



06. Introdução ao Web Services

📅 Date	@24/11/2022
📁 Categoria	Java para Web
📖 Curso	Java Servlet: autenticação autorização e o padrão MVC

Tópicos

- Projeto da aula anterior
- O que é um Web Service?
- Download das bibliotecas JSON e XML
- Serviço com JSON
- Retornar HTML?
- Serviço com XML
- Para saber mais: Autenticação
- Download do projeto do cliente
- Cliente do serviço
- Bibliotecas
- Para saber mais: O que é API?
- Faça como eu fiz na aula
- O que aprendemos?

Retornar HTML?

Por que o nosso serviço web não deve devolver HTML?

- Uma página HTML possui toda definição da interface, informações desnecessárias para o cliente do *web service*.
- Quando falamos de *web services*, temos que dar suporte aos vários tipos de clientes, não apenas navegadores. Como o HTML possui toda estrutura da página, vêm muitas informações que talvez não sejam necessários para o cliente.

Para saber mais: Autenticação

No nosso exemplo sobre *web service*, não falamos sobre a autenticação, ou seja, o nosso serviço não tem proteção nenhuma e poderia ser utilizado por qualquer usuário anônimo.

Então como funciona a autenticação no Web Service?

Existem, como sempre, várias maneiras de fazer isso (e até existem padrões) mas vamos focar em uma solução comum.

Quando é feita a requisição do serviço, junto com os seus dados, deveria ser enviada uma identificação do usuário. Essa identificação normalmente é chamada de *API Key*. Baseado nessa chave, o servidor pode verificar a identidade e permissões. A chave então é única para cada usuário e deve ser reenviada em cada requisição (repare que isso é parecido com a `JSessionID`, só que não existe cookie nem `HttpSession`).

Da onde vem a API Key?

A chave de acesso normalmente é criada pela aplicação para cada usuário. No entanto, antes de usar o serviço, devemos criar uma conta no provedor do serviço para informar os nossos dados e a forma de pagamento. A partir daí, recebemos a chave que devemos utilizar em cada requisição para o serviço.

Bibliotecas

- **Gson** - Utilizada para retornarmos um JSON para o cliente.
- **XStream** - Utilizada para retornarmos um XML para o cliente.
- **HttpClient** - Utilizada para simular uma requisição no sistema.

Para saber mais: O que é API?

Quando você ouve a palavra *Web Service*, muitas vezes também se fala de **API (Application Programming Interface)**. Aliás, hoje em dia se fala mais em API do que em serviço. O que seria uma API?

Os grandes provedores de serviços web possuem vários serviços para oferecer diversas funcionalidades. Como desenvolvedor/desenvolvedora, precisamos saber como chamar esses serviços. Ou seja, precisamos saber o endereço, qual é o método HTTP, quais são os parâmetros ou quais dados enviar, qual formato e estrutura (XML ou JSON), etc. Tudo isso faz parte da definição da API do serviço.

Normalmente, o provedor oferece uma documentação extensa sobre a API e até serviços de teste para "brincar" e testar as requisições.

Para finalizar, saiba que o termo API é muito utilizado em vários contextos. No contexto de *Web Service*, a API define a interface das funcionalidades que o serviço oferece, mas já falamos nesse curso também sobre a API de **Reflection**, que é uma biblioteca Java (outro contexto). A API de Reflection define a interface Java (nome de classes e métodos).

O que aprendemos?

Nesta aula, focamos no mundo de *web service* e introduzimos esse tópico extenso. Aprendemos:

- Que um *web service* usa HTML, JSON ou XML como retorno
- Que um *web service* oferece alguma funcionalidade para seu cliente
- Que um *web service* é útil quando precisa oferecer uma funcionalidade para cliente diferentes
- Que para o *web service* não importa se o cliente foi escrito em Java, C# ou outra linguagem, pois usamos um protocolo e formatos independentes da plataforma
- Como gerar JSON no código Java através de GSON

- Como gerar XML no código Java através de XStream
 - Como escrever um web service através de um `HttpServlet`
 - Como criar um cliente HTTP a partir do código Java, usando a biblioteca `Apache HttpClient`
 - Como gerar JSON ou XML a partir do cabeçalho `Accept` da requisição
-