

## 1: Soap Making Recipes, Tips & Tutorials - Teach Soap

*SOAP is an open-standard, XML-based messaging protocol for exchanging information among computers. This is a brief tutorial that introduces the readers to the fundamentals of SOAP before moving on to explain its various elements, encoding, and how SOAP is transported.*

This content is part of in the series: Understanding web services specifications, Part 1 [http: This content is part of the series: Understanding web services specifications, Part 1](http://) Stay tuned for additional content in this series. Before you start This tutorial series chronicles the building of a web services-based workflow system at a small fictional newspaper, the Daily Moon. It is for developers who want to learn more about the concepts that underlie web services, so that they can more efficiently create applications for them. Non-technical users will also some find value in these tutorials, as they explain concepts before jumping into programming. You should have a basic understanding of programming, and, if you want to follow along with the actual programming examples, a grasp of Java. About this series This tutorial series teaches the basic concepts of web services by following the exploits of the fictional newspaper, the Daily Moon, as the staff uses web services to create a workflow system to increase productivity in these turbulent times. This first part starts simply, explaining the basic concepts behind web services and showing you how to use SOAP, the specification that underlies most of what is to come, connecting the classifieds department with the Content Management System. Future installments in this series will build upon the basic concepts: Part two takes things a step further, explaining how to use Web Services Description Language WSDL to define the messages produced at expected by web service, enabling the team to more easily create services and the clients that connect to them. Part three finds the team with a number of services in place and a desire to locate them easily. In response, Universal Description, Discovery and Integration UDDI provides a searchable registry of available services at a way to publicize their own services to others. Interoperability is the key word in part six, as services from several different implementations must be accessed from a single system. Part six covers the requirements and tests involved in WS-I certification. About this tutorial This tutorial introduces you to the concept of web services, and to the team at our fictional Daily Moon Classifieds Department. You will follow along as the team integrates with an existing web services system, and witness the creation of a service. During the course of this tutorial, you will learn the following: How to install a web services implementation into an application server How to programmatically create a SOAP message. Programming examples are shown in Java using the Apache Axis2 project, but the concepts apply to virtually any other language and environment. Prerequisites In order to follow along with the code for this tutorial, you will need to have the following software available: Apache Geronimo or another application server. You will be creating various web services throughout the course of this tutorial, and you will need an application on which to run them. You can also use other application servers such as WebSphere Application Server. You can download Apache Axis2 at: This tutorial uses version 0. What are web services? SOAs have been around for a long, long, time. Originally they mostly consisted of middleware applications in which a single type of middleware owns, at the very least, both ends of the wire. Traditional applications In the beginning, there were computers. And it was good. Computers performed seemingly miraculous tasks, automating many of the things that people did by hand, starting with complex calculations, and moving to financials, and many other tasks. But traditional applications are "silos". You had the option to write batch processes to move data from one system to another, but that was no substitute for real-time integration. Distributed computing The next step in our evolutionary chain is distributed computing. Technologies such as CORBA, MTS, and Enterprise Java Beans EJB , provided a system that included a registry of sorts so that applications could find components with which they wished to interact, and then call these components as though they were located on the local machine. These systems were supported by middleware, or more specifically, message-oriented middleware, which provided both of these requirements. Applications could now be built in such a way that they could access resources on other systems, even if they were in different geographic locations. But there was still a problem. While applications were free to communicate anywhere within the system, it was still a closed system. At the very least, your

client application had to use the same technology as the server application. Also, the systems were not designed, as a rule, for access from outside of the individual organization that had created them. Web services

The next, almost inevitable link in this evolutionary chain is web services. This expands the world for your application to include anyone who can reach it over your network. It is a simple system, and as such, there are many aspects of enterprise-level computing that are not covered by it. Fortunately, many of these aspects have been taken into consideration, and have their own specifications to determine how this transaction should take place to incorporate many of the security and other aspects of traditional message-oriented middleware. There are other XML-based means for sending messages between systems, some of which are applicable to an enterprise environment, and some of which are not. For example, Amazon was one of the first Web-based companies to provide web services access to its system to the public. Has slight dent near the middle. Harder than a human head. Less than miles. Driven by my daughter until I took it away. It is just whatever the data happens to be. In this case, commands are sent to a system via XML such as shown in Listing 3. In some ways, they are. However, we are not talking about a simple application to display the weather on your web site. These capabilities are covered by additional specifications that have sprung up around SOAP-based web services, which makes SOAP a better choice for enterprise-level applications in the long run. Basic web services specifications

Web services specifications typically fall into two categories: The basic specifications are: It also details the way applications should treat certain aspects of the message, such as elements in the "header", which enable you to create applications in which a message is passed between multiple intermediaries before reaching its final destination. This tutorial will cover the SOAP specification. Web Services Description Language is a specification that details a standard way to describe a SOAP-based web service, including the form the messages should take, and where they should be sent. It also details the response to such a message. WSDL, when combined with the appropriate tools, enables you to create a call to a web service programmatically without ever actually knowing what the web service is looking for; the application can extract those details from the WSDL file and provide you with programmatic interfaces to use. Universal Description, Discovery and Integration is a standard that has undergone somewhat of a change since its initial inception. The idea was to provide a way for companies to register their services in a global registry, and search that global registry for services they may be interested in using. However, UDDI has taken hold as an internal registry of services and service information; part three of this series details its use. Those are the basics. There are also literally dozens of extended standards to make SOAP-based services more useful. Part four of this series covers WS-Security. This specification expands on WS-Security, enabling you to more specifically detail how and by whom a service can be used. Part five of this series covers WS-Policy. Although web services are supposed to be designed for interoperability, in actuality there is enough flexibility in the specifications that interpretations between different implementations can cause problems. WS-I provides a set of standards and practices to prevent the sorts of problems, as well as standardized tests to check for problems. WS-I is the subject of part six of this series. At the very least, enterprise-level computing requires you to compose multiple services into an overall system, and WS-BPEL provides a way to specify interactions such as branching and concurrent processing that are necessary for creating such systems. Other specifications that play an important role in web services are not covered in this series including WS-ReliableMessaging, which enables you to be certain that one and only one copy of a message has been received, and that it has definitely been received; WSRF, the Web Services Resource Framework, which enables you to use state in what is essentially a stateless environment; and Web Services Distributed Management WSDM , which discusses the issue of management of and using web services. You will create an application that creates a SOAP-based message, sends it to the service, and receives a response. Having done that, you will see how to create a service that responds to requests and sends a response of its own. The first step is to get the software in place. Why do you need a Web application server? Well, because you really are going to find it difficult to serve web services without one. A Web application server listens for requests, translates those requests into something the actual service can understand, and then does any necessary processing. Actually, it assumes a J2EE application server. Geronimo is small, easy to install, and easy to manage. Download the software see Prerequisites and extract the files into a target directory. Any directory is acceptable, but you want to avoid

those with a space in their name, such as "Program Files", "Documents and Settings", or their descendants. For example, the test installation for the tutorial uses e: To start the server, open a command prompt window and execute the following commands: It is not, however, advisable. The Apache Axis project has been making this task simpler for several years now, creating an environment in which it is easy to create and process web services. The software includes applications that help you create a web service from a plain object, create a Java object from web service, and process both. The Apache group has instituted a new version of Axis, Axis2, which takes all the work done on Axis and ratchets it up a notch by changing the architecture to allow it a greater degree of extensibility. This is important, because web service specifications are springing up all of the time. Make sure to download both the Binary Distribution and the War Distribution. The former will help in building clients, the latter in building services. To install Axis2 into the web server, copy the axis2. Verifying the axis to installation To make sure that everything has been installed properly, point your browser to http:

## 2: Java SOAP Web Services Tutorial - The Java Programmer

*It is important for web applications to be able to communicate over the Internet. The best way to communicate between applications is over HTTP, because HTTP is supported by all Internet browsers and servers. SOAP was created to accomplish this. SOAP provides a way to communicate between.*

Client The Client class of errors indicates that the message was incorrectly formed or did not contain the appropriate information in order to succeed. It is generally an indication that the message should not be resent without change. Server The Server class of errors indicates that the message could not be processed for reasons not directly attributable to the contents of the message, but rather to the processing of the message. The message may succeed if re-sent at a later point in time. Now imagine that you want to add some authentication information to the original message so the receiver can determine whether the sender has sufficient rights to execute the transfer. A way to do this would be to add the credentials information into the body as shown here: It also means that other applications in need of security must develop their own solutions to the problem; ultimately, interoperability suffers. For common needs such as security, it makes more sense to define standard SOAP headers that everyone agrees on. Then, vendors can build support for the extended functionality into their generic SOAP infrastructure and everyone wins. This approach increases developer productivity and helps ensure higher levels of interoperability at the same time. This is exactly the type of thing the SOAP extensibility model was designed to facilitate. Extensibility Most existing protocols make a distinction between control information e. SOAP is no different in this regard. Besides ease of use, the key benefit of the extensible Envelope is that it can be used with any communications protocol. As GXA matures and SOAP headers become standardized, it will become easier for developers to define rich application protocols, without having to reinvent the wheel each time. The Header element, like the Body element, is a generic container for control information. It may contain any number of elements from any namespace other than the SOAP namespace. Elements placed in the Header element are referred to as header blocks. As with other protocols, header blocks should contain information that influences payload processing. Hence, this is the right place to put something like a credentials element that helps control access to the operation: The following example illustrates how to require the processing of the credentials header: The processing model, however, allows for more interesting architectures like the one in Figure 3, which contains multiple intermediary nodes. Intermediary nodes make it possible to design some interesting and flexible networking architectures that can be influenced by message content. Roles are given unique names in the form of URIs so they can be identified during processing. When a SOAP node receives a message for processing, it must first determine what roles it will assume. It may inspect the SOAP message to help make this determination. Every SOAP node is required to assume the next role. Hence, when a SOAP message arrives at any given SOAP node, the node must process all mandatory headers targeted at the next role, and it may choose to process optional headers also targeted at the next role. In addition to next, SOAP 1. The following SOAP message illustrates how to use actor: MustUnderstand status code, and discontinue processing. The SOAP Fault element provides the faultactor child element to specify who caused the fault to happen within the message path.

### 3: SOAP Web Services Tutorial: Simple Object Access Protocol EXAMPLE

*Simple Object Access Protocol (SOAP) is a standard protocol specification for message exchange based on XML. Communication between the web service and client happens using XML messages.*

Pin6K Guest post by Lexie of Lexie: When I heard about soap felting, I knew I wanted to give it a try, and I knew just who to ask for help! Felted soap is simply soap wrapped in wool that has been matted together. It also removes the ever-present problem of slippery soap. Felting your soap makes it last longer, and the wool has antimicrobial properties. Remember, not all bars will be perfect. It is almost impossible to completely ruin anything with this project trust me, I tried. First, tightly wrap some wool around your bar of soap. The entire bar needs to be covered with multiple layers. You can add different color strips of wool at this point, too. Have fun and be creative! Next, you will wrap your soap in the netting or nylon stocking. Make sure to cover the entire bar. Once your soap is covered in the wool and netting, you will gently place the bar of soap in the hot water. Once your bar of soap is wet, gently move it around in your hands. Move it from hand to hand and slightly squeeze it. After a few minutes you will notice that the wool is shrinking and sticking to the soap. Bubbles will start to form which will help you with the process. If you need some extra bubbles, you can add some liquid castile soap to the bar. Make sure you rub all the sides of the soap. At this point, take the soap to the sink, and run it under cold water. Remove the soap from the netting and check to make sure the wool is sticking. If you are able to tear any pieces off, you need to keep felting. Squeeze the soap to make sure the wool is very tight around the soap. The final thing you want to do is rinse the felted soap with cold water then place on a towel to dry. I put the soap on a towel that was sitting on a cooling rack that was also sitting on a towel. It will take a few days to dry. Putting the soap in direct sunlight will speed up this process. To use your felted soap: Just use it as you would any soap-filled washcloth. Soak the felted soap with water and rub the wool directly on your skin. This is a great way for children to bathe themselves! Place the soap in a soap saver dish to dry between uses. Just like any other soap, you need to keep your felted soap as dry as possible between uses. This will make it last longer. The felted wool will shrink as your bar shrinks. The more wool you use on your bar of soap, the longer your soap will last. For those of you who want to see all of these steps in action, check out my YouTube tutorial: Not up for another DIY project, but still want some felted soap? Thanks for your support! Have you ever made felted soap before? Lexie is a follower of Jesus, the wife of the very talented Stephen McNeill, and a work-at-home mother of two. Her passions include spending time with friends and family, reading, traveling, and teaching. In an effort to live a more natural and frugal lifestyle, she began making and selling her own skin care products including lotion, lip balm, deodorant, sunlotion, diaper cream, and soap. She loves sharing these passions with others on her blog and helping other families catch the vision of living a more natural lifestyle.

## 4: Swirled Rose Soap Tutorial - Soap Queen

*SOAP is known as the Simple Object Access Protocol. In this tutorial, learn what is SOAP, elements of a SOAP message, envelope element, building blocks, communication model and practical SOAP Example.*

SoapUI is an open-source testing tool which can operate in cross-platforms. It has very simple User Interface which is very easy to handle by both technical and non-technical users. Using SoapUI tool, tester can automate both functional tests as well non-functional tests and can execute compliance, regression, security and load tests for Web APIs. Diagram below represents the protocols supported by SoapUI. In an application where there are web services or web APIs, SoapUI tool is a boon to testers as it helps to cover both functional as well as non-functional tests with no hassles as its user interface is very simple to use. As we know in functional testing we covers the functional or behavioral testing aspects of the web services and web APIs. Using SoapUI the tester can leverage below features: This the best feature for new testers to build the test automation by simply using drag and drop feature of SoapUI. This accelerates the development of scripts for test automation. Depending on the functional test scenarios, SoapUI supports advance scripting option which help in writing the custom scripts for tests automation scripts development. This offers flexibility of writing scripts within the tool. Hassel Free Environment Switching: SoapUI tool helps tester to debug their test scripts and permits to write data driven tests. Tester may put various check points to verify that their test is getting executed as expected. No doubt, depending on the above mentioned capability of the SoapUI tool and its robustness in writing the test automation scripts. Using SoapUI tool, the tester can validate the compliance tests and can automate it using tests scripts. Compliance tests mainly involves the source and its data authentication using licence certificates, etc. Regression testing is conducted to analyze any defect or malfunctioning introduced within the web services after new change is implemented with the current release or in simple words, it is used to find any side effects of the new change or fix. Tester can execute the test suits to ensure existing functionalities are not breaking and new change is right in place using SoapUI tool. Load testing is conducted to analyze the load that a web service or web API can withhold before it breaks. SoapUI has number of LoadUI agents over which load can be distributed and can analyze the various performance parameters. Leveraging the simple user interface and handling LoadUI agents of SoapUI tool, load testing becomes very simple and easy to operate. SoapUI has advance reporting system to capture various performance parameters for load testing. Also it permits performance monitoring for end-to-end system load testing. This kind of testing is supported by SoapUI to ensure authorization and authentically in request and response model of web services and web APIs. It scans and detects for any exposure of service parameters in the structured messages during cross site scripting. It scans and detects for any potential SQL injections that could harm database, thus securing database. SoapUI has the capability to scan and detect for huge documents within the message that could cause stack overflow. SoapUI has the capability to do number of scans and ensures the security of the web services and wen APIs. These scans are vulnerability scans, fuzzing scans and boundary scans to mitigate the potential erratic behaviour of the web services. SoapUI tool can be used in collaboration with number of popular building tools as follows: Apache ant uses the command line for building projects. Similarly, it can use SoapUI command line to execute the test suites. Hudson is an integration tool built on JAVA platform. By integrating SoapUI with Hudson, it has the benefit to discover bugs very early on the click of every commit of code from development team. Maven is a project management automation tool which is mainly used to build JAVA projects. It has capability to automate the build, version control and documentation from a central repository. Using Maven build, we can also build and execute SoapUI tests following number of simple commands.

### 5: 37 Homemade Soap Tutorials, Recipes, And Ideas You Can DIY At Home!

*SOAP is an acronym for Simple Object Access Protocol. It is an XML-based messaging protocol for exchanging information among computers. SOAP is an application of the XML specification. Although SOAP can be used in a variety of messaging systems and can be delivered via a variety of transport.*

This recipe is a fun blend of old and new. Soap Queen Recipe type: Cold Process Soap Serves: Lots of Lather Quick Mix 4. Sodium Hydroxide Lye Distilled Water 2 oz. Suit up for safety to prep lye. Wear long sleeves, pants, and closed-toe shoes. Put on safety goggles and gloves. Set up a scale and make sure it is set to ounces. In a large glass measuring cup, weigh out 4. In a separate large glass measuring cup, weigh out 4. Slowly add the lye to the water and gently stir until the lye has fully dissolved. Fully melt the entire bag of Lots of Lather Quick Mix in the microwave leave cap on until there are no chunks or cloudiness and give the bag a good shake. Weigh out 33 oz. In a small glass cup, weigh 2 ounces of Wild Rose Fragrance Oil. Dispersing the mica colorant in oil helps it mix into the soap batter more easily. Slowly add the lye solution to the bowl of Lots of Lather Quick Mix. Pulse the blender on and off to mix. Continue to pulse and stir for about 1 minute. The soap will become the texture of thin pudding. To one container of soap, add all of the dispersed Rose Pearl Mica. Use a whisk to mix in the color thoroughly. Pour dollops of white soap into the pink soap in different areas and various heights. Use a spoon to stir the soap times to create more swirls in the batter. Be careful to not overmix. Pour the batter into the mold and tap it firmly on the counter to release bubbles. Use a spoon to create texture on the top of the soap. If the soap is not holding its shape, allow it to sit in the mold for minutes, then try again. Sprinkle pink rose petals on the top of the soap. Place the soap in a warm room-temperature area where it will not be disturbed. Let soap harden in the mold for days. Pull two sides of the mold away from the soap. If the sides of the mold do not easily release, let it harden for another day. Carefully remove the soap and place on a cutting board. Using a sharp, non-serrated knife, cut the soap into bars. Lay the soap on its side for a cleaner cut. Allow the bars to cure on a well-ventilated shelf for weeks to create a firm, long-lasting bar.

### 6: XML Web Services

*Welcome to this tutorial course on SOAP web services in Java. We'll start with an introduction to web services. We'll understand what they are and how they a.*

Want to become a craft tool reviewer for DIYProjects. Enter your email address for a chance! July 20, By Rynna Machate 10 Comments Make your own natural homemade soap and never pay for this wash buddy again! Learn how you can make these DIY homemade soap with any of the tutorials, recipes, and ideas you will find as you read on below! This would be a great gift you can share with your family and friends. Introduce your little ones to the art of soap making with this easy and colorful homemade soap idea. What is even better about this homemade soap tutorial is the no-lye part which makes it an easy soap to make. You can check out the handmade soap recipe and the detailed process here , where you can take an idea for your future soap-making projects. With anti-aging and sun protection, sunflower oil is your go-to ingredient for a moisturizing soap. For this recipe, Earl Grey tea is at play. This recipe is also a melt-and-pour process without lye , so you can safely DIY your soap even with the kids around. Put all these benefits on your aloe vera homemade soap and enjoy! Why not combine it with goat milk and turn it into a handmade soap? In only 10 minutes, you get a creamy and sweet-smelling soap you can make right at home. It has also a lovely color or two color combination rather, which makes for a dreamy lathering. Make this delectable-looking soap in this tutorial which also comes with two more homemade soap recipes. You can relish their benefits outside of your body in this zesty homemade soap. This would be the perfect gift you can give this Fall! Follow the step-by-step tutorial here. You will love this minty poppy seed scrubby soap with two sides. One side will keep you cool and fresh while the other side will exfoliate your skin! See how you can make this interesting homemade soap here. Apparently, even a homemade soap will come out of a crockpot if you know what to do! Make this DIY glycerin soap with toy sharks inside. With this idea, you can put any toys you want in your homemade soap. A star-studded unicorn soap will definitely send them flying to the tub for a scrub. You will need to do a fun experiment of mixing up colors with the right stroke and swirl. The process is somewhat complex but the results are worth it! They might not be able to help it just by the looks of this watermelon soap! Melt glycerin soap, add your favorite scent, pour into molds, and scrub away! This homemade coconut soap recipe will also give you a plus with its exfoliating properties. Buttermilk Baby Soap Baby Soap: This honey oatmeal soap is easy to create, which even your girls can make. A great idea for a slumber party with the other teens! This tutorial will also set the stage for your soap-making innovations which you may one day put up for sale. This recipe, also have coconut, castor, hemp seed, and olive oil for a moisture powerhouse! Not only will the soap exfoliate, it will also disinfect with the loofah and tea tree oil combination. Make this DIY soap for everyone in the family. For a soap, you are always in contact with, it is best to go organic. So say bye-bye hotel soaps with harsh chemicals and say hello to natural homemade soap you made on your own. There you go, bath and body products aficionados! Delicious-smelling natural homemade soap recipes you can make on your own. With these ideas, we fully expect your home and yourself to be just as sweet-smelling! We hope you enjoyed these amazing homemade soap tutorials and recipes! Try them out at home and let us know how it goes. You can share your experience with us in the comments section below! This post was originally published on November 18, , and has been updated for quality and relevancy.

## 7: Understanding web services specifications, Part 1: SOAP

*This tutorial will cover the SOAP specification. WDSL: Web Services Description Language is a specification that details a standard way to describe a SOAP-based web service, including the form the messages should take, and where they should be sent.*

For example, there could be a web application designed in Java, another in .Net and another in PHP. But data exchange between these heterogeneous applications would be complex. So will be the complexity of the code to accomplish this data exchange. One of the methods used to combat this complexity is to use XML Extensible Markup Language as the intermediate language for exchanging data between applications. Every programming language can understand the XML markup language. Hence, XML was used as the underlying medium for data exchange. But there are no standard specifications on use of XML across all programming languages for data exchange. That is where SOAP comes in. We will look into further details on the SOAP protocol in the subsequent chapters. Below are some of the reasons as to why SOAP is used. When developing Web services, you need to have some of language which can be used for web services to talk with client applications. SOAP is the perfect medium which was developed in order to achieve this purpose. This protocol is also recommended by the W3C consortium which is the governing body for all web standards. SOAP is a light-weight protocol that is used for data interchange between applications. SOAP is designed to be platform independent and is also designed to be operating system independent. Hence, there is no sort of customization which is required to run the web services built on the SOAP protocol to work on the World Wide Web. This is the root element in the SOAP message. A Header element that contains header information. The header element can contain information such as authentication credentials which can be used by the calling application. It can also contain the definition of complex types which could be used in the SOAP message. By default, the SOAP message can contain parameters which could be of simple types such as strings and numbers, but can also be a complex object type. A simple example of a complex type is shown below. Suppose we wanted to send a structured data type which had a combination of a "Tutorial Name" and a "Tutorial Description," then we would define the complex type as shown below. All of the required elements of the structure along with their respective data types are then defined in the complex type collection. Below is an example of the SOAP body which actually works on the complex type defined in the header section. Here is the response of the Tutorial Name and Tutorial Description that is sent to the calling application which calls this web service. Whenever a client application calls a method in the web service, the web service will automatically generate a SOAP message which will have the necessary details of the data which will be sent from the web service to the client application. The next element is the SOAP body which contains the details of the actual message. Our message contains a web service which has the name of "Guru99WebService". Now, the above SOAP message will be passed between the web service and the client application. You can see how useful the above information is to the client application. The SOAP message tells the client application what is the name of the Web service, and also what parameters it expects and also what is the type of each parameter which is taken by the web service. This enables the client application which calls the web service to know when the SOAP message ends. The following points can be noted on the SOAP envelope element. It is absolutely mandatory for SOAP message to have an envelope element. Every Envelope element needs to have at least one soap body element. If an Envelope element contains a header element, it must contain no more than one, and it must appear as the first child of the Envelope, before the body element. The envelope changes when SOAP versions change. Below is an example of version 1. When a success is generated, the response from the server will always be a SOAP message. MustUnderstand - An immediate child element of the Header element, with the mustUnderstand attribute set to "1", was not understood. Client - The message was incorrectly formed or contained incorrect information. Server - There was a problem with the server, so the message could not proceed. So the application could have a specific error message for different business logic scenarios. Example for Fault Message An example of a fault message is given below. The error is generated if the scenario wherein the client tries to use a method called TutorialID in the class

GetTutorial. The below fault message gets generated in the event that the method does not exist in the defined class. This was the simplest type of communication, but it had a lot of limitations. In the normal RPC style communication, the client would just call the methods in its request and send the required parameters to the server, and the server would then send the desired response. The above communication model has the below serious limitations Not Language Independent – The server hosting the methods would be in a particular programming language and normally the calls to the server would be in that programming language only. Not the standard protocol – When a call is made to the remote procedure, the call is not carried out via the standard protocol. This was an issue since mostly all communication over the web had to be done via the HTTP protocol. Firewalls – Since RPC calls do not go via the normal protocol, separate ports need to be open on the server to allow the client to communicate with the server. Normally all firewalls would block this sort of traffic, and a lot of configuration was generally required to ensure that this sort of communication between the client and the server would work. To overcome all of the limitations cited above, SOAP would then use the below communication model The client would format the information regarding the procedure call and any arguments into a SOAP message and sends it to the server as part of an HTTP request. The server would then unwrap the message sent by the client, see what the client requested for and then send the appropriate response back to the client as a SOAP message. The practice of unwrapping a request sent by the client is known as Demarshalling. This topic will look at using the Microsoft. Net framework to build an ASMX web service. This type of web service supports both SOAP version 1. This WSDL document is required by the calling client application so that the application knows what the web service is capable of doing. In our example, we are going to create a simple web service, which will be used to return a string to the application which calls the web service. This web service will be hosted in an Asp. We will then invoke the web service and see the result that is returned by the web service. Visual Studio will also show us what the SOAP message being passed between the web service and the calling application. The first pre-requisite to setup our Web service application which can be done by following the below steps. Please ensure that you have Visual Studio installed on your system for this example. Step 1 The first step is to create an empty ASP. Once you click on the New Project option, Visual Studio will then give you another dialog box for choosing the type of project and to give the necessary details of the project. This is explained in the next step. The project has to be of this type in order to create web services project. By choosing this option, Visual Studio will then carry out the necessary steps to add required files which are required by any web-based application. Give a name for your project which in our case has been given as webservice. Then ensure to give a location where the project files will be stored. Once done you will see the project file created in your solution explorer in Visual Studio Just provide a name of Tutorial Service for the web service name file. Step 4 Add the following code to your Tutorial Service asmx file. This line of code provides a name for your web service file. This is an important step because it gives way for the client application to call the web service via the name of the web service. Normally a class file is used to encapsulate the functionality of a web service. So the class file will have the definition of all the web methods which will provide some functionality to the client application. Here [WebMethod] is known as an attribute which describes a function. The subsequent step creates a function called "Guru99WebService", but with the inclusion of this step of adding a [WebMethod] attribute makes sure that this method can be invoked by a client application. If this attribute is not in place, then the method can never be called by a client application. This function is a web service which can be called by any client application. We are using the return statement to return the string "This is a Guru99 Web service" to the client application. If the code is executed successfully, the following Output will be shown when you run your code in the browser. The output clearly shows that the name of our web service is "Guru99 Web Service" which is the result of giving a name for our web service. We can also see that we can to invoke the web service. If we click the Invoke button, we will get the below response in the web browser. The above output, It clearly shows that by invoking the web method, the string "This is a Gu99 Web service" is returned. Visual Studio also allows you to view the SOAP message request and response which is generated when the above web service is called. The SOAP request which is generated when the web service is called is shown below. The first part of the SOAP message is the envelope element which is what was discussed in the prior chapters.

This is the encapsulating element which is present in every SOAP message. This tells the client application that the web service being called returns an object of the type string. This is very useful because if the client application which otherwise would not know what the web service returns. Summary SOAP is a protocol which is used to interchange data between applications which are built on different programming languages. This makes it a perfect for usage within web applications.

### 8: Soapui Tutorial |Soapui Training |Web services testing

*Energy, Melt and Pour Soap Tutorial Dean Wilson. Loading Unsubscribe from Dean Wilson? Coconut Lime, Melt and Pour Soap Tutorial - Duration: Dean Wilson 52, views.*

### 9: Soap Queen - Tutorials on soapmaking, bath fizzies, lotions and more

*Tutorials on soapmaking, bath fizzies, lotions and more.*

*The house on foster hill Report from Grimes Creek After a Hard Winter Food for the Thoughtless Minds Great male comedians Unusual Suspects (Cartoon Network) Diligence (Caleb Ccolley) A record of Canadian historical portraits and antiquities exhibited by the Numismatic and Antiquarian Soc Choosing a Career in Hotels, Motels, and Resorts (World of Work (New York, N.Y.)) Slang and cant in Jerome K. Jeroms works Biblical geography United Nations peace operations The Black family in slavery and freedom, 1750-1925 Business, Work, and Benefits Icon and Iconostasis The thoracic level The discovery of America and the discovery of man Chicago citation style guide The Insiders Guide to Greater Lexington and the Kentucky Bulegrass Daily life in Papal Rome in the eighteenth century The baptists in Hunslet in the late nineteenth century E mujeres extraordinarias john macarthur Html5 canvas tutorial Ill take another one of those, please! (orgasms in pregnancy) Graham Wallas and the great society Henry Moore in La Jolla Leibniz to Des Bosses, 21 July 1707 Surgery at a Glance (At a Glance (Blackwell)) Prayer Committee (Revival Preparation Guidebook) The movie dialogue quiz book Michael Prendergast Remodelling: the critical importance of leadership Plumbing step-by-step Kenneth waltz theory of international politics chapter 5 Shooting . o i35 XI Accessories Epidemiologic studies of polyomaviruses and cancer Dana E.M. Rollison. Pregnancy Essentials The complete womans herbal Differentiated Instruction for Language Arts Steps in product design Art, activism, and oppositionality*