

Utilizando o settings.xml do Maven no CronApp

O arquivo settings.xml possui elementos que são usados na configuração e execução do Maven, como no pom.xml, porém este não devem ser empacotados em qualquer projeto ou distribuídos para um público. Esse arquivo incluem valores que direcionam para repositórios locais, servidores de repositórios remotos alternativos e informações de autenticação.

Pré-requisitos

Antes de começar a seguir os passos do tutorial é preciso ter certeza de que se tem um ambiente minimamente preparado para reproduzir o exemplo. Abaixo estão os requisitos principais.

Requisitos:

1. Projeto do tipo web criado. Caso haja dúvidas de como criar esse tipo de projeto acesse o link ([Criando Projeto Web](#)).
2. Ter conhecimento sobre o Maven. Para mais informações, acesse o link ([Maven](#)).
3. Ter conhecimento sobre as configurações do settings.xml, veja a documentação de referência em ([Settings Reference](#) e [Settings](#)).

Visão geral: exemplo

Nesse tutorial iremos aprender a criar o arquivo settings.xml do Maven, permitindo, o download de bibliotecas privadas com acesso restrito.

Nesta Página

- Pré-requisitos
- Visão geral: exemplo
- Resumo do settings.xml
- Criando o arquivo settings.xml
- Chamando biblioteca com acesso privado
 - 1. Chamada no pom.xml
 - 2. Configurando o settings.xml
- Baixando as dependências

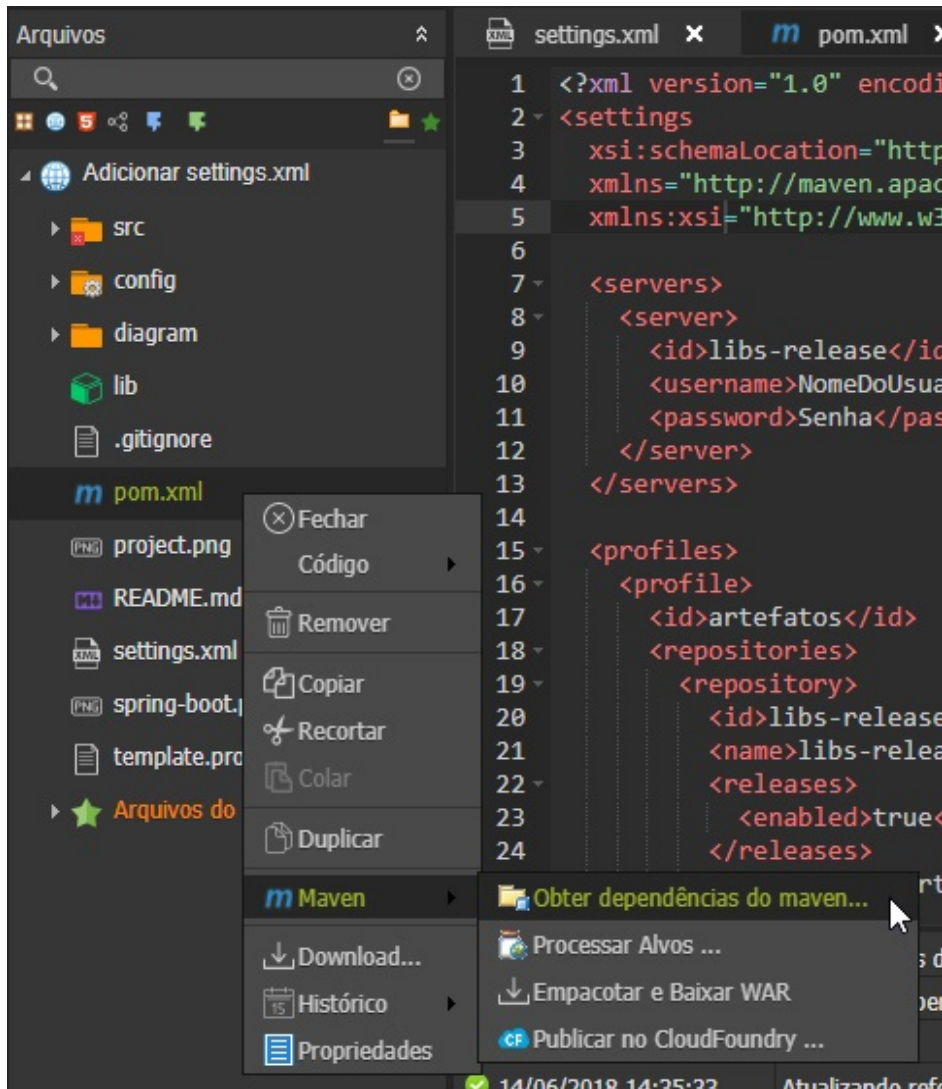


Figura 1 - Obtendo dependências Maven com repositórios privados e acesso restrito

Resumo do settings.xml

A estrutura XML do settings está organizada de forma a facilitar a configurações no Maven, abaixo segue um resumo simples da sua estrutura.

- **"Elementos da raiz"**: tags de configuração do usuário e sistema;
- **proxies**: referente a configurações da requisição do proxy;
- **servers**: contém informações necessárias para as configurações do servidor;
- **mirrors**: espelho para download de determinadas dependências alternativas;
- **profile**: modifica alguns parâmetros do ambiente no processo de build;
 - **activation**: chaves de ativação de um perfil;
 - **so**: ativa os atributos de um sistema operacional;
 - **property**: usada para ativar um perfil;
 - **file**: especificação de arquivo usado para ativar um perfil;
 - **repositories**: estabelece conexões com o repositório remoto;
 - **releases**: políticas de download;
 - **snapshots**: políticas de download;
 - **pluginRepositories**: contém as informações necessárias para estabelecer conexões com o repositório remoto.

Saiba mais

Veja a estrutura XML completa e em detalhes no link [settings](#).

Criando o arquivo settings.xml

Por padrão, o arquivo settings.xml não é gerado junto ao criarmos um projeto no CronApp. Nesse caso, o programador deverá criar o arquivo no diretório raiz do seu projeto.

Para criar o arquivo na raiz do projeto, clique com o botão direito do mouse no nome do projeto, na árvore de arquivos, selecione **Novo Outros** e insira o nome do arquivo "settings.xml", como na Figura 2.

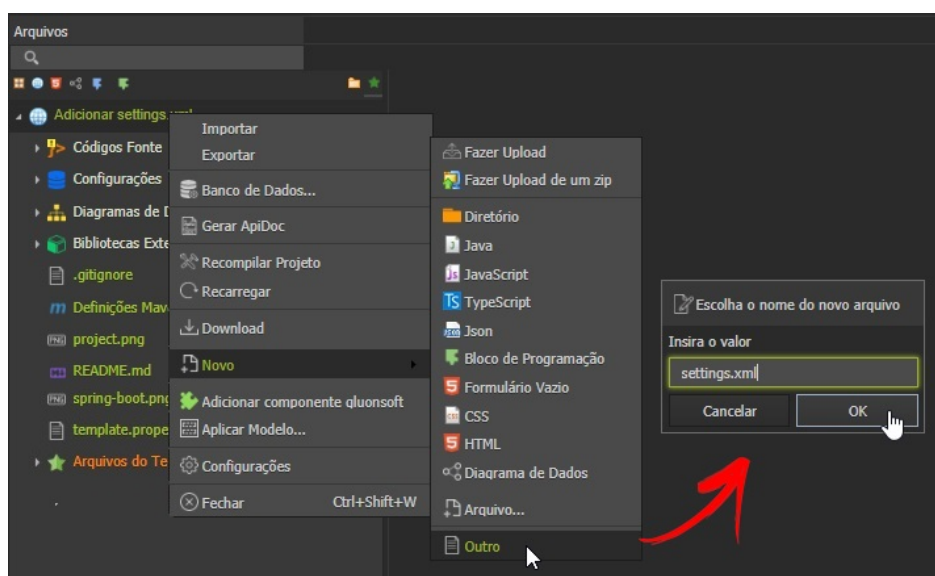


Figura 2 - Criando o arquivo settings.xml na raiz do projeto.

Chamando biblioteca com acesso privado

1. Chamada no pom.xml

No Maven, o arquivo pom.xml (ou Definições Maven) é onde são colocamos todos as bibliotecas que são baixadas no nosso projeto pelo Maven, no caso de bibliotecas privadas com acesso restrito, isso também deve acontecer, porém os dados de autenticação ficarão no arquivo settings.xml.

Abra o arquivo pom.xml (ou Definições Maven) e adicione os dados do repositório privado dentro da tag <repositories>. No exemplo da figura 3 estamos chamando os dados de uma dependência privada fictícia.

```
49 - <repositories>
50 -   <repository>
51 -     <id>br.com.cronapp</id>
52 -     <name>dependencies-api-rest-cliet</name>
53 -     <url>https://artefatos.cronapp.io/libs-release</url>
54 -   </repository>
55 - </dependencies>
```

Figura 3 - Adicionando os dados do repositório no pom.xml.

Mais sobre

Para mais informações sobre o pom.xml e como configurá-lo, acesse a documentação oficial "Pom reference".

2. Configurando o settings.xml

De volta ao arquivo settings.xml, abra o arquivo que está na raiz da árvore de arquivos (Figura 4) para começar a editá-lo.

Figura 4 - Localização do arquivo criado.

vamos adicionar o código abaixo, com as informações mínimas necessárias para permitir o acesso a uma dependência privada com acesso restrito. Nela estamos inserindo informações como nome do diretório, localização, nome do usuário, senha e algumas outras informações.

```
settings.xml
<?xml version="1.0" encoding="UTF-8"?>
<settings
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.1.0
http://maven.apache.org/xsd/settings-1.1.0.xsd"

  xmlns="http://maven.apache.org/SETTINGS/1.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <servers>
    <server>
      <id>libs-release</id>
      <username>NomeDoUsuario</username>
      <password>Senha</password>
    </server>
  </servers>
  <profiles>
    <profile>
      <id>artefatos</id>
      <repositories>
        <repository>
          <id>libs-release</id>
          <name>libs-release</name>
          <releases>
            <enabled>>true</enabled>
          </releases>

          <url>https://artefatos.cronapp.com.br/libs-release</url>
        </repository>
      </repositories>
    </profile>
  </profiles>
  <activeProfiles>
    <activeProfile>artefatos</activeProfile>
  </activeProfiles>
</settings>
```

Saiba mais

Caso tenha algum dúvida sobre o uso de algumas das tags, favor acessar a documentação oficial em [settings.xml](#).

Baixando as dependências

Para baixar as dependências, salve todos os arquivos editados, clique com o botão direito do mouse no arquivo pom.xml (ou Definições Maven) e selecione **Maven Obter dependências do maven**. (Figura 5)

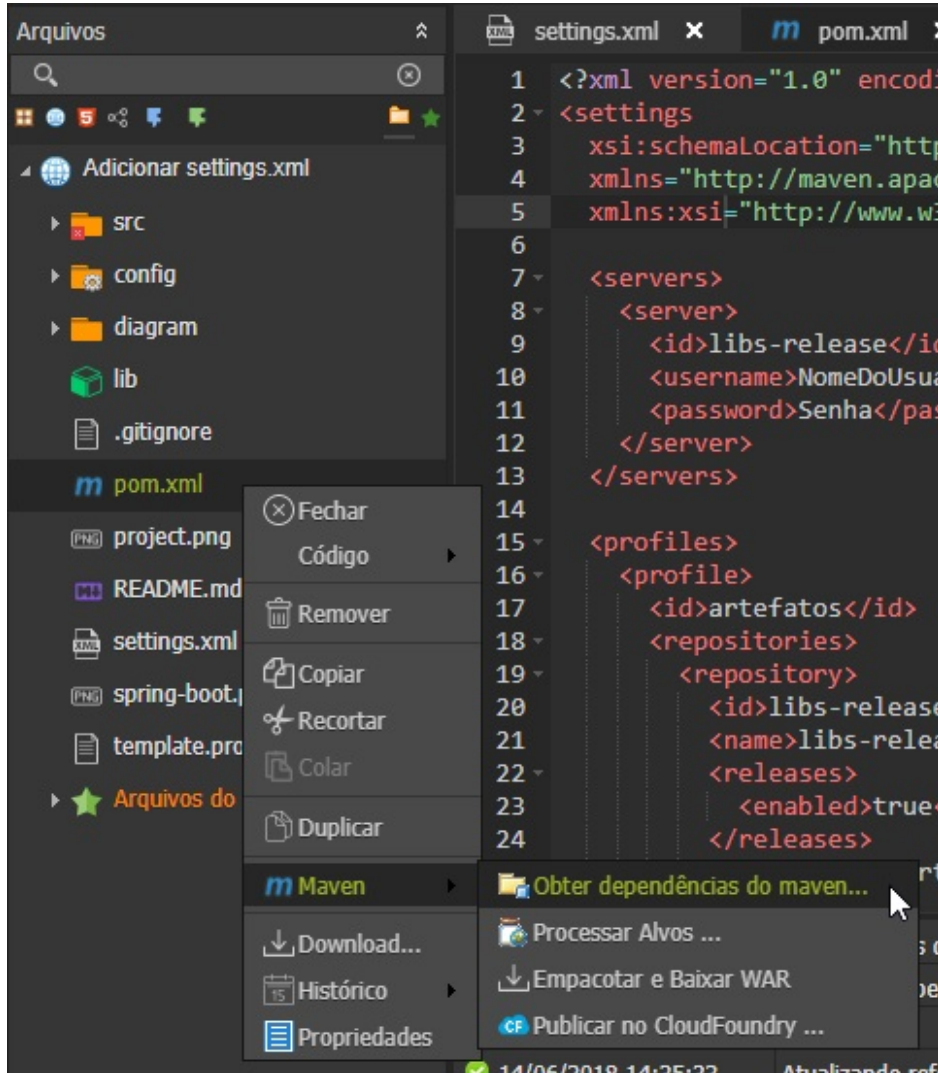


Figura 5 - Obtendo dependências do maven.

Após baixar as dependências, a mensagem "Dependências do Maven obtidas com sucesso." deve aparecer no console, confirmando o êxito na ação, como mostrado na Figura 6.

Figura 6 - Dependências obtidas com sucesso.