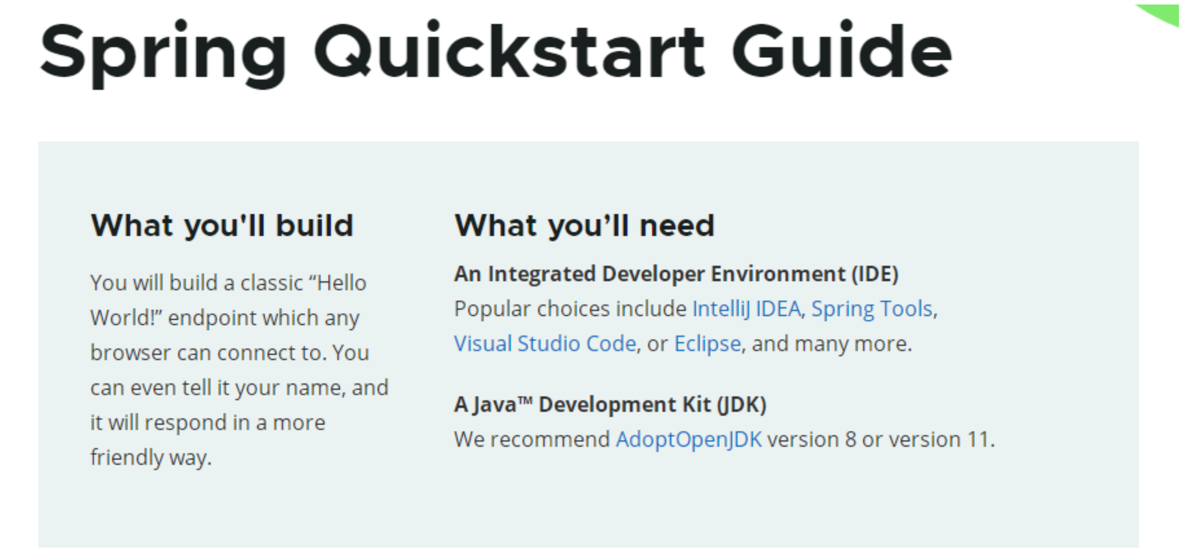
spring.io/quickstart

本期课程讲解Spring官网的`快速上手`页面

官网地址 https://spring.io/quickstart

Spring Quickstart Guide



Spring Quickstart Guide

What you'll build

You will build a classic "Hello World!" endpoint which any browser can connect to. You can even tell it your name, and it will respond in a more friendly way.

What you'll need

An Integrated Developer Environment (IDE)

Popular choices include IntelliJ IDEA, Spring Tools, Visual Studio Code, or Eclipse, and many more.

A Java™ Development Kit (JDK)

We recommend AdoptOpenJDK version 8 or version 11.

Spring 快速开始指南

Guide

指南

What you'll build

接下来你将要构建的是什么?

build

构建

You will build a classic "Hello World!" endpoint which any browser can connect to. You can even tell it your name, and it will respond in a more friendly way.

您将构建一个经典的"Hello World!"任何浏览器都可以连接的端点。你甚至可以告诉它你的名字，它会以一种更友好的方式回应你。

You will build a classic "Hello World!" endpoint which any browser can connect to

你将要构建的是一个经典的 helloworld端点，任何浏览器都可以连接上。

classic 经典的

endpoint 端点，终端

browser 浏览器

connect 连接

You can even tell it your name, and it will respond in a more friendly way.

你甚至可以告诉它你的名字，它会以一种更友好的方式回应你。

respond 应答

What you'll need

你需要什么

An Integrated Developer Environment (IDE)

一个集成开发环境

Integrated 集成

Developer 开发

Environment 环境

Popular choices include IntelliJ IDEA, Spring Tools, Visual Studio Code, or Eclipse, and many more.

比较流行的选择包括 IntelliJ IDEA, Spring Tools, Visual Studio Code, 或者Eclipse，等等。

include 包含，包括

A Java™ Development Kit (JDK)

对jdk的要求

We recommend AdoptOpenJDK version 8 or version 11.

我们推荐使用 **AdoptOpenJDK** 8 或者 11版。

recommend 建议，推荐

Step 1: Start a new Spring Boot project

第一步，创建一个新的springboot项目

Step 步骤

project 项目

Use start.spring.io to create a "web" project.

使用**start.spring.io**这个网站创建一个web项目

create 创建

In the "Dependencies" section search for and add the "web" dependency as shown in the screenshot.

在"依赖项"部分，搜索并添加"web"依赖，如屏幕截图所示。

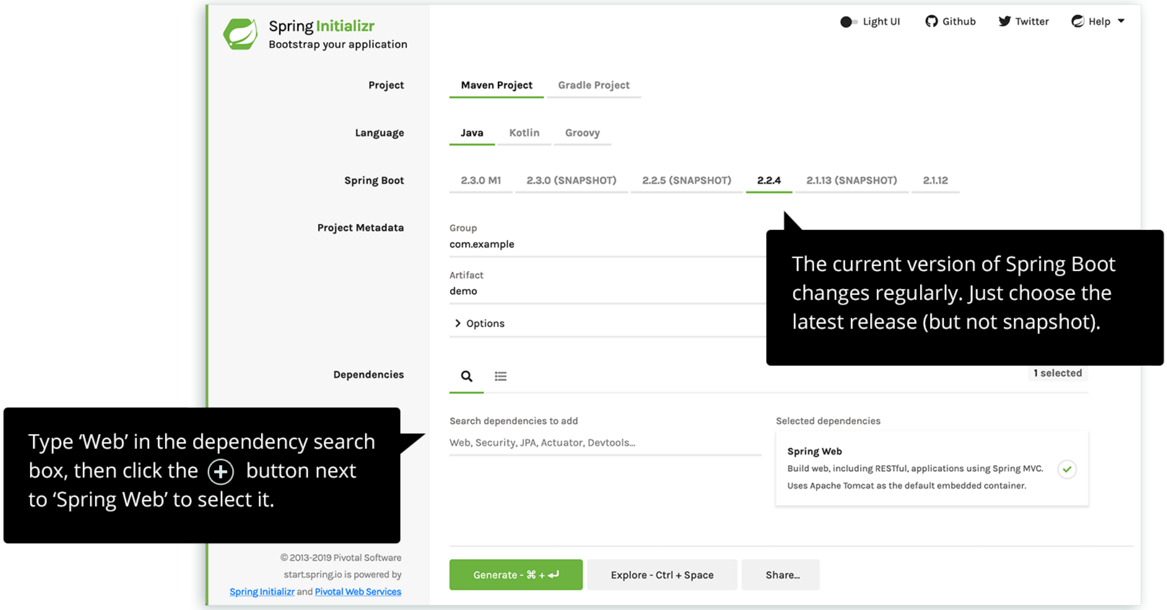
Dependencies 依赖

section 一部分

search搜索

shown in .... 如xx所示

screenshot 屏幕截图



Spring Initializr
Bootstrap your application
Light UI
Github
Twitter
Help
Generate - ⌘ + ↵
Explore - Ctrl + Space
Share...
© 2013-2019 Pivotal Software
start.spring.io is powered by
Spring Initializr and Pivotal Web Services

Project: Maven Project | Gradle Project

Language: Java | Kotlin | Groovy

Spring Boot: 2.3.0 M1 | 2.3.0 (SNAPSHOT) | 2.2.5 (SNAPSHOT) | 2.2.4 | 2.1.13 (SNAPSHOT) | 2.1.12

Project Metadata: Group com.example; Artifact demo; Options

Dependencies: 1 selected

Search dependencies to add: Web, Security, JPA, Actuator, Devtools...

Selected dependencies: Spring Web — Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

The current version of Spring Boot changes regularly. Just choose the latest release (but not snapshot).

Type 'Web' in the dependency search box, then click the ⊕ button next to 'Spring Web' to select it.

Hit the green "Generate" button, download the zip, and unpack it into a folder on your computer.

点击绿色的**生成**按钮，下载zip文件，并将其解压缩到你电脑上上的一个文件夹里。

Hit 点击

Generate生成

button 按钮

download 下载

zip 压缩格式

unpack 解开，解压缩

folder 文件夹

Step 2: Add your code

第二步，添加你的代码

Open up the project in your IDE and locate the `DemoApplication.java` file in the `src/main/java/com/example/demo` folder.

用ide打开刚下载的项目，并在`src/main/java/com/example/demo`文件夹中找到`DemoApplication.java`文件

locate 定位，位于

Now change the contents of the file by adding the extra method and annotations shown in the code below.

现在，修改文件内容，添加一些额外的方法和注解，如下代码所示

change 改变，修改

contents 内容

file 文件

extra 额外，扩展

method 方法

annotations 注解（不是注释）

```
              package com.example.demo;
              import org.springframework.boot.SpringApplication;
              import
org.springframework.boot.autoconfigure.SpringBootApplication;
              import org.springframework.web.bind.annotation.GetMapping;
              import org.springframework.web.bind.annotation.RequestParam;
              import org.springframework.web.bind.annotation.RestController;

              @SpringBootApplication
              @RestController
              public class DemoApplication {


                  public static void main(String[] args) {
                  SpringApplication.run(DemoApplication.class, args);
                  }

                  @GetMapping("/hello")
                  public String hello(@RequestParam(value = "name", defaultValue
= "World") String name) {
                  return String.format("Hello %s!", name);
                  }

              }
```

You can copy and paste the code or just type it.

你可以直接复制粘贴代码，或者自己敲。

The `hello()` method we've added is designed to take a String parameter called `name`, and then combine this parameter with the word `"Hello"` in the code.

`hello()`这个方法是我们添加的，我们把它设计成了一个可以接受String类型参数的方法，这个参数名是`name`,在方法体里，连接了`"Hello"`这个词。

designed 设计的

take a String parameter 获取到一个String类型的参数

combine 连接，联合

This means that if you set your name to `“Amy”` in the request, the response would be `“Hello Amy”`.

这么写的意思是，如果在请求发过来的时候`name`这个参数被设置成了`amy`,那么响应的结果就是`“Hello Amy”`

request 请求

response 响应

The `@RestController` annotation tells Spring that this code describes an endpoint that should be made available over the web.

`@RestController`这个注解告诉Spring，我们的这些代码想要开启一个可用的web服务端点

describes 描述

The `@GetMapping(“/hello”)` tells Spring to use our `hello()` method to answer requests that get sent to the `http://localhost:8080/hello` address.

`@GetMapping(“/hello”)` 这个注解写在了`hello()`这个方法上，告诉Spring我们想用这个方法应答请求，当请求地址为`http://localhost:8080/hello`时会执行这个方法

address 地址

Finally, the `@RequestParam` is telling Spring to expect a `name` value in the request, but if it’s not there, it will use the word “World” by default.

最后，`@RequestParam` 告诉Spring在处理请求时，期望接收到一个传递过来的`'name'`值，但是如果没有传值过来那么就使用`' World '`这个词作为默认值。

expect 期望

default 默认