conventional paper books. You can even improve your eBook reading experience with help of alternatives provided by the software program like the font size, full screen mode, the specific variety of pages that need to be exhibited at once and also change the colour of the background. You ought not use the eBook continually for many hours without rests. You must take appropriate breaks after specific intervals while reading. However, this will not mean that you should step away from the computer screen every now and then. Constant reading your eBook on the computer screen for a long time without taking any rest can cause you headache, cause your neck pain and suffer from eye sores and also cause night blindness. So, it is important to provide your eyes rest for some time by taking rests after particular time intervals. This can help you to prevent the problems that otherwise you may face while reading an eBook always. While reading the eBooks, you need to prefer to read big text. So, raise the size of the text of the eBook while reading it at the screen. It is suggested that never use eBook reader in full screen mode. It is suggested not to go for reading the eBook in full screen mode. Although it may look simple to read with full screen without turning the page of the eBook fairly often, it set ton of anxiety in your eyes while reading in this mode. Always prefer to read the eBook in the exact same span that will be similar to the printed book. This is so, because your eyes are used to the span of the printed book and it would be comfortable that you read in exactly the same way. Test out various shapes or sizes until you find one with which you will be comfortable to read eBook. By using different techniques of page turn you can additionally improve your eBook experience. You can try many methods to turn the pages of eBook to improve your reading experience. Check out whether you can turn the page with some arrow keys or click a particular section of the screen, apart from using the mouse to manage everything. Prefer to make us of arrow keys if you are leaning forwards. Lesser the movement you have to make while reading the eBook better will be your reading experience. Specialized dilemmas One issue on eBook readers with LCD screens is that it is not going to take long before you strain your eyes from reading. This will definitely definitely help make reading easier. By using every one of these powerful techniques, you can definitely enhance your eBook reading experience to a terrific extent. This advice will help you not only to prevent particular risks that you may face while reading eBook consistently but also facilitate you to take pleasure in the reading experience with great comfort. Practical Guide for Programmers The Practical Guides pdf, epub, docx and torrent then this site is not for you. The download link provided above is randomly linked to our ebook promotions or third-party advertisements and not to download the ebook that we reviewed. We recommend to buy the ebook to support the author. Thank you for reading.

Chapter 4 : Java Socket Server Examples (TCP/IP)

Book Description. Most Internet applications use sockets to implement network communication protocols. This book's focused, tutorial-based approach helps the reader master the tasks and techniques essential to virtually all client-server projects using sockets in Java.

You will also learn how to create a multi-threaded server. Here are the typical steps involve in developing a server program: Create a server socket and bind it to a specific port number 2. Listen for a connection from the client and accept it. This results in a client socket is created for the connection. Read data from the client via an InputStream obtained from the client socket. Close the connection with the client. The steps 3 and 4 can be repeated many times depending on the protocol agreed between the server and the client. The steps 1 to 5 can be repeated for each new client. And each new connection should be handled by a separate thread. Create a Server Socket: Create a new object of the ServerSocket class by using one of the following constructors: The maximum number of queued incoming connections is set to 50 when the queue is full, new connections are refused. So when to use which? Use the first constructor for a small number of queued connections less than 50 and any local IP address available. Use the second constructor if you want to explicitly specify the maximum number of queued requests. And use the third constructor if you want to explicitly specify a local IP address to be bound in case the computer has multiple IP addresses. And of course, the first constructor is preferred for simple usage. For example, the following line of code creates a server socket and binds it to the port number Once a ServerSocket instance is created, call accept to start listening for incoming client requests: And the connection is represented by the returned Socket object. Read data from the client: Once a Socket object is returned, you can use its InputStream to read data sent from the client like this: So if you want to read the data at higher level, wrap it in an InputStreamReader to read data as characters: Use the OutputStream associated with the Socket to send data to the client, for example: Invoke the close method on the client Socket to terminate the connection with the client: Of course the server is still running, for serving other clients. A server should be always running, waiting for incoming requests from clients. In case the server must be stopped for some reasons, call the close method on the ServerSocket instance: The ServerSocket class also provides other methods which you can consult in its Javadoc here. Implement a multi-threaded server: So basically, the workflow of a server program is something like this: Once the first client is connected, the server may not be able to handle subsequent clients if it is busily serving the first client. Therefore, to solve this problem, threads are used: Hence the workflow is updated to implement a multi-threaded server like this: Java Server Socket Example 1: Time Server The following program demonstrates how to implement a simple server that returns the current date time for every new client. Server is listening on port And the following code is for a client program that simply connects to the server and prints the data received, and then terminates: If the client is on the same computer with the server, type the following command to run it: Mon Nov 13 Then the client terminates and the server is still running, waiting for new connections. Java Socket Server Example 2: The following server program echoes anything sent from the client in reversed form hence the name ReverseServer. Run this server program using the following command: The following program connects to the server, reads input from the user and prints the response from the server. Run it using the following command: Keep this first client program running, and start a new one. In the second client program, you will see it asks for input and then hangs forever. Java Socket Server Example 3: Now let run this new server program and run several client programs, you will see the problem above has solved. All clients are running smoothly. Let experiment the examples in this lesson in different ways:

Chapter 5 : IP Sockets in Java, Second Edition: Practical Guide for Programmers - PDF Free Download

TCP/IP Sockets in Java Second Edition The Morgan Kaufmann Practical Guides Series Series Editor: Michael J. Donahoo TCP/IP Sockets in Java: Practical Guide for Programmers, Second Edition Kenneth L. Calvert and Michael J. Donahoo SQL: Practical Guide for Developers Michael J. Donahoo and Gregory Speegle C# Practical Guide for

Programmers Michel de Champlain and Brian Patrick Multi-Tier.

Chapter 6 : TCP/IP Sockets in Java, 2nd Edition - O'Reilly Media

Note: If you're looking for a free download links of TCP/IP Sockets in Java, Second Edition: Practical Guide for Programmers (The Practical Guides) pdf, epub, docx and torrent then this site is not for you. racedaydvl.com only do ebook promotions online and we does not distribute any free download of ebook on this site.