API RESTfull com SpringBoot

Vamos aprender como desenvolver uma aplicação Java, utilizando o ecossistema Spring. Será uma aplicação bem simples, aonde receberá dois dígitos números, fará a soma e devolverá um documento JSON com os valores recebidos e o resultado da soma.

Um pouco de teoria...

Uma aplicação web é uma forma inteligente de disponibilizar algo para um grande número de pessoas (podendo ser o mundo todo) através de um único local. Porém uma aplicação web para atender uma grande quantidade de consumidores, seja eles um device, uma página web, uma aplicação mobile ou qualquer outro meio, precisamos de uma linguagem em comum entre essas aplicações. Nesse post iremos aprender como criar uma aplicação API Restfull simples.

O que é API?

API é um acrônimo para Application Programming Interface (Interface de Programação de Aplicações). Quando disponibilizamos uma API, estamos ofertando funcionalidades para que aplicações externas possam utilizá-las sem a necessidade de conhecer a sua implementação.

E o que é RESTfull?

RESTfull, essa é fácil, RESTFull é uma aplicação REST. Mas o que é REST? REST é um acrônimo para Representational State Transfer (Transferência de Estado Representacional). REST não é um framework, REST é uma especificação, ou seja, um conjunto de regras/princípios que auxilia o desenvolvedor a criar uma API com interfaces bem definidas. Sim, isso é uma explicação bem simplista do que é REST, vale muito a pena conhece-lo um pouco mais, procure a documentação oficial (RFCs), esses sites também podem lhe auxiliar um pouco mais: <https://becode.com.br/o-que-e-api-rest-e-restful/>, <https://pt.wikipedia.org/wiki/REST> e <http://blog.caelum.com.br/rest-principios-e-boas-praticas/>.

E o SpringBoot?

Essa é um projeto da Spring que nos auxilia na configuração do nosso projeto. O SpringBoot trabalha com starters que você pode adicionar no seu projeto. Starter são basicamente dependências que agrupam outras dependências. Por exemplo, caso você deseja trabalhar com banco de dados, basta adicionar o starter spring-data, que o SpringBoot fará a importação de todas as dependências necessárias para trabalhar com banco de dados. A função do SpringBoot é auxiliar o desenvolvedor para que faça o mínimo possível de configuração e dedique-se mais ao negócio. Mas o SpringBoot não impende que você tenha configurações customizadas. Aqui <https://www.devmedia.com.br/spring-boot-simplificando-o-spring/31979> você consegue um pouco mais de informação sobre o SpringBoot

Por último, JSON...

JSON é um acrônimo para JavaScript Object Notation (Notação de Objetos JavaScript). Esse formatação de dados ganhou um grande parcela das APIs, pois ele é simples e leve e, o melhor, nada verboso. Aqui representamos os nossos dados na estrutura chave/valor. Um documento JSON inicia com colchete de abertura “{“ e termina com colchete de fechamento “}”. Dentro dos colchetes possui uma estrutura de chave/valor desordenado, separando valor e valor por dois pontos “:”. Os dados são separados por vírgula e suporta dados aninhados, um documento JSON dentro de outro documento JSON. Esse site <https://www.json.org/json-pt.html> tem uma explicação simples e legal do JSON

A nossa teoria é uma visão bem simples, com o único intuito de que você tenha uma visão macro do que iremos desenvolver. Não se limite a apenas esse conteúdo, o próprio site da Spring traz vários exemplos e tutoriais de como utilizar seus projetos, abuse do conhecimento, você só aprenderá se tentar, e claro, com certeza você errará, mas persista!

Vamos para a nossa aplicação.

Pré-requisitos:

Ambiente instalado (ConfigEnvironment.docx)

Iniciando o projeto

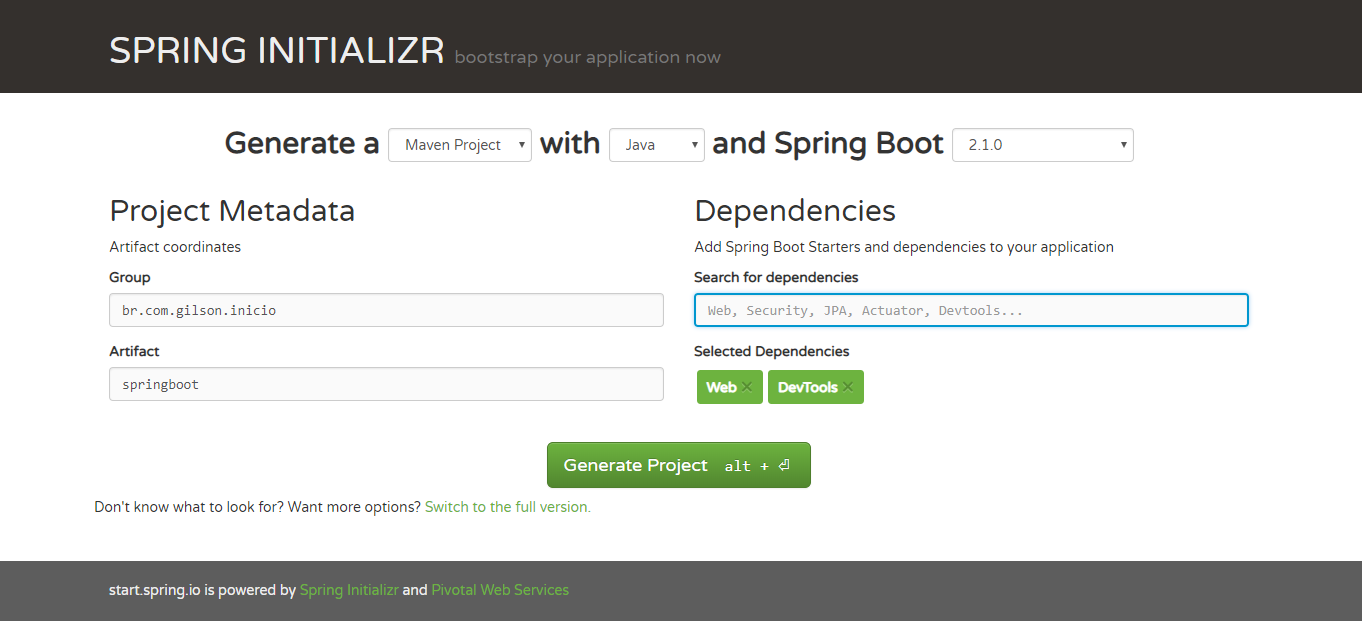
Você tem duas maneiras (ok, mais de duas, mas iremos abordar apenas duas...) de iniciar o seu projeto Java com SpringBoot, uma é utilizando um serviço web do projeto Spring, o starter.spring.io e o outro é diretamente na IDE STS.

Projeto Start Spring

A primeira que iremos explicar será através do site que o projeto Spring desenvolveu. Acesse o site start.spring.io, em Generate selecione Maven Project, em with selecione Java e em Spring Boot, deixe a versão 2.1.0 (a versão estável no dia do desenvolvimento desse tutorial). Em Group insira o pacote inicial do seu projeto, a convenção orienta que o pacote possua o domínio, ponto “.”, empresa, ponto novamente “.” e projeto; e no Artifact, informe o nome do seu serviço, uma identificação desta API. Isso é uma orientação, se você inserir qualquer outro valor funcionará perfeitamente, mas para manter o padrão de mercado é recomendável iniciar conforme a convenção. Tente ao máximo seguir a convenção, pois assim você desenvolverá um software simples e fácil de manutenção.

Em Dependencies você adicionará os starters que você necessitará no seu projeto. Se você precisa incrementar ou remover algum starter mais a frente, não tem problemas, desde que você não esteja utilizando esse starter em seu projeto. Para adicionar a dependência você precisa digitá-la e quando localizá-la, clique em cima dela que automaticamente será incrementada ao seu projeto.

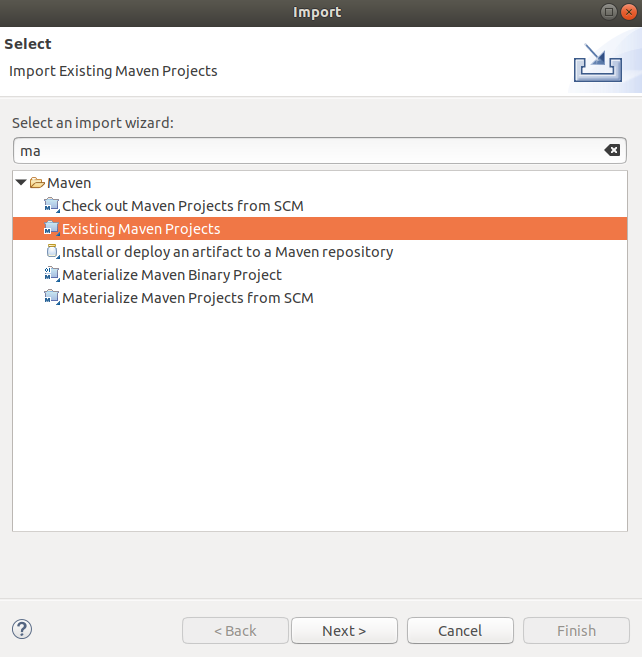
Verifique o nosso exemplo, estamos criando um projeto Maven em Java utilizando a versão 2.1.0 do SpringBoot, o nosso group é br.com.gilson.inicio e o Artifact é springboot, estamos utilizando apenas os starters Web e DevTools.



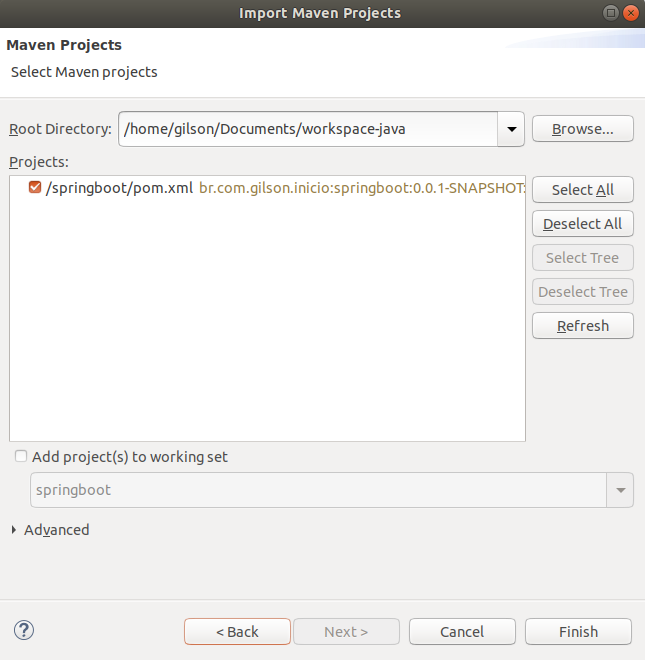
Se você clicar no link “switch to the full version”, você terá uma versão completa (como o link diz) do starter. Nessa visão você terão uma visão completa dos starters disponíveis e poderá alterar a versão do Java, o tipo do pacote (JAR ou WAR) e o nome completo do pacote. Ambas as visões o resultado final será o mesmo.

Agora clique no botão Generate Project, será aberto uma janela para salvar um arquivo do tipo ZIP, esse é o nosso projeto inicial. Salve esse arquivo em disco e descompacte-o.

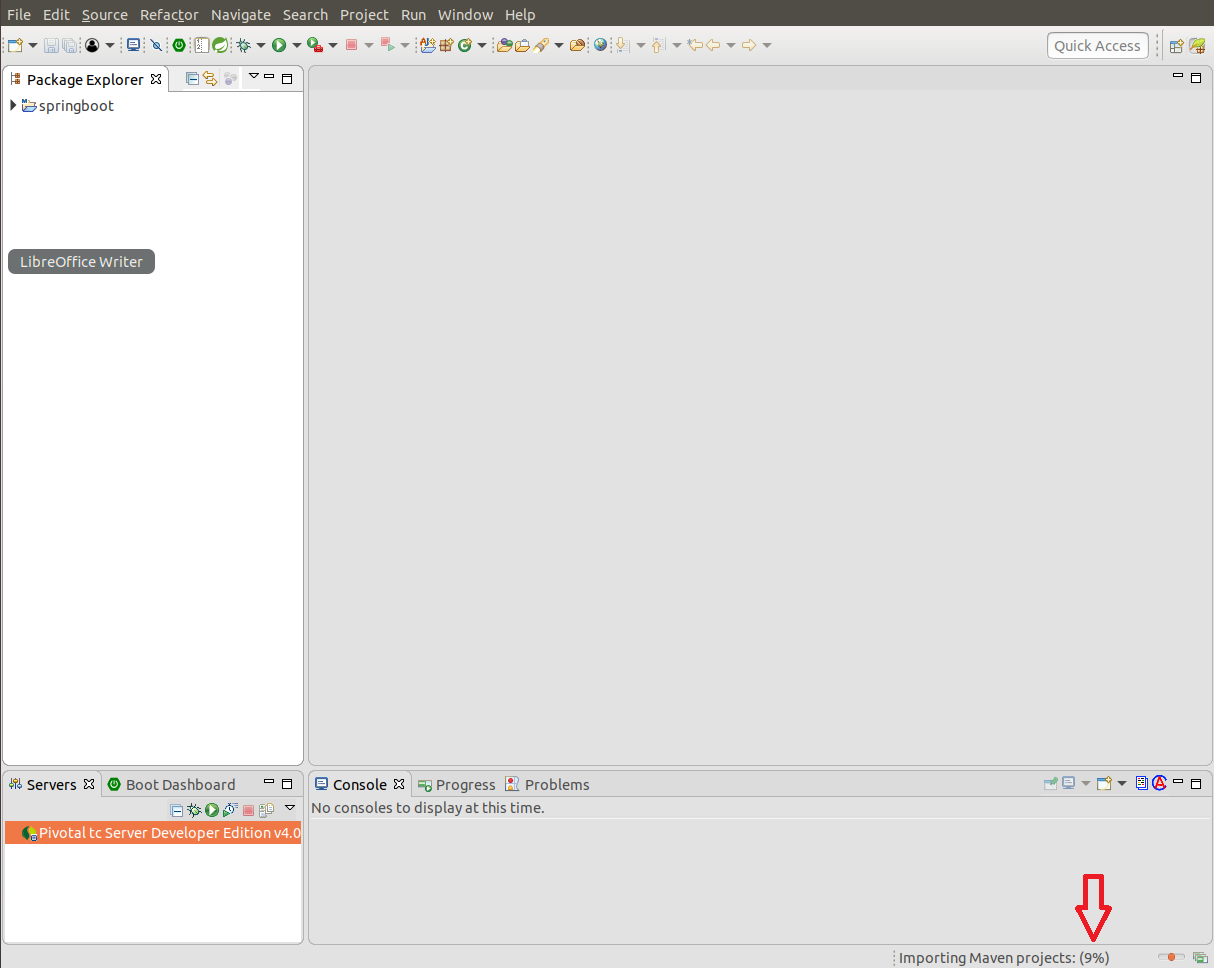
Agora acesse o seu IDE STS, clique no menu File/Import e em “Selects as import wizard” digite Maven, selecione a opção “Existing Maven Projects” e clique em Next.



Clique no botão Browser e localize o diretório do projeto descompactado, o projeto deverá ser exibido no dashboard Projects. Selecione o seu projeto e clique em Finish



Agora o Maven fará o download de todas as dependências fornecidas pela SpringBoot, note na barra de status do seu IDE aonde informa que esta realizando o download dos recursos, aguarde o término do download, o tempo dependerá da sua conexão com a internet:

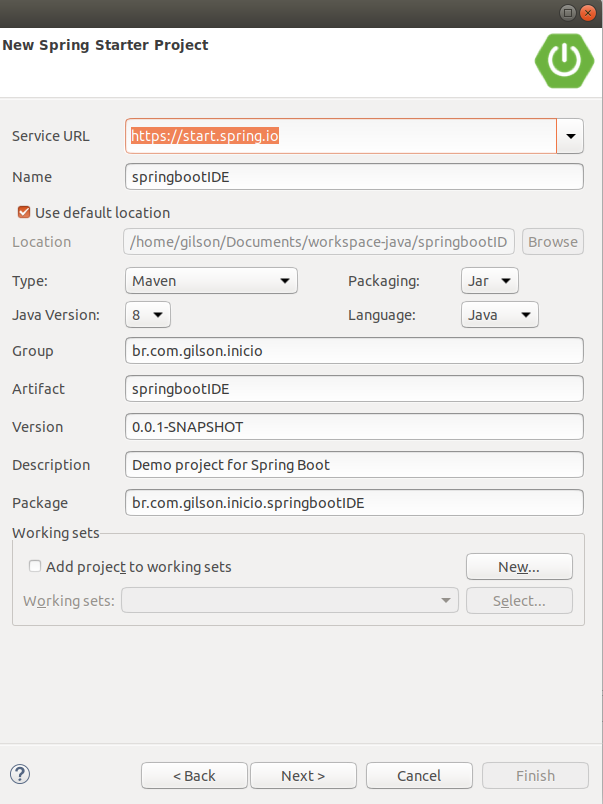


Parabéns, você aprender a primeira forma de iniciar um projeto Java utilizando o SpringBoot.

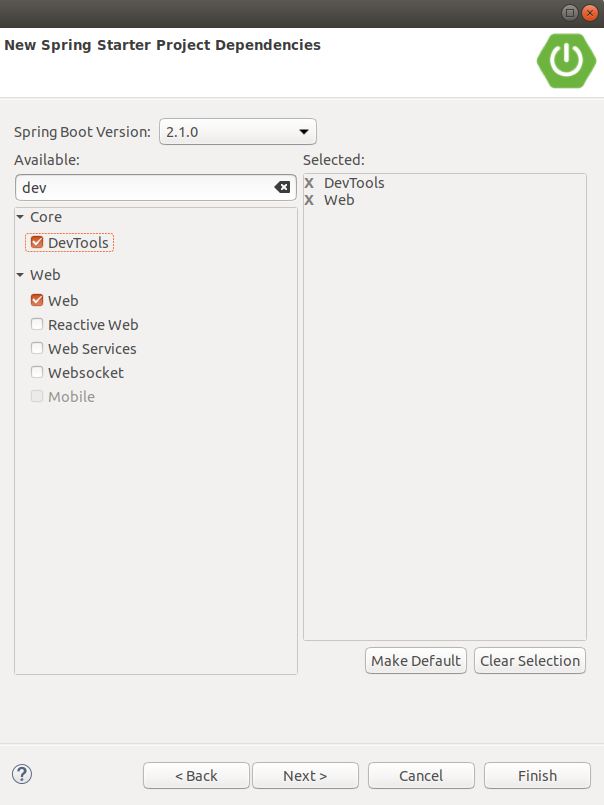
Agora vamos conhecer a segunda forma, muito mais simples.

No seu IDE STS, clique em File/New/Spring Starter Project, ou pressione <CTRL>+<N> e em Wizards digite “Spring Starter”, dai você seleciona a opção Spring Starter Project e clique em Next.

Essa tela possui as mesmas configurações da visão FULL (e mais alguns campos) da primeira forma, a diferença é que nessa tela você não selecionará os starters do Spring. O campo Nome é o mesmo valor do campo Artifact, perceba que quando você preenche esse campo, o campo Artifact recebe o mesmo valor; o campo type é similar o campo “Generate a”; o campo Packet não é preenchido automaticamente, você pode inserir o conteúdo do campo Group mais o conteúdo do campo Artifact. Preencha os campos utilizando os mesmos critérios da primeira forma. depois clique em Next:



Agora você seleciona os startes do Spring, como na primeira forma iremos selecionar Web e DevTools. Para isso digite o nome em Available ou localize navegando pelas arvores, depois de selecionar o starter, selecione-os. Agora clique em Finish:



Pronto, você já iniciou o seu projeto Java utilizando o SpringBoot. Perceba na barra de status do seu IDE que o download não foi realizado, isso ocorre pois o Maven cria um repositório em sua máquina e faz o download, depois disso, em todas as situações que você precisar daquele pacote específico, ele não irá buscar na internet, mas sim na sua máquina, ou seja, o Maven, quando detecta que precisa de um determinado pacote, primeiro ele busca no repositório local, e importa para o seu projeto, caso não localize no repositório local, então faz o download para a sua máquina. Vale a pena aprender Maven, utilizamos muito no dia a dia.

Agora que o projeto esta criado, iremos criar a nossa API.

Criando nossa API