Bom dia novamente. Bem-vindo, todos ao Java EE6 Developing Web Services com as tecnologias JAX-WS e JAX-RS. O JAX-WS é para SOAP e o JAX-RS é para serviços da Web REST. Começarei introduzindo o curso e depois, é claro, depois disso, entrarei no conceito de serviços da web.

Então deixe-me ir em frente e começar com o curso em si, que é apenas o primeiro capítulo para uma introdução aos serviços da web. Na verdade, o que vamos fazer hoje, vamos fazer um pouco de revisão. Provavelmente vocês estão um pouco familiarizados com o conceito de XML. Então nós temos um capítulo só para fazer um pouco de revisão sobre XML. E então, vamos entrar, é claro, no conceito de uma API para serviços da web. Eu não sei se vocês ouviram falar do JAXB. JAXB é uma API que permite que você [INAUDÍVEL] objetos XML, então vamos mostrar isso a você. E depois, é claro, depois disso, apresentaremos os serviços da Web SOAP. Isso é o que vamos fazer hoje.

E começaremos entrando nos serviços da Web SOAP e, quando terminarmos com o cliente de serviços da Web SOAP, entraremos no REST e, em seguida, no cliente REST e, depois disso, entraremos na construção de serviços da Web SOAP. , abaixo, abaixo, acima e, em seguida, criando um serviço da Web RESTful na API JAX-RS e, em seguida, entrarei em mais capítulos depois, e discutirei todos eles.

Então, deixe-me começar pela introdução a este curso, que explica a necessidade de serviços da Web, procurando definir esses serviços da Web, observando as características desses serviços da Web, explicando o conceito de uma mensagem, um formato Nesse caso, XML ou JSON, vamos explicar os dois e por que o XML e o JSON identificam as duas principais abordagens para o desenvolvimento de serviços da Web, que são, a propósito, SOAP e REST. Estes são os dois principais. E explicando as vantagens de desenvolver serviços da Web em um contêiner Java EE.

Agora, quando falamos de um contêiner, nesse caso, seria um contêiner da Web ou, a propósito, você poderia ser uma Web e um contêiner EJB. Nós vamos falar sobre isso.

Portanto, os objetivos deste curso, que é, a propósito, um curso bastante atraente. É um curso que abrange as mais recentes e maiores tecnologias de serviços da web. E começaremos criando documentos XML usando as declarações de namespace e o esquema XML. Isso é praticamente uma revisão do conceito de XML, falando sobre XML, formato de XML, a sintaxe em XML, resolvendo conflitos de nome usando os namespaces, falando sobre validação de XML através de esquema XML e assim por diante.

Então, claro, falaremos sobre como produzir e consumir XML e JSON, entrar na criação de clientes de serviço da Web RESTful, entrar na criação de clientes de serviço SOAP usando a API JAX-WS. É claro, quando eu digo aqui, criando clientes de serviço web REST, se você der uma olhada aqui, nós não dissemos, usando o JAX-RS. E vou falar sobre isso quando chegar lá. Temos algumas APIs que permitem criar clientes de serviço da web RESTful. Uma é Java usando as conexões HTTP puras da interface do usuário, e a outra é a API Jersey.

Então, nós vamos obter - uma vez que nós construímos, é claro, um REST e clientes de serviço web SOAP, nós vamos construir também o serviço web REST usando o JAX-RS, e então, é claro, nós vamos construir um serviço da web SOAP usando o JAX-WS. No serviço da Web SOAP, vamos discutir algumas maneiras de criar esses serviços da web de baixo para cima e de cima para baixo. Vamos explicar o que é o uso de cada um deles, a vantagem e desvantagem de cada um deles, e assim por diante.

Então, depois disso, é claro, porque estamos lidando com a criação de serviços da Web no contexto de aplicativos corporativos e Java e, como você sabe, precisamos lidar com erros. Não é sempre rosado. De vez em quando, você tem problemas, e precisamos descobrir como lidar com erros em serviços da Web SOAP e erros em serviços da Web REST.

Depois disso, começamos apresentando o conceito de segurança de serviços da Web, no qual falamos sobre segurança de serviços da Web em geral. E então entramos na segurança do WS, que é uma API usada para proteger os serviços da Web SOAP. Mas também falamos de uma tecnologia muito boa. Eu não sei se você pessoal, alguém ouviu falar da tecnologia OAuth. Então vamos dar uma olhada nisso. E é uma fonte aberta, e vamos ver a implementação, a parte de implementação de Jersey. Isso será, a propósito, no final do curso.

Então vocês, praticamente, fomos apresentados, e então é a próxima coisa. Naturalmente, o público típico deste julgamento de curso é, senhoras e senhores, é, claro, um Java Standard Edition e o Java Enterprise Edition. E, claro, vou falar sobre isso um pouco mais tarde. Os pré-requisitos para este curso, seria bom saber sobre o Java SE 7. Vocês estão familiarizados um pouco com o Standard Edition Server? Então, falaremos um pouco novamente em algum momento e mencionaremos o que há de novo na SE 7.

**Part 2**

Portanto, em termos de implementação de serviços da Web, sabemos que os serviços da Web SOAP agora são suportados pela API JAX-WS. E, claro, para aqueles de vocês provavelmente se lembram, havia uma API antes do JAX-WS, e isso é chamado JAX-RS, mas o que devemos saber, senhoras e senhores, é que o JAX-WS não é compatível com JAX-WS RS E assim, se você tiver serviços da Web antigos que usam o JAX-RPC, o que você precisa fazer é precisar instalá-los, todas essas bibliotecas instaladas, porque o JAX-WS é realmente muito bom.

Na verdade, em muitos casos, chamamos comparar o JAX-WS ao JAX-RPC. Nós sempre dizemos que o JAX-WS é a varinha mágica. Realmente, faz um monte de coisas para você debaixo do capô. E vamos ver isso ao longo do curso quando falamos sobre os diferentes tópicos do JAX-WS, além de, além disso, quando fazemos os laboratórios.

Portanto, o JAX-WS faz parte da Standard Edition, mas não é um servidor HTTP de nível de produção. Então, lembre-se de uma coisa é que, e vamos falar sobre isso, porque realmente o servidor HTTP de nível de produção no Java Standard Edition é realmente para testar seu aplicativo. Se você vai fazer um POC, prova de conceito, isso é uma coisa boa. Mas não é um tipo de servidor de qualidade de produção. É por isso que você precisa de um servidor de aplicativos como, o que vocês têm? WebLogic ou GlassFish e assim por diante.

Nós temos, é claro, serviços web SOAP, temos alguns extras, e estes são o que chamamos de extensões WS, que não estão incluídas, e vamos falar sobre elas assim que entrarmos na API JAX-WS. Também discutiremos quais são os recursos dessas extensões, quais são as extensões JAX-WS, e por que elas são úteis. Nós discutiremos isso também.

Muito provavelmente vocês ouviram falar do projeto da pilha do Metro. É uma implementação JAX-WS muito popular e muito popular que usa o projeto GlassFish. Então, quando entramos, por exemplo, nas extensões WS e na segurança dos serviços da Web SOAP, discutiremos esse projeto de pilha do Metro, que agora chamamos de suíte da pilha do Metro, que é realmente bastante atraente. O WebLogic, é claro, a pilha de serviços da Web agora é baseada na pilha do Metro.

Então, essa é a implementação dos serviços da Web SOAP usando o JAX-WS. Também vamos dar uma olhada nos serviços da web RESTful, que são suportados pelo JAX-RS. E, a propósito, o JAX-RS não faz parte do Standard Edition. O JAX-RS, claro, cobre apenas os serviços, como eu mencionei para você anteriormente, e fala apenas sobre o, é claro, ele é usado apenas no lado do servidor.

No lado do cliente, ainda não temos o JAX-RS. Mas, é claro, temos algumas APIs que nos permitem criar clientes para conversar com os serviços Web RESTful, e essa seria a API de conexão HTTP de URL, e a segunda é a API Jersey. E quando chegarmos lá, discutiremos isso em detalhes.

Implementação de qualidade de produção de Jersey - que também é uma implementação de referência. Este é exatamente o que vamos usar para um cliente de serviço web RESTful.

Portanto, em termos de serviços da Web dentro do contêiner Java EE, bem, os serviços da Web dependem da comunicação HTTP. Isso seria praticamente o de facto, como eu disse - o tipo mais popular de representar, é claro, o protocolo de transporte para serviços da web. E, claro, existem diferentes tipos de servidores que podemos usar, e é aqui que podemos usar o servidor Tomcat ou GlassFish ou WebLogic. Todos estes são exemplos de contêineres.

Então, obviamente, alguns de vocês estão usando o Tomcat, alguns de vocês estão usando o WebLogic, alguns de vocês estão usando o GlassFish, e assim por diante. E, é claro, apenas um servidor de perfil completo do Java EE oferece suporte ao JAX-WS e ao JAX-RS, de certa forma, se você quiser dar suporte a tudo.

Agora, deixe-me explicar um pouco sobre esse perfil completo versus o que chamamos de perfil da web e deixe-me ir ao quadro. E se eu mostrar a você esse diagrama que passei cedo, esse diagrama realmente representa o que chamamos de perfil completo, o que significa que toda a especificação está incluída - a especificação EJB completa, a especificação completa do servidor, a especificação Java completa e adiante.

Mas em muitos casos, você pode não precisar de todo o perfil completo, toda a especificação EJB completa. Você pode precisar apenas do que chamamos de uma versão do EJB Lite, uma versão de subconjunto dos EJBs, o que significa que você usa apenas a parte do contêiner da Web do servidor de aplicativos.

E, claro, alguns de vocês podem fazer a pergunta, e os EJBs, Joe? Agora, isso não deve ser um problema, porque agora os EJBs podem ser incluídos internamente, especialmente a versão EJB Lite ou um subconjunto de EJBs pode ser incluído como um contêiner EJB incorporado dentro do contêiner da web. É por isso que, atualmente, usamos EJBs dentro do contêiner da web e colocamos todo o aplicativo corporativo em um arquivo WAR, em vez de ter um arquivo corporativo inteiro. Isso é muito mais simples, muito mais rápido, muito mais fácil de trabalhar e assim por diante.

Então, e continuamos com isso, novamente, dissemos que o Metro e o Jersey podem ser adicionados ao contêiner básico da web, como o Tomcat, para trabalhar com ele. E as chances, como alguns de vocês disseram, usam o Tomcat, de modo que, na verdade, por que você pode fazer, você aproveita todos esses dois projetos para, é claro, ter seu servidor web Tomcat, claro, neste caso, suporte a webservices.

É claro que os serviços da Web simplificam e fornecem esse tipo de wrapper neutro quanto à plataforma em torno da lógica de negócios, e é por isso que gostamos disso. Claro, como dissemos anteriormente, e eu mencionei isso antes e eu disse, se você quiser, se a sua empresa precisa de transações, precisa de segurança, precisa de multithreading, e precisa de concorrência e assim por diante, é por isso que eu disse que seria legal realmente fazer uso do servidor de aplicativos. E a razão é porque o seu servidor de aplicativos, que neste caso é essa camada, permite que você passe por uma configuração.

E hoje em dia, a propósito, a configuração você tem uma escolha. Você pode usar o que chamamos de descritores de implantação, provavelmente, se você tiver o web.xml e o ejb-jar.xml e assim por diante. Mas hoje em dia, melhor ainda, nem sequer usamos as tecnologias mais recentes do Java 6. Nem precisamos usar esses descritores de implantação para configurar alguns desses recursos, como transações, segurança ou persistência.

Na verdade, podemos usar o que mais, o que há de novo, o que foi introduzido para substituir esses longos descritores de implementação XML, anotações. Então, isso se tornou bastante atraente. E agora, isso é o que tornou o serviço Java EE quiteattractive para criar uma empresa em serviços da web, porque agora podemos configurar tudo isso, e eles nos dão esses serviços, fornecidos pelo aplicativo servidor de graça. É só uma questão de configuração.

Essa é a vantagem que você tem. E nós vemos, a propósito, vamos ver exemplos onde, na verdade, essas anotações são muito úteis para configurar recursos diferentes para os serviços da Web em nosso servidor de aplicativos WebLogic. É o que vamos usar, é claro, nos laboratórios.

Seguindo em frente, vamos continuar com esses conceitos. Novamente, estamos olhando, é claro, a outra questão é quais são as ferramentas de serviços web? E aqui, o que queremos fazer é usar ferramentas para testar nossos serviços da web. E, portanto, o serviço testingweb, ferramentas de teste, são, é claro, comumente usados ​​no desenvolvimento quando construímos nossos serviços da web, e há outros diferentes. Portanto, os servidores de aplicativos têm um mecanismo pelo qual podemos testar nossos serviços da Web, como o GlassFishor WebLogic. Vamos ver exemplos sobre isso.

Neste curso particular, nós também lhe damos uma outra ferramenta, que é a cURL. Esse é um tipo de cliente de linha de comando de uma ferramenta que permite testar alguns dos serviços da web. Mas também usamos extensões no navegador. O que estamos usando aqui é a extensão do cliente Firefox, e vocês vão instalar isso logo no começo em um dos primeiros laboratórios que vamos fazer.

E estes são apenas alguns dos quizzes - nós os temos no final de cada capítulo, e a ideia é apenas, neste caso, testar nosso conhecimento uma vez que vocês, nesse caso, passaram pela palestra. Então, por exemplo, aqui, qual protocolo é normalmente usado pelos serviços da web? E nós damos a você diferentes - e há algumas escolhas múltiplas baseadas no que você vai ler como respostas. Você verá que o HTTP é realmente o que é usado, e esse é o protocolo de hipertextreamento.

Outro pequeno questionário aqui, onde diz, quais das seguintes são características comuns de serviços web? E nós sempre lhe damos mais de uma resposta neste caso. E lendo tudo isso, você verá que o a realmente funciona bem, o que é um formato de intercâmbio de dados baseado em texto. É por isso que os serviços da Web na verdade são bastante atraentes, porque são neutros e interoperáveis. E o outro é uma descrição de serviço acessível ao cliente, o que significa que o cliente irá em frente e acessará uma descrição do serviço, o que permite, é claro, que ele crie um cliente de serviço da web.

E no final de cada capítulo, também falamos um pouco sobre alguns dos recursos. Estes são apenas diferentes tipos de links que permitem que você leia um pouco sobre a arquitetura de serviços da Web, a pilha do Metro, a pilha de Jersey, o tutorial do EE 6 e o ​​soforth. E isso é basicamente o que queríamos dizer sobre a introdução aos serviços da Web e a introdução ao site.

Part 3

Depois que terminamos a primeira palestra, vamos em frente e mostrar o que precisamos fazer para as práticas de palestra um. Isso, a propósito, seria usando o seu guia de atividades. Em geral, quando você faz o download do seu eKit, o que chamamos de eKit, quais seriam seus livros. Você tem um guia do aluno e você tem um guia de atividades.

O guia do aluno é para a aula. Seria bom ter o guia do aluno ao seu lado quando eu lecionar, porque assim, quando eu uso os slides e explico alguns dos conceitos, você pode ler, por exemplo, e, claro, fazer anotações sobre o que eu estou discutindo. Isso seria muito útil. E os guias de atividades são realmente para fazer os laboratórios.

Então, por exemplo, depois da primeira lição, vamos fazer uma prática para a lição 1, que é a configuração do WebLogic e dos serviços da web. Isso é o que vamos fazer. Então, na prática, vamos em frente e configurar nosso servidor WebLogic e, é claro, no NetBeans. E, no segundo, vamos criar e implantar o centro de serviços da Web.

Vamos implantar, nesse caso, vamos abrir um projeto chamado Calculate Our App, vamos implantá-lo e passar por todas essas tarefas. A propósito, os laboratórios sempre têm uma visão geral. É altamente aconselhável, a propósito, senhoras e senhores, ler a visão geral sobre os laboratórios e ler as suposições. Porque alguns dos laboratórios dependem de todos os laboratórios que vocês vão fazer.

E depois disso, as tarefas, se você seguir as tarefas passo a passo e ler todas as tarefas, não há dúvida de que os laboratórios funcionarão adequadamente sem nenhum problema. Eu também quero mencionar mais uma coisa. Se, por exemplo, você tiver que sair em alguma tarefa. Depois de alguma tarefa, vamos assumir que você está aqui, e de repente seu chefe te chama, ou você tem que sair por algum tempo, seria bom também fazer anotações de que você está nessa etapa. Desta forma, quando você voltar, você não vai pular um passo. Se você puder pular uma etapa, provavelmente o laboratório não funcionará.

Então, em primeiro lugar, vamos criar e implantar o aplicativo de amostra do serviço da web. Este seria, a propósito, um serviço web SOAP. E no próximo exercício aqui, vamos seguir em frente e fazer a mesma coisa.

Use as ferramentas de teste de serviço da web nas quais usaremos um cliente de teste de serviço da web, nesse caso, que é o cliente de teste do WebLogic. Também usaremos o cURL como linha de comando para testar serviços da web. E então vamos em frente e instalar a extensão de cliente restante para testar os serviços da Web através do navegador Firefox.

E assim você passará por todas essas etapas passo a passo. Esses passos são muito, muito elaborados e variam em detalhes, então não há como você ter um problema. Como eu mencionei para você, a chave, senhoras e senhores, é seguir, ler os passos, passo a passo, e segui-los. Não há dúvida de que você não terá nenhum problema em fazer os laboratórios.

Practice 1-2 Creating and deploying Web Service Sample Applications

ESTÁ BEM. Vamos dar uma olhada agora na prática 1.2 da lição um. E aqui vamos criar e implantar um aplicativo de exemplo de serviço da web. Então, novamente, seguindo as etapas do guia de atividades nesta prática. Eles querem que nós comecemos neles. E então queremos que abrimos o aplicativo da calculadora. Então vamos em frente e fazer isso.

E então o arquivo. Nós dizemos Open Project. E então, é claro, isso nos levará à unidade D. E então, da unidade D, vamos em frente e olhamos para os laboratórios. E então nos laboratórios eles querem que, neste caso, abram os exercícios.

E então meu laboratório 01, e claro, abre o primeiro, que é o aplicativo da calculadora. Então, destacamos isso e clicamos em OpenProject. Porque vai levar algum tempo, então dê uma olhada no escaneamento aqui. É apenas a maneira como o NetBeans abre o projeto no começo. Ele verifica todas as multas e todas as bibliotecas e assim por diante, relacionadas a este laboratório. É por isso que está levando um pouco de tempo.

Então dê um pouco de tempo até que tudo abra. E depois disso você é bom. ESTÁ BEM. E então estamos apenas esperando um pouco. Andboom Agora, é claro, está quase totalmente implementado.

Então, espere um pouquinho até a digitalização aqui - até que, é claro, desapareça. E você não vê mais isso. Agora isso é feito, eles querem que cliquemos com o botão direito. E, claro, neste caso, e implemente o aplicativo.

Então clique com o botão direito e selecione Deploy. Isso irá prosseguir e implantar o aplicativo. ESTÁ BEM. Então, estamos apenas esperando um pouco para ver se o aplicativo está implementado.

Vai levar o que vai levar. E, claro, no final, quando for implantado, você verá em verde que é feito com sucesso. E é isso que estamos fazendo agora. E, novamente, com esses laboratórios, normalmente leva um pouco de tempo, porque é um aplicativo corporativo. Então dê um tempo para, claro, fazer a coisa certa. E indo para implantar as aplicações e inicia o aplicativo e assim por diante.

E como você vê, está feito. Agora foi bem sucedido. E tudo está feito.

E para descobrir que seu aplicativo está realmente implantado, você clica na guia Serviços. E então você abre. Você precisa clicar sobre este ponto A ao lado do servidor de aplicativos. E então você abre aplicativos. E porque é um arquivo de guerra. E a razão por que é um arquivo de guerra é também o ícone, que é esse globo azul que você vê aqui, ou esfera.

E então, se você clicar em Aplicativos da Web, verá que está implantado. E isso é muito bonito. Se você quiser implantar, basta clicar com o botão direito do mouse e clicar em Implantar, e isso será desimplementado. Mas é claro que agora não queremos fazer isso.

Depois disso, o que eles querem que façamos é que eles nos querem agora para abrir o Firefox. E então eles querem que copiemos esse link que você vê aqui.

Então, se nós apenas queremos dizer, podemos fazer isso de duas maneiras. Podemos fazê-lo desta forma, onde, neste caso, voltando ao nosso aplicativo. Seria bom, a propósito, neste caso, apenas abrir o seu guia do aluno aqui na sua área de trabalho da máquina que eles lhe dão.

E assim, claro, esta é outra maneira de também ver a prática. Então estamos fazendo agora a prática 1.2. E eles querem que nós tomemos este link aqui, como você vê. E você pode copiar. E deixe-me destacar isso aqui. E depois, claro, copie.

E depois abri-lo no Firefox. Eu já abri o Firefox mais cedo. Então, vou em frente e cole aqui. E então, claro, não esqueça, eu apenas copiei o ponto com ele. Então eu não deveria estar fazendo isso.

E então, claro, isso, como você vê, foi o arquivo. Então este foi o arquivo que temos neste caso para o aplicativo. O arquivo sempre começa com a tag de definição, a tag raiz e termina com a tag raiz de definição.

E depois, claro, depois disso, tem os tipos, que são representados aqui. E, em seguida, as mensagens e o tipo de porta, que geralmente são mapeados para a classe ou a interface em Java. E então você tem a ligação.

Nós podemos dar uma olhada nestes. Esse foi o arquivo. Aqui está o serviço e a porta e assim por diante. Nós podemos ver isso, é claro, mais adiante nas diferentes palestras.

Então deixe-me minimizar esse cara aqui. E depois disso, claro, abra neste caso meu NetBeans. E o que eles querem nos todo agora está aberto o segundo projeto, que é, neste caso, um aplicativo cliente de calculadora. É o que eles querem que façamos. Isso seria no passo 5. É isso que eles querem. Então vou em frente e abro.

Então, novamente, vai levar o que vai levar, é claro, abrir. Então, haveria uma varredura, neste caso, apenas para abrir isso. Então, aqui eles querem que a gente clique com o botão direito e limpe e construa isso.

Então vamos em frente e fazer isso. Então, vai passar o WSDL com o arquivo. E essa é a primeira coisa que eles querem que façamos. Soobviously, como você vê aqui, aquele botão vermelho não está mais lá.

E então, depois de limpá-lo e construí-lo, nesse caso, só para ter certeza de que você sabe, está tudo limpo e construído, então eles querem, é claro, clicar com o botão direito neste projeto e executá-lo. Vai clicar com o botão direito neste projeto e executá-lo. E, claro, quando corre, vai

E, claro, devemos ser capazes de adivinhar, você deve receber um erro aqui no caso do erro 500. Vamos em frente e fazer isso, só assim você pode ver isso.

E se nós o rodarmos, então está se conectando. E logo nos mostrará o resultado, que, a propósito, seria um erro, porque é claro que o aplicativo não está implementado. Porque eu acabei de desdobrar isso.

Vamos dar uma olhada nisso. E vamos ver isso neste caso - então deixe-me apenas checar aqui que ele está realmente desimpedido abrindo o NetBeans, entrando nos serviços, e certificando-se de que é - então sim, obviamente, vai levar algum tempo para implementar. Mas, uma vez que ele não foi implementado, se você executá-lo, ele deverá fornecer um erro 500.

Nós também podemos, a propósito, neste caso, basta copiar este link. Ou eu posso clicar neste link diretamente. E isso me levará para, neste caso, a guia. Deixe-me apenas fechar esse cara aqui. Feche todas as abas. E então faça de novo.

E obviamente está me dando-- então, se eu clicar nisto, ele vai me dar um erro 500. Então, isso obviamente mostra que o aplicativo está desimpedido. E o aplicativo cliente, obviamente, neste caso, não conseguiu encontrar o aplicativo.

Então é isso que eles querem que façamos nisto. E então vamos continuar com essa prática. Claro que a próxima coisa que eles querem que façamos com este caso, você pode simplesmente voltar ao NetBeans. E desta vez eles querem que nós abrimos outro projeto. E este projeto é chamado Hello Rest.

Então isso é abrir um projeto. Esse seria o terceiro. Olá Mundo. Qual é realmente um serviço web RESTful. Nós vamos falar sobre serviços web RESTful de qualquer maneira.

E, obviamente, os serviços Web RESTful também são bastante interessantes. E aprendemos sobre eles mais tarde nas lições desse curso específico. Então, se eu clicar com o botão direito nesse cara, então eu praticamente abro este projeto. E, claro, agora mesmo o que eles querem que façamos é clicar com o botão direito do mouse nesse projeto e selecionar Executar.

Quando selecionamos Run, é claro que, nesse caso, implemente e execute-o. E vamos dar uma olhada nisso. E então vai abrir um navegador. E vai ligar, claro, neste caso, a linguagem de definição de aplicativos da web.

Então, vai demorar um pouco. Enquanto isso, deixe-me apenas clicar no NetBeans e mostrar que a aplicação está realmente implementada no WebLogic clicando no aplicativo da web. E aqui está o meu Hello World que já está distribuído.

Claro que agora a próxima coisa que queremos dar uma olhada neste caso é o navegador. E vamos dar uma olhada neste navegador. E vemos que aqui está o meu, neste caso, o meu navegador.

E se eu agora colocar, por exemplo, qualquer nome, por exemplo, Tom. E então aperte Enviar. Isso, claro, neste caso, diz que a próxima alteração foi feita com sucesso.

Agora o que eles querem que façamos neste caso, e voltando neste caso para o arquivo PDF. E esse seria o próximo neste link. Se eu clicar neste link, isso me dará o arquivo WADL. Então, clico em Permitir aqui.

E então, claro, deixe-me ir em frente e - ou eu apenas copiei o link. Deixe-me ir em frente e copiar o link também, e colocá-lo dentro do Firefox. Deixe-me ir em frente e fazer isso.

E aqui está o meu link, copie. E então deixe-me ir em frente e abrir o Firefox. E, claro, cole-o. E então aperte Return. Andhit Return. Claro, e isso me dá o aplicativo da Web e a linguagem eficiente para serviços da Web RESTful. Falamos sobre isso quando entramos nos serviços da web.

Portanto, a tag aqui é MS zero no aplicativo. Essa seria a tag raiz. E, claro, você tem um monte de outras tags que estão apresentando, neste caso, o recurso, a gramática, a resposta, a apresentação e assim por diante.

E isso é basicamente o que eles querem que façamos para essa prática. 1.2. Obrigado.