



## Faça como eu fiz na aula

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, siga os passos listados abaixo.

---

- 1) No pacote `br.com.alura.gerenciador.servlet`, crie a servlet `EmpresasService`, que estende `HttpServlet`, possui mapeamento para `/empresas` e o método `service`.
- 2) Dentro do método `service`, liste as empresas e leia o cabeçalho `Accept` da requisição, guardando o valor em uma variável e imprimindo-o.
- 3) Agora, verifique o conteúdo da variável criada acima. Se ela conter a palavra **xml**, você deve converter a lista de empresas para ser apresentada como XML, além disso, defina o tipo de conteúdo da resposta da requisição e devolva o XML para o cliente. Faça o mesmo para JSON, se a variável conter a palavra **json**. Se não conter nenhuma das duas palavras, devolva um JSON com uma mensagem dizendo que o tipo não foi encontrado.
- 4) Baixe [aqui \(https://caelum-online-public.s3.amazonaws.com/1001-servlets-parte2/06/libs-webservice.zip\)](https://caelum-online-public.s3.amazonaws.com/1001-servlets-parte2/06/libs-webservice.zip) um ZIP com os jars necessários e extraia-o. Em seguida, coloque os jars dentro da pasta **WebContent/WEB-INF/lib**.
- 5) Crie o cliente do serviço, criando o projeto Java **cliente-webservice**. Nesse projeto, crie o pacote `br.com.alura.cliente` e dentro dele, crie a classe `ClienteWebService`, que terá o método `main`.
- 6) Baixe [aqui \(https://caelum-online-public.s3.amazonaws.com/1001-servlets-parte2/06/libs-http-client.zip\)](https://caelum-online-public.s3.amazonaws.com/1001-servlets-parte2/06/libs-http-client.zip) um ZIP com os jars necessários e extraia-o. Você verá uma pasta **lib**, que deve ser colocada na raiz do projeto. Além disso, adicione os jars ao *Build Path* do projeto.

7) No método `main` da classe `ClienteWebService`, faça um POST para `http://localhost:8080/gerenciador/empresas`, com o cabeçalho `Accept` aceitando um JSON. Além disso, execute esse POST, retorne o seu conteúdo como `String` e imprima-o.

8) Para testar, você pode subir o Tomcat, executar a classe `ClienteWebService` e verificar o que será impresso no console.

## Opinião do instrutor

A classe `EmpresasService` ficará assim:

```
@WebServlet("/empresas")
public class EmpresasService extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void service(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        List<Empresa> empresas = new Banco().getEmpresas();

        String valor = request.getHeader("Accept");

        System.out.println(valor);

        if(valor.contains("xml")) {
            XStream xstream = new XStream();
            xstream.alias("empresa", Empresa.class);
            String xml = xstream.toXML(empresas);

            response.setContentType("application/xml");
            response.getWriter().print(xml);
        }
    }
}
```

```
} else if(valor.endsWith("json")) {  
    Gson gson = new Gson();  
    String json = gson.toJson(empresas);  
  
    response.setContentType("application/json");  
    response.getWriter().print(json);  
} else {  
    response.setContentType("application/json");  
    response.getWriter().print("{\"message':'no content'");  
}  
}  
}
```

[COPIAR CÓDIGO](#)

E a classe ClienteWebService ficará assim:

```
public class ClienteWebService {  
  
    public static void main(String[] args) throws Exception {  
  
        String conteudo = Request  
            .Post("http://localhost:8080/gerenciador/empresas")  
            .addHeader("Accept", "application/json")  
            .execute()  
            .returnContent()  
            .asString();  
  
        System.out.println(conteudo);  
    }  
}
```

[COPIAR CÓDIGO](#)

Continue com os seus estudos, e se houver dúvidas, não hesite em recorrer ao nosso fórum!