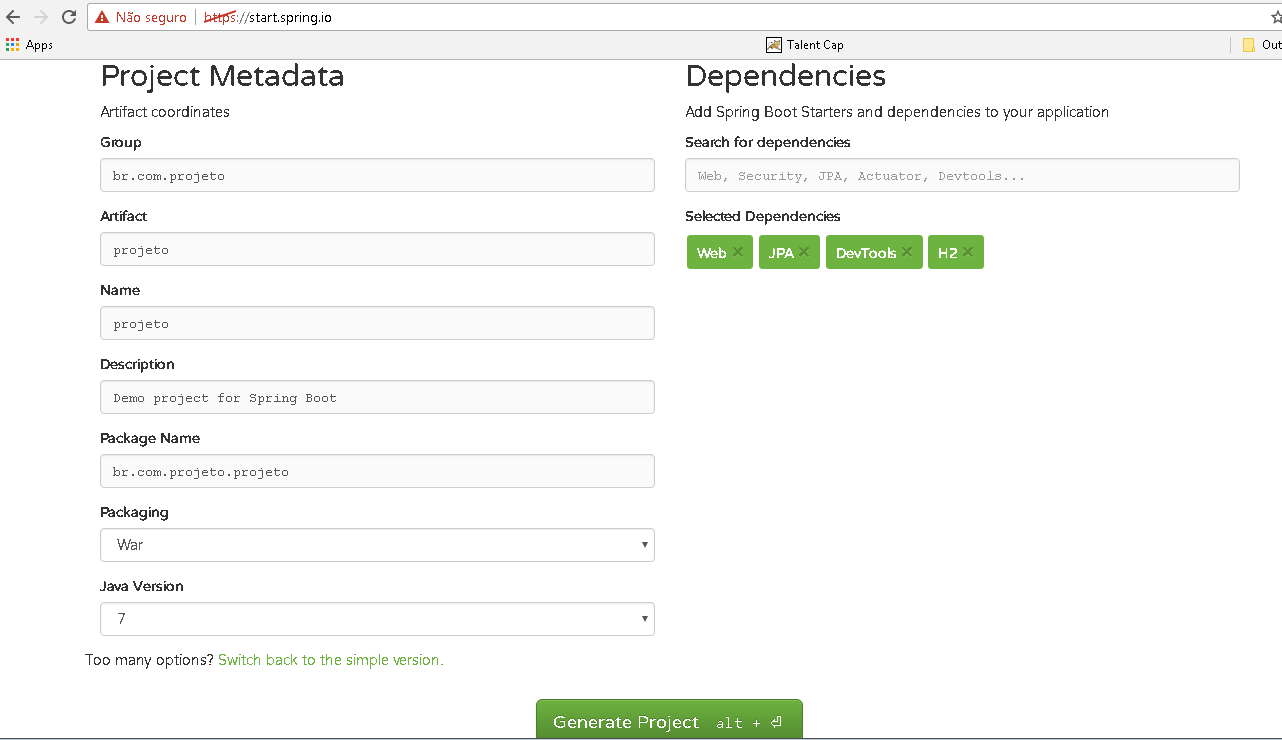
## Gerar o projeto Spring Boot

Neste exemplo usei o banco de dados em memória (H2), DevTools (ferramentas que auxiliam no desenvolvimento) e Web para criar os serviços REST com as anotações do spring boot (ao se usar o Jersey não selecione esta dependência).



**Usar a versão 1.5.10 do Spring Boot, pois a versão 2.0.0 usa a JDK 8 que não é compatível com o WAS 8.5**



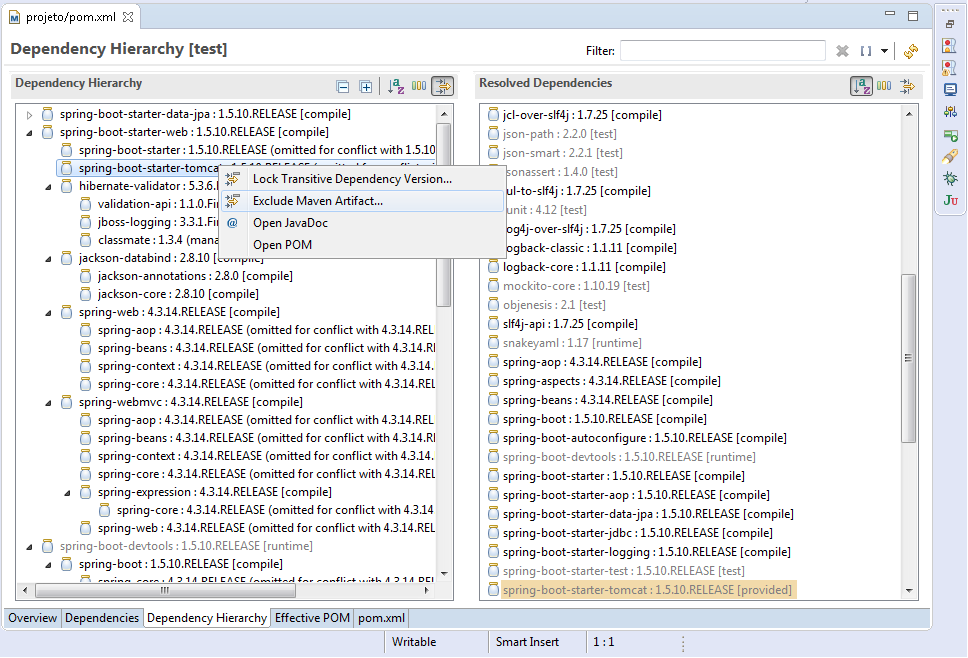
## Configuração do projeto

Importar o projeto no Eclipse (Existing maven Project)

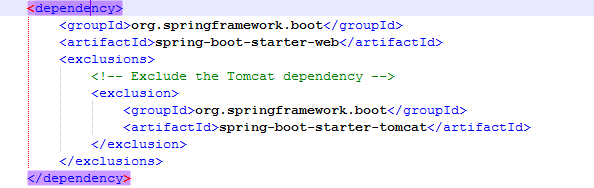
Configurar o pom.xml conforme descrito abaixo:

**É necessário remover o tomcat embarcado como dependência do projeto starter-web**

Removendo pelo Eclipse:



Removendo diretamente pelo POM:

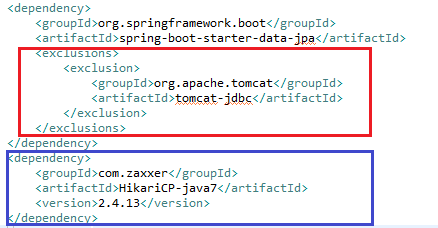


**----------------------------------------------------------------------------------------------------------------------------------**

**Recomendo usar a biblioteca HikariCP ao invés da biblioteca padrão para esta versão do Spring Boot**

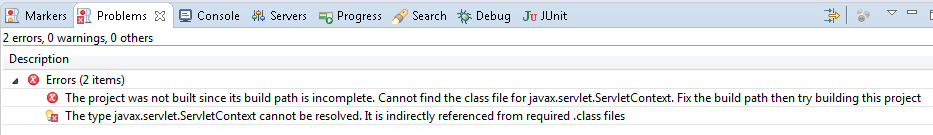
**(Tomcat JDBC). A biblioteca HikariCP se tornou a padrão na versão 2.0.0 do Spring Boot devido a uma performance melhor e menos bugs do que os contatados no Tomcat Jdbc, para trocar uma biblioteca por outra siga os passos abaixo:**

Remover a dependência do tomcat-jdbc do projeto starter-jpa, e adicionar a dependência do Hikari-CP (versão que usa a JDK7):



**----------------------------------------------------------------------------------------------------------------------------------**

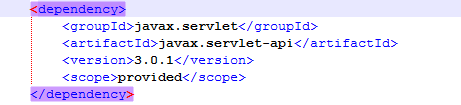
A partir deste ponto, o projeto apresenta um erro de compilação:



Este erro ocorre, pois o projeto sendo EE requer um container que implemente Servlets, como removemos o tomcat embarcado, o projeto não reconhece mais qual é o container que será usado.

**É necessário adicionar uma nova dependência de Servlet** informando que vamos alguma solução que implementa a especificação de servlets (javax.servlet-api) e esta dependência estará disponível quando a aplicação for executada (scope = provided).

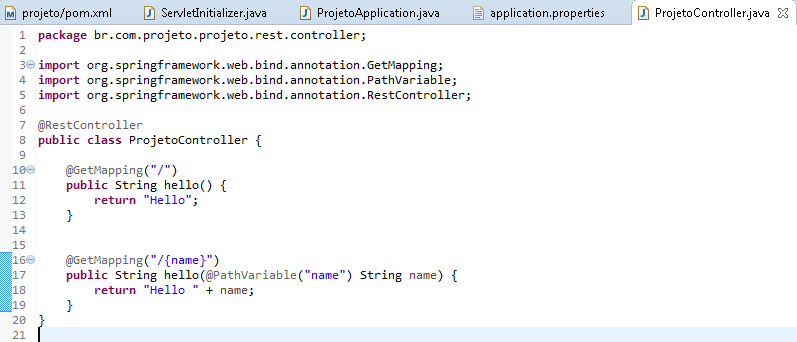
(No nosso caso tem que ser a versão 3.0.?, pois a versão 3.1 ou maior não funciona no WAS 8.5)



# Após estes passos já é possível realizar o deploy da aplicação.

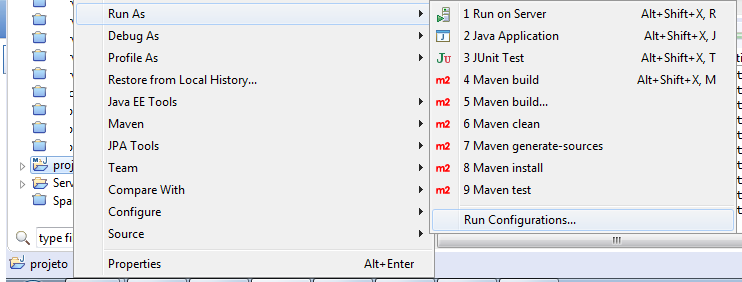
## Criando um serviço REST para testar a aplicação:

Criamos um recurso simples para testar a aplicação após o deploy:

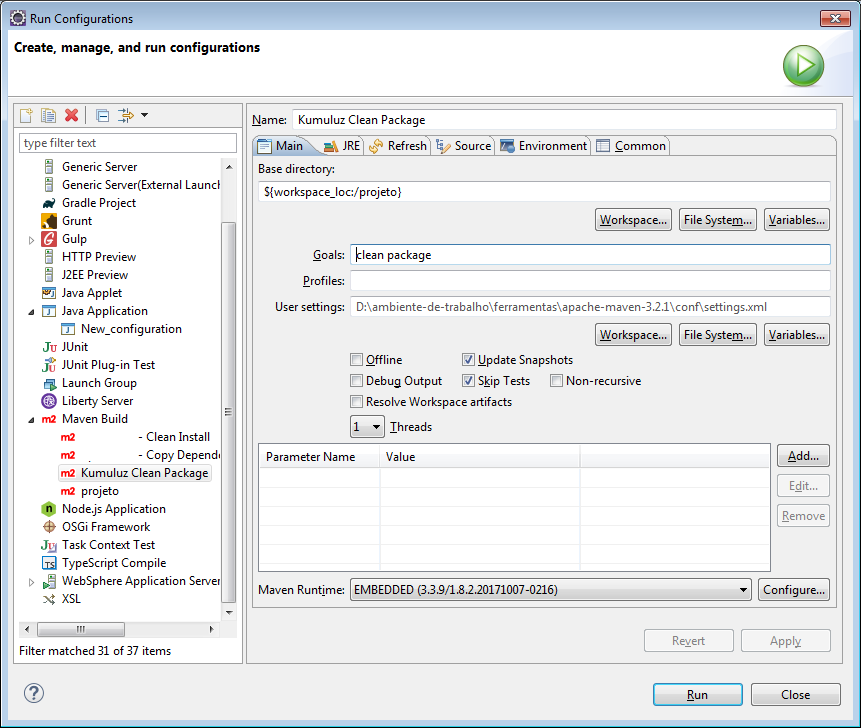


Geramos nosso WAR para realizar o deploy:

Projeto > Run As > Run Configurations

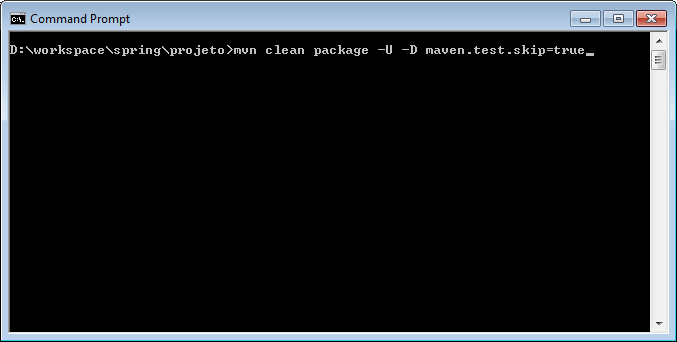


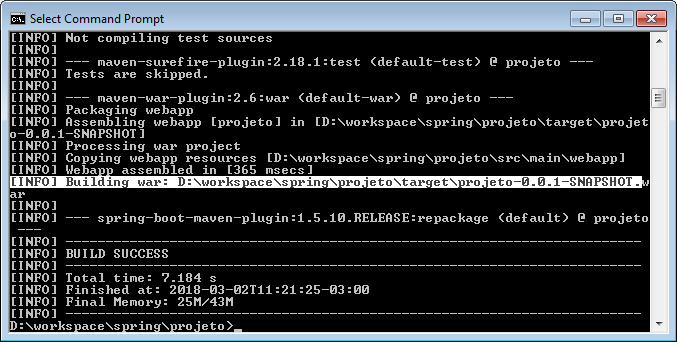
Escolher a pasta do projeto, executar o goal Maven: “clean package”, forçar o update das dependências e pular os testes:



Este processo de compilação pode ser executado também pela linha de comando do maven na pasta do projeto (onde se encontra o pom.xml):

**mvn clean package -U -D maven.test.skip=true**



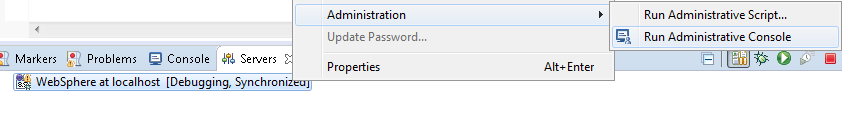


O War é criado (BUILD SUCCESS) e é informado onde ele se localiza:

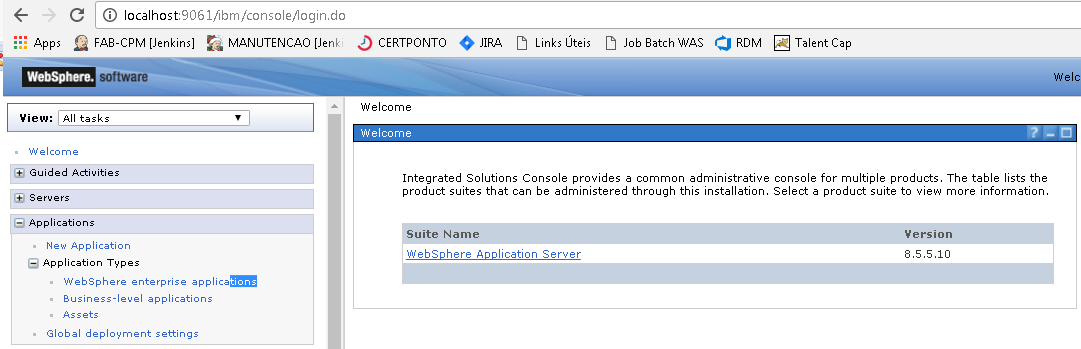
**[INFO] Building war**: D:\workspace\spring\projeto\target\projeto-0.0.1-SNAPSHOT.war

## Realizando o Deploy da aplicação:

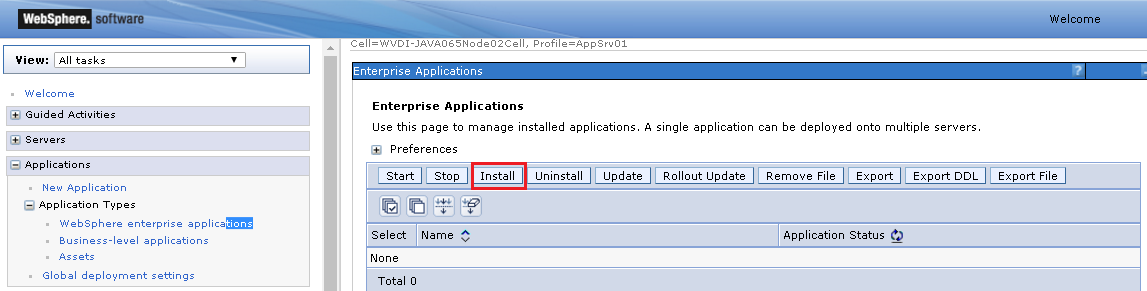
Acessar o console do WAS:



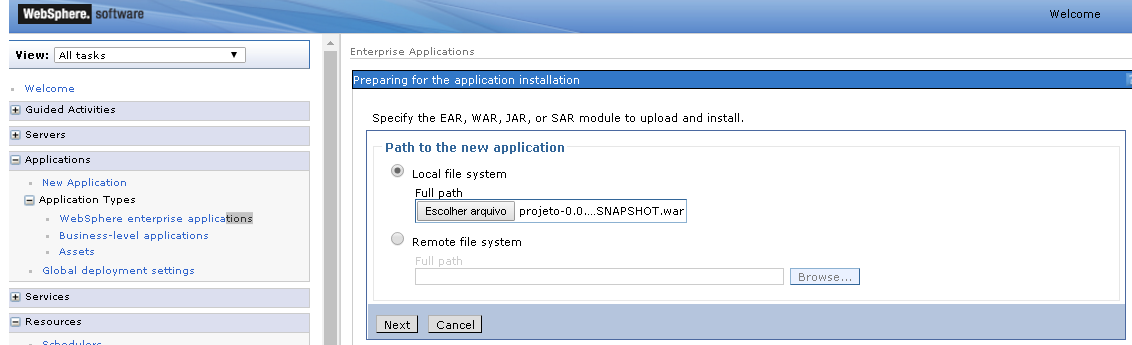
Applications > Aplication Types > WebSphere enterprise applications



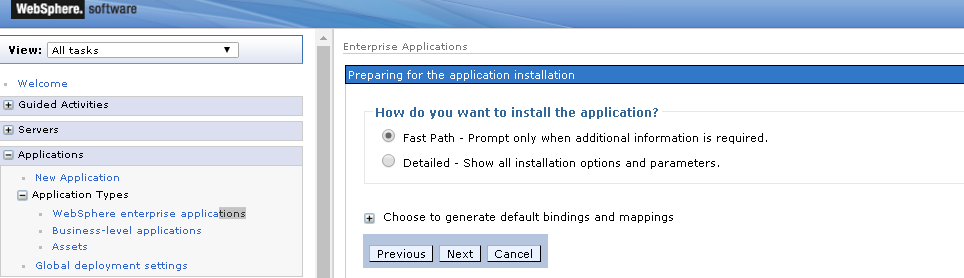
Install



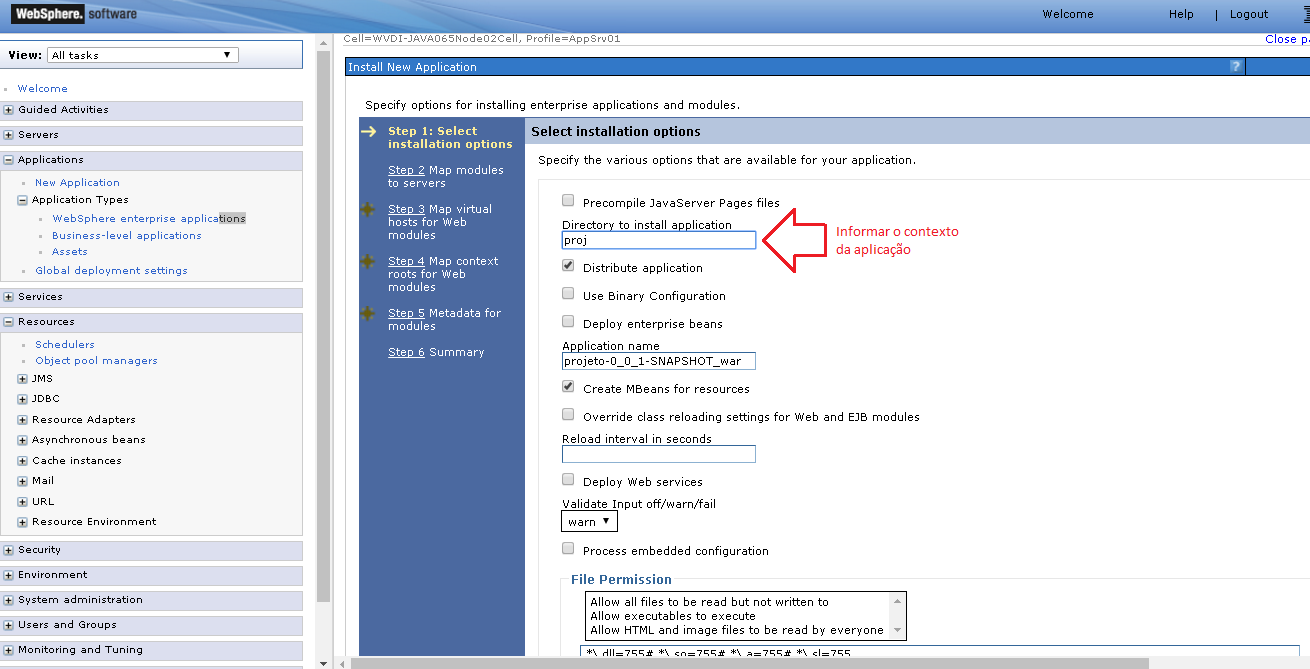
Informar o arquivo .WAR gerado, Next



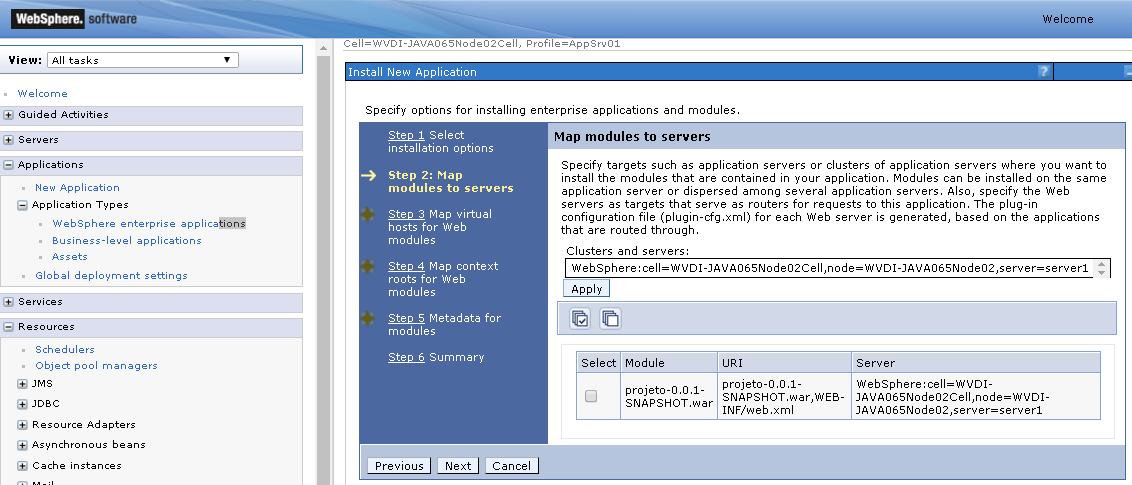
Next



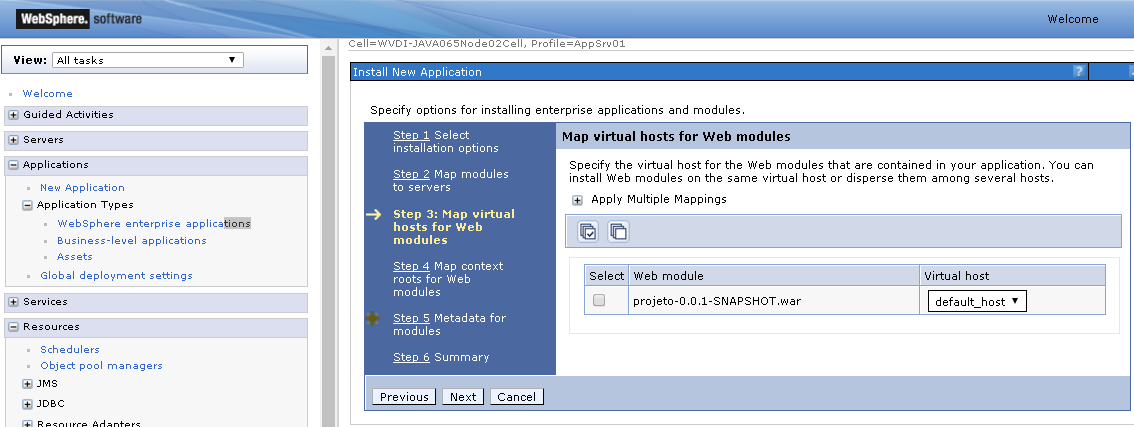
Informar o contexto da aplicação, Next



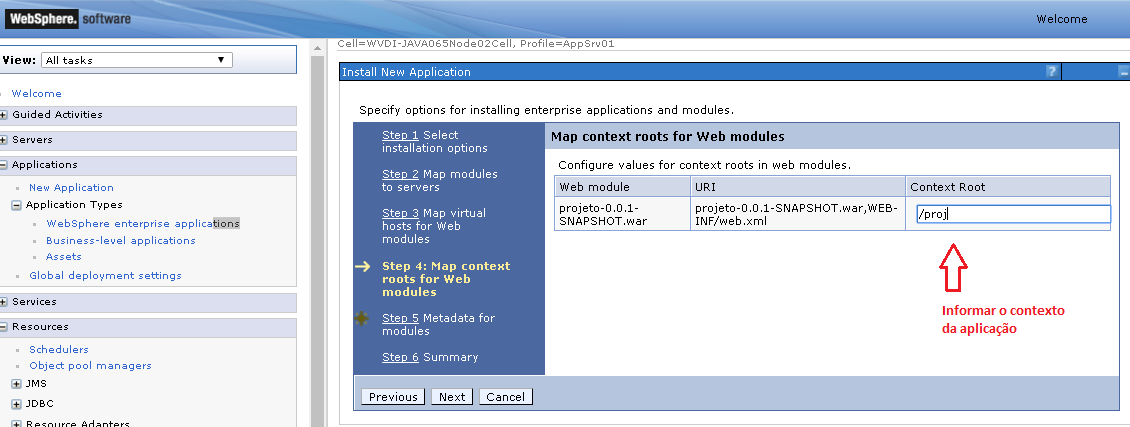
Next



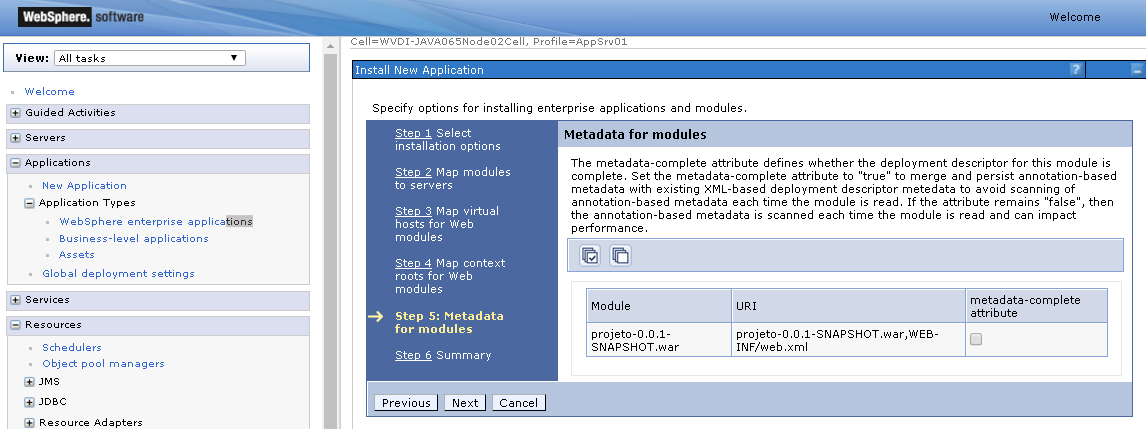
Next



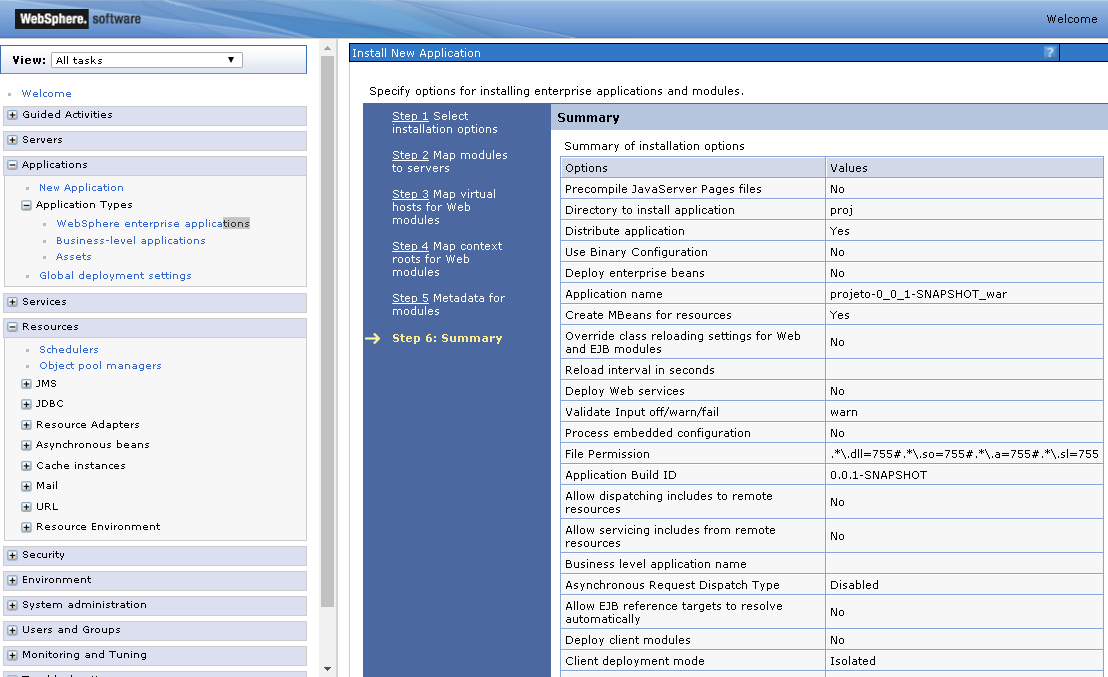
Repetir o contexto da aplicação, Next



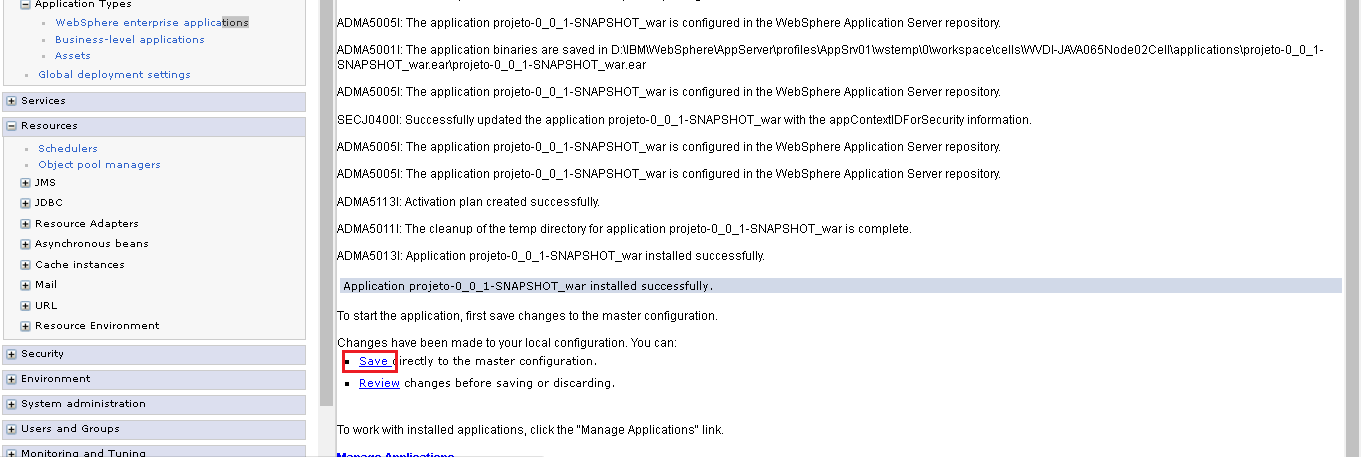
Next



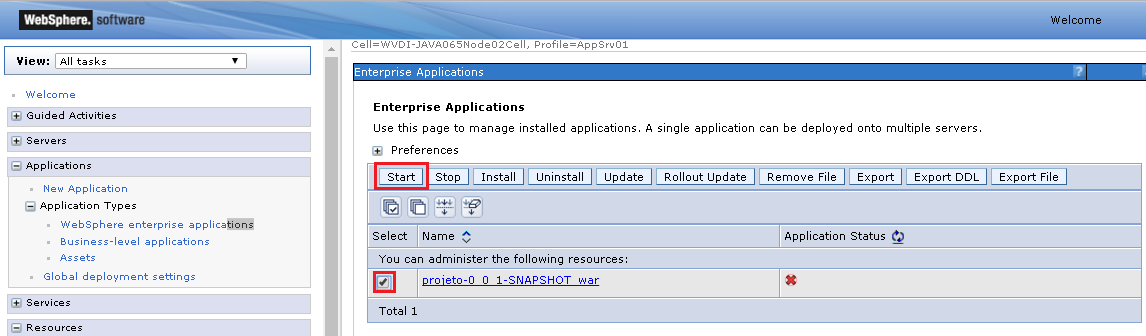
Finish



Salvar as configurações:

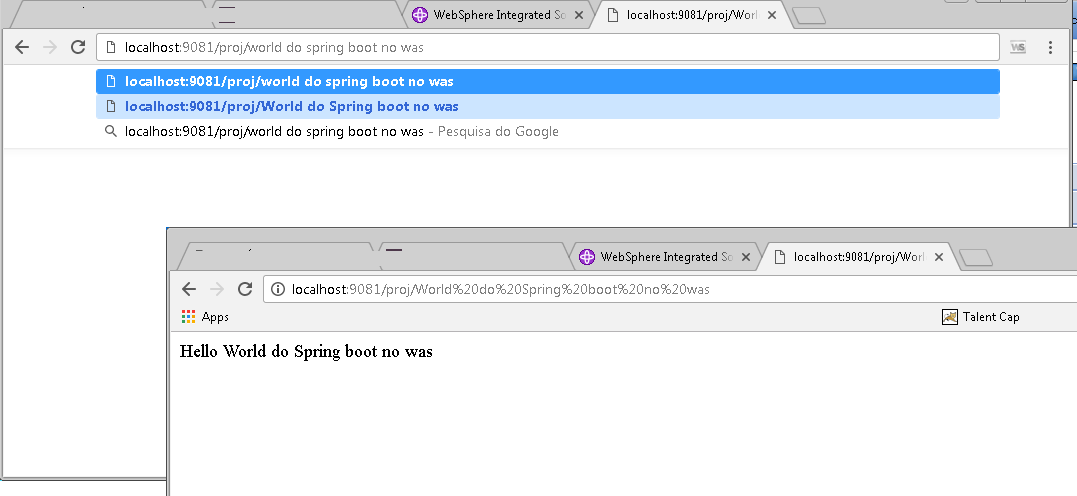


Iniciar a aplicação:



Obs.: No console e log do was, será exibido alguns erros referente a JPA que ainda não está configurado.

Após a aplicação ter iniciado com êxito (caso a aplicação não inicie consultar os logs), testar o serviço rest criado:



Sintaxe da URL criada:

http:// servidor : porta / contexto / parâmetros do serviço rest

## Configurando o JPA

Para usar o banco de dados H2 (banco de dados na memória), usar no arquivo de configuração (/projeto/src/main/resources/application.properties) os dados abaixo:

#####

## A Configuração abaixo habilita o banco de dados em memória

#####

# H2

spring.h2.console.enabled=true

spring.h2.console.path=/h2

# Datasource

spring.datasource.url=jdbc:h2:file:~/test

spring.datasource.username=sa

spring.datasource.password=

spring.datasource.driver-class-name=org.h2.Driver

spring.jpa.show-sql=true

Para usar uma configuração de JDNI, usar no arquivo de configuração os dados abaixo:

#####

## Configuração para usar o Datasource via JNDI

####

spring.datasource.jndi-name=java:jdbc/nome-do-datasource

spring.datasource.driver-class-name=com.microsoft.sqlserver.jdbc.SQLServerDriver

spring.jpa.show-sql=true

spring.jpa.hibernate.dialect=org.hibernate.dialect.SQLServer2012Dialect

Código Fonte do projeto:

<https://github.com/fabriciomaciel/springBootWas>

GNU General Public License v3.0