Allgemeine Infos

Projekt Konfigurator:
http://start.spring.io/
Allgemeine Blogeinträge:
http://blog.codeleak.pl/2014/04/how-to-spring-boot-and-thymeleaf-with-maven.html

Properties und Filterung

* Erbt das Projekt von spring-boot-parent so ist im Mavenbuild bereits die Filterung überschrieben:
** In der application.properties werden Ausdrücke @xxx@ durch maven Werte ersetzt.
* Thymleaf ersetzt Ausdrücke der Art ${xxx} und #{xxx} zur Laufzeit mit Werten

applications.properties

spring.thymeleaf.cache: false	Schaltet den Cache für Thymleaf aus. Damit wird bei jedem Request das Template neu übersetzt und bei der Entwicklung braucht der Server für HTML Anpassungen nicht neu gestartet werden.
endpoints.shutdown.enabled	Aktiviert den Endpoint zum stoppen des Servers per POST Request auf den Management Port

Das Herunterfahren des Servers lässt sich bei der Konfiguration des Managementports auf **8080** und eingeschaltetem Endpoint wie folgt von der Console aus auslösen: **curl -X POST http://localhost:8080/shutdown**

Bootstrap - Definition Liste horizontal

```
<dl class="dl-horizontal">
<dt>
<span class= " gly phicon glyphi con -tr ash " >
</ spa n>
</d t>
<dd >Buch lösche n</ dd>
<dt>
<span class= " gly phicon glyphi con -wa rni ng- -
sig n">< /sp an>
</d t>
<dd >ISBN ungült ig< /dd>
</d l>
```

Interceptors

Interceptoren in Spring Boot können genutzt werden um:
* Requests an den Controller zu modifizieren (Header etc.)
* Responses an den Client zu modifizieren (Header etc.)
Ein Interceptor wird erstellt durch eine Klasse mit:
* @Component
* implements HandlerInterceptor
Folgende Lifecycle Hooks sind verfügbar:
* preHandle: Modifikation des Request vor Erreichen des Controllers. M
zurückliefern, damit ein Response erzeugt wird.
* postHandle: Modifikation des Response vor Herausgabe an den Clien
* afterCompletion: Modifikation wenn Request und Response fertig bea
Jeder Interceptor muss bei der InterceptorRegistry angemeldet werden

```
@Component
public class Produc tSe rvi ceI nte rce pto rAp pCon
s
WebMvc Con fig ure rAd apter {
   @Autowired
   Produc tSe rvi ceI nte rceptor produc tSe rvi ceI
tor;
   @Override
   public void addInt erc ept ors (In ter cep tor Re
gistry) {
      regist ry.a dd Int erc ept or( pro duc tSe rvi
ce ptor);
   }
}
```

Quelle: https://www.tutorialspoint.com/spring_boot/spring_boot_interceptor

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com