

The Banner in Springboot

- Create one file on class path named with **banner.txt** and the same file will be set as banner automatically
- banner.txt may contain any type of string
- `spring.main.banner-mode=off`

We can turn off the banner with the `spring.main.banner-mode` property.

- `SPRING_MAIN_BANNER-MODE=off`

It is also possible to disable the banner with the `SPRING_MAIN_BANNER-MODE` environment variable.

```
new SpringApplicationBuilder(MyApplication.class)
```

```
.bannerMode(Banner.Mode.OFF)
```

```
.run(args)
```

```
var app = new SpringApplication(Application.class);
```

```
app.setBannerMode(Banner.Mode.OFF);
```

```
app.run(args);
```

Spring Boot custom banner

- The banner can be specified in text file or in an image file. The default name for a banner is banner.txt or banner.(png|jpg|gif). It image is automatically transformed into ASCII. The default location of banner files is src/main/resources.

`spring.banner.location=classpath:custom-banner.txt`

`spring.banner.image.location=classpath:custom-banner.png`

- The default location of a text file banner can be configured with the **spring.banner.location** property and the default location of the image banner is configured with the **spring.banner.image.location**.
- There are properties for customizing image banners, including

`spring.banner.image.width`

`spring.banner.image.height`

`spring.banner.image.margin`

`spring.main.log-startup-info=false`

`spring.output.ansi.enabled=always`

Spring Boot custom banner

- <https://devops.datenkollektiv.de/banner.txt/index.html>
- <https://springhow.com/spring-boot-banner-generator/>
- `${AnsiColor.YELLOW}`
- `${AnsiColor.DEFAULT}`
- `spring.banner.location=banner-production.txt`
- `spring.banner.image.location=banner-image.png`
- `spring.main.banner-mode=off`
- <https://www.javacodemonk.com/custom-banner-in-spring-boot-93a85044>
- <https://devops.datenkollektiv.de/banner.txt/index.html>
- <http://www.network-science.de/ascii/>
<https://textkool.com/en/ascii-art-generator?hl=full&vl=full&font=Red%20Phoenix&text=stacktraceguru>
- <http://patorjk.com/software/taag/#p=display&f=Doom&t=stacktraceguru>
- <https://manytools.org/hacker-tools/ascii-banner/>
- <https://www.kammerl.de/ascii/AsciiSignature.php>

banner.txt file Placeholder

<u>Variable</u>	<u>Description</u>
<code>\${application.version}</code>	The version number of your application, as declared in MANIFEST.MF
<code>\${application.formatted-version}</code>	The version number of your application, as declared in MANIFEST.MF and formatted for display
<code>\${spring-boot.version}</code>	The Spring Boot version
<code>\${spring-boot.formatted-version}</code>	The Spring Boot version formatted for display
<code>\${application.title}</code>	The title of the application, as declared in MANIFEST.MF

The text can be coloured with variables such as `${AnsiColor.NAME}` or `${AnsiBackground.NAME}`.

Banner.txt

The diagram illustrates the derivation of the word 'x' from the letters 'a' and 'b' using a series of 11 rules. The rules are numbered 1 through 11. The transformations involve various geometric shapes and symbols, including lines, curves, and the letter 'x'.

Rule 1: $a \rightarrow b$

Rule 2: $a \rightarrow b$

Rule 3: $a \rightarrow b$

Rule 4: $a \rightarrow b$

Rule 5: $a \rightarrow b$

Rule 6: $a \rightarrow b$

Rule 7: $a \rightarrow b$

Rule 8: $a \rightarrow b$

Rule 9: $a \rightarrow b$

Rule 10: $a \rightarrow b$

Rule 11: $a \rightarrow b$

The final result is the word 'x'.

Application Name: `${application.title}`

Application Version: `${application.version}`

```
${AnsiColor.RED} :: Spring Boot
```

```
${spring-boot.formatted-version} :: ${AnsiColor.RED}
```

```
${AnsiColor.DEFAULT}
```

MyFirstRestController.java

```
package org.tarkesh.colpackage;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
```

```
public class MyFirstRestController {
```

```
    @RequestMapping(value = "/")
```

```
    public String greeting() {
```

```
        return "Welcome to Spring REST web service";
```

```
    }
```

```
}
```