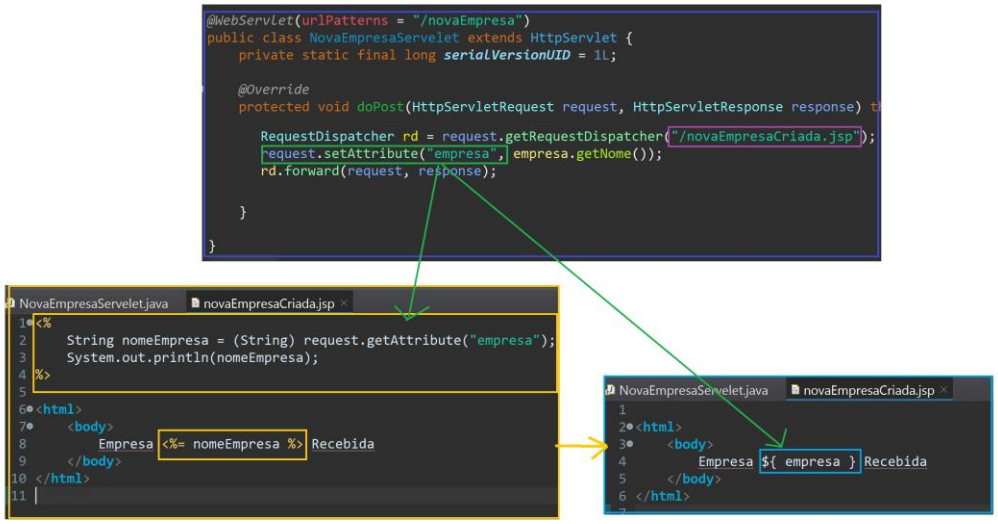
Java – *Servelet – Expression Language*

Eu achei os **Scriplets** bem esquisitos e pelo jeito eu não era o único. Visando simplificar um pouco da complexidade que é o **scriptlets** foi introduzido o uso de ***"expression language".***

A **sintaxe da expression language** é bem comum em várias linguagens na real, ela é aquele role de usar **"${}"**, onde **tudo que estiver dentro das chaves** pode tentar ser interpretado e mostrado na tela, isso vale para ***operações simples* ou mostrar o *valor de variáveis***.

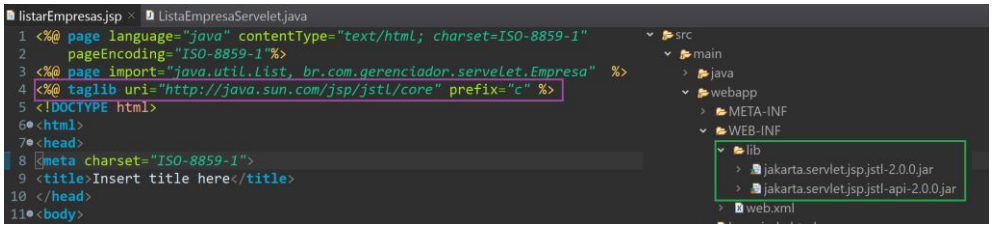
A **expression language** ajuda muito a reduzir a complexidade/verbosidade que uma JSP pode conter, por exemplo :



Java – *Servelet – Expression Language - JSTL*

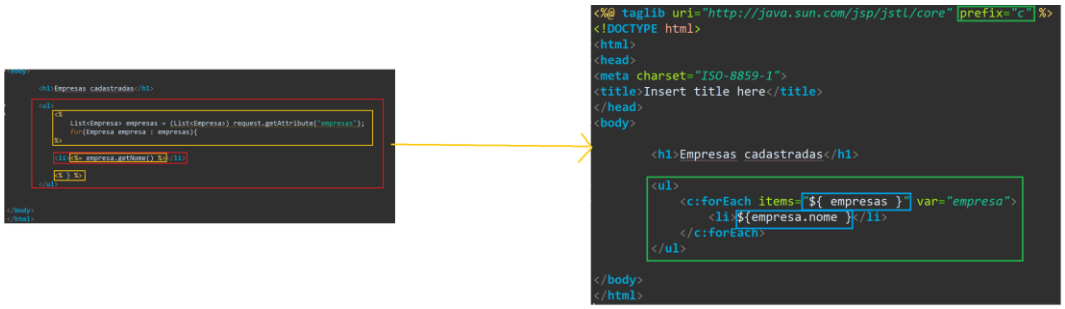
Expression language já ajuda a escrever códigos mais amigáveis em arquivos **JSP**, no entanto ela não tem o poder de fazer o mesmo com iterações *(na verdade, ta mais pro tomcat)*. Isso acaba gerando umas **iterações** bem estranhas como essa, onde eu preciso intercalar **scriptlets** com **tags HTML**:

Obviamente a comunidade também não curtia isso, então surgiu **algumas bibliotecas** para facilitar a escrita de **código java** em **JSP**, na verdade, deixá-lo mais amigável. Uma **dessas lib foi a JSTL**, que como uma biblioteca precisa **ser baixada no projeto** e **importada no JSP**:



Java – *Servelet – JSTL iteração*

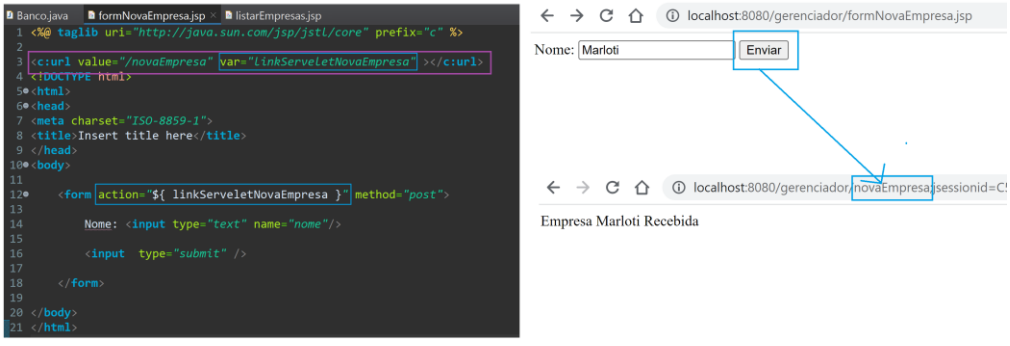
Com a **JSTL** importada eu posso fazer um ***"for"*** usando uma **tag html** combinando com **expression language**, o que vai transformar todo aquele **scriptlet** zuado em algo mais semântico e simples. Onde eu referencio o **prefixo da JSTL** com uma tag específica:



Java – *Servelet – JSTL definir caminhos*

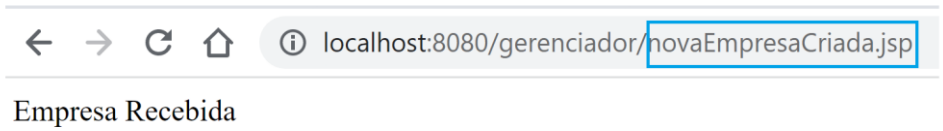
Outra tag bem útil do **JSTL** é a de **url**, nela podemos **definir um caminho** e **armazenar em uma variável**, dessa maneira eu posso controlar o fluxo de algo sem precisar me preocupar com o contexto ***(geralmente a pasta root do projeto é o contexto).***

Por exemplo, esse **formulário chama um método post** no **caminho da url armazenada**:



Java – *Servelet – JSTL condicionais*

Dependendo do fluxo de interação da aplicação, algumas informações precisam ser checadas. Por exemplo, se eu acessar a página de **novaEmpresa** diretamente sem passar nenhuma empresa, eu vou ter o seguinte resultado:



Mas não era isso que eu queria, então através do **core do JSTL** eu posso definir um comportamento **baseado em uma condição**: