**How to create a Windows Service for Spring Boot Application by Procrun**

Procrun is a set of applications that allow Windows users to wrap (mostly) Java applications (e.g. Tomcat) as a Windows service. The service can be set to automatically start when the machine boots and will continue to run with no user logged onto the machine.

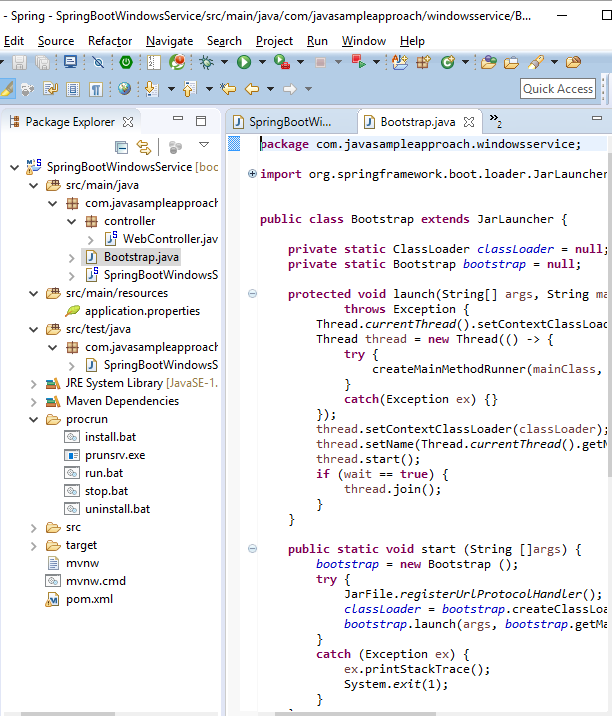
The tutorial will guide you how to create a Windows Service for Spring Boot Application by Procrun.

**I. Technologies – create a Windows Service for Spring Boot Application by Procrun**

– Java 1.8  
– Maven 3.3.9  
– Spring Tool Suite – Version 3.8.1.RELEASE  
– Spring Boot: 1.4.0.RELEASE  
– Procrun: commons-daemon-1.0.15

**II. Overview**

***1. Structure of Project***



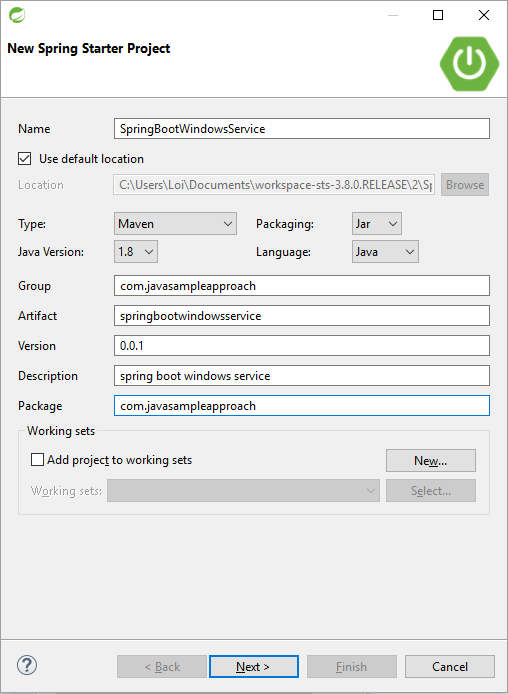
***2. Step to do***

– Create Spring Boot project  
– Add needed dependencies  
– Re-implement the Entry point of Spring Boot Application  
– Implement a Bootstrap class for Start & Stop windows-service function  
– Create a simple web controller  
– Configure build plugin & start-class  
– Download procrun  
– Build install, un-install, run, stop Procrun command-line for Spring Boot Windows Service  
– Run & Enjoy result

**III. Practices**

***1. Create Spring Boot project***

Open Spring Tool Suite, on main menu, choose **File->New->Spring Starter Project**, input project info as below images:



Press **Next** then **Finish**, spring boot project is created successfully.

***2. Add needed dependencies***

Open pom.xml file, add dependencies.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | <dependencies>      <dependency>          <groupId>org.springframework.boot</groupId>          <artifactId>spring-boot-starter-web</artifactId>      </dependency>        <dependency>          <groupId>org.springframework.boot</groupId>          <artifactId>spring-boot-starter-test</artifactId>          <scope>test</scope>      </dependency>    <strong>          <dependency>          <groupId>org.springframework.boot</groupId>          <artifactId>spring-boot-loader</artifactId>          <scope>provided</scope>      </dependency>  </strong>  </dependencies> |

***3. Re-implement the Entry point of Spring Boot Application***

For **start/stop** function of a Windows-Service, re-implement entry point class:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | package com.javasampleapproach.windowsservice;      import java.lang.management.ManagementFactory;    import org.slf4j.Logger;  import org.slf4j.LoggerFactory;  import org.springframework.boot.ExitCodeGenerator;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  import org.springframework.context.ApplicationContext;    @SpringBootApplication  public class SpringBootWindowsServiceApplication {      private static final Logger logger = LoggerFactory.getLogger(SpringBootWindowsServiceApplication.class);      private static ApplicationContext applicationContext = null;        public static void main(String[] args) {          String mode = args != null && args.length > 0 ? args[0] : null;            if (logger.isDebugEnabled()) {              logger.debug("PID:" + ManagementFactory.getRuntimeMXBean().getName() + " Application mode:" + mode + " context:" + applicationContext);          }          if (applicationContext != null && mode != null && "stop".equals(mode)) {              System.exit(SpringApplication.exit(applicationContext, new ExitCodeGenerator() {                  @Override                  public int getExitCode() {                      return 0;                  }              }));          }          else {              SpringApplication app = new SpringApplication(SpringBootWindowsServiceApplication.class);              applicationContext = app.run(args);              if (logger.isDebugEnabled()) {                  logger.debug("PID:" + ManagementFactory.getRuntimeMXBean().getName() + " Application started context:" + applicationContext);              }          }      }  } |

***4. Implement a Bootstrap class for Start & Stop windows-service function***

Create a **Bootstrap** class that extends **JarLauncher**. The class will implement 2 function **start & stop** for Windows-Service.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65 | package com.javasampleapproach.windowsservice;    import org.springframework.boot.loader.JarLauncher;  import org.springframework.boot.loader.jar.JarFile;      public class Bootstrap extends JarLauncher {        private static ClassLoader classLoader = null;      private static Bootstrap bootstrap = null;        protected void launch(String[] args, String mainClass, ClassLoader classLoader, boolean wait)              throws Exception {          Thread.currentThread().setContextClassLoader(classLoader);          Thread thread = new Thread(() -> {              try {                  createMainMethodRunner(mainClass, args, classLoader).run();              }              catch(Exception ex) {}          });          thread.setContextClassLoader(classLoader);          thread.setName(Thread.currentThread().getName());          thread.start();          if (wait == true) {              thread.join();          }      }        public static void start (String []args) {          bootstrap = new Bootstrap ();          try {              JarFile.registerUrlProtocolHandler();              classLoader = bootstrap.createClassLoader(bootstrap.getClassPathArchives());              bootstrap.launch(args, bootstrap.getMainClass(), classLoader, true);          }          catch (Exception ex) {              ex.printStackTrace();              System.exit(1);          }      }        public static void stop (String []args) {          try {              if (bootstrap != null) {                  bootstrap.launch(args, bootstrap.getMainClass(), classLoader, true);                  bootstrap = null;                  classLoader = null;              }          }          catch (Exception ex) {              ex.printStackTrace();              System.exit(1);          }      }        public static void main(String[] args) {          String mode = args != null && args.length > 0 ? args[0] : null;          if ("start".equals(mode)) {              Bootstrap.start(args);          }          else if ("stop".equals(mode)) {              Bootstrap.stop(args);          }      }  } |

***5. Create a simple web controller***

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | package com.javasampleapproach.windowsservice.controller;    import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RestController;    @RestController  public class WebController {          @RequestMapping("/hello")      public String hello(){          return "hello";      }    } |

***6. Configure build plugin & start-class***

Open **pom.xml** file, configure start-class:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <properties>          ...      <!-- Main Class -->      <start-class>com.javasampleapproach.windowsservice.SpringBootWindowsServiceApplication</start-class>          ...  </properties> |

Configure Maven build plugin:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | <build>      <plugins>          <plugin>              <groupId>org.springframework.boot</groupId>              <artifactId>spring-boot-maven-plugin</artifactId>              <configuration>                  <excludes>                      <exclude>                          <groupId>org.springframework.boot</groupId>                          <artifactId>spring-boot-loader</artifactId>                      </exclude>                  </excludes>              </configuration>          </plugin>          <plugin>              <groupId>org.apache.maven.plugins</groupId>              <artifactId>maven-antrun-plugin</artifactId>              <version>1.8</version>              <executions>                  <execution>                      <phase>package</phase>                      <configuration>                          <target>                              <zip destfile="${project.build.directory}/${project.build.finalName}.jar"                                  update="true" compress="store">                                  <fileset dir="${project.build.directory}/classes"                                      <strong>includes="com/javasampleapproach/windowsservice/Bootstrap.class"</strong> />                              </zip>                          </target>                      </configuration>                      <goals>                          <goal>run</goal>                      </goals>                  </execution>              </executions>          </plugin>      </plugins>  </build> |

***7. Download procrun***

Download Procrun from: [Commons-Deamon](https://commons.apache.org/proper/commons-daemon/download_daemon.cgi)

***8. Build install, un-install, run, stop Procrun command-line for Spring Boot Windows Service***

– Create **install.bat** file

|  |  |
| --- | --- |
| 1  2  3  4  5 | prunsrv.exe //IS//Procrun-SpringBootWindowService --DisplayName="Procrun-SpringBootWindowService" --Description="Procrun-SpringBootWindowService" ^                              --Startup=auto --Install=%CD%\prunsrv.exe --Jvm=auto --Classpath=%CD%\..\target\springbootwindowsservice-0.0.1.jar ^                              --StartMode=jvm --StartClass=com.javasampleapproach.windowsservice.Bootstrap --StartMethod=start --StartParams=start ^                              --StopMode=jvm --StopClass=com.javasampleapproach.windowsservice.Bootstrap --StopMethod=stop --StopParams=stop ^                              --StdOutput=auto --StdError=auto --LogPath=%CD% --LogLevel=Debug ^ |

– Create**uninstall.bat** file

|  |  |
| --- | --- |
| 1 | prunsrv.exe //DS//Procrun-SpringBootWindowService |

– Create **run.bat** file

|  |  |
| --- | --- |
| 1 | prunsrv.exe //ES//Procrun-SpringBootWindowService |

– Create **stop.bat** file

|  |  |
| --- | --- |
| 1 | prunsrv.exe //SS//Procrun-SpringBootWindowService |

– Use **echo %PROCESSOR\_ARCHITECTURE%** for determined the appropriate procrun.exe

– Place all files: install.bat, uninstall.bat, run.bat, stop.bat and prunsrv.exe in a **procrun**folder.

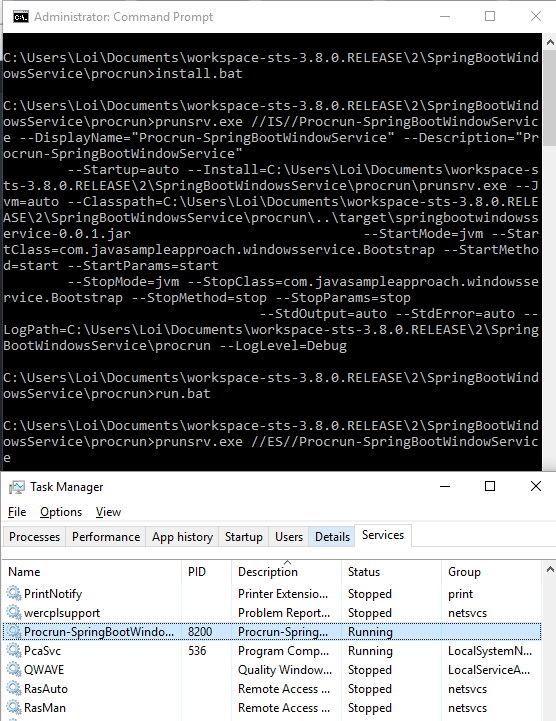
***9. Run & Enjoy result***

-Maven build project with goals:

|  |  |
| --- | --- |
| 1 | clean install |

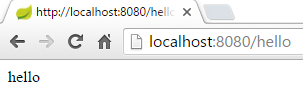
– Open **cmd** on windows with mode:**administrator privileges**, cd to **procrun** folder.

– Run **install.bat** – for register a Window Service  
– Run **run.bat** – for start the Windows-Service



Make request:

|  |  |
| --- | --- |
| 1 | <http://localhost:8080/hello> |



– Run **stop.bat**: for stop Windows-service  
– Run **uninstall.bat**: for un-register Windows-Service