



# Spring Boot Proxies

Die unsichtbare Magie hinter den  
Annotationen

Wie Spring deine Klassen "einpackt", um dir Arbeit abzunehmen. Eine  
Einführung für absolute Anfänger.

# Die Analogie: Der Butler

---

Stell dir vor, du bist ein vielbeschäftiger Chef  
(deine Java-Klasse).

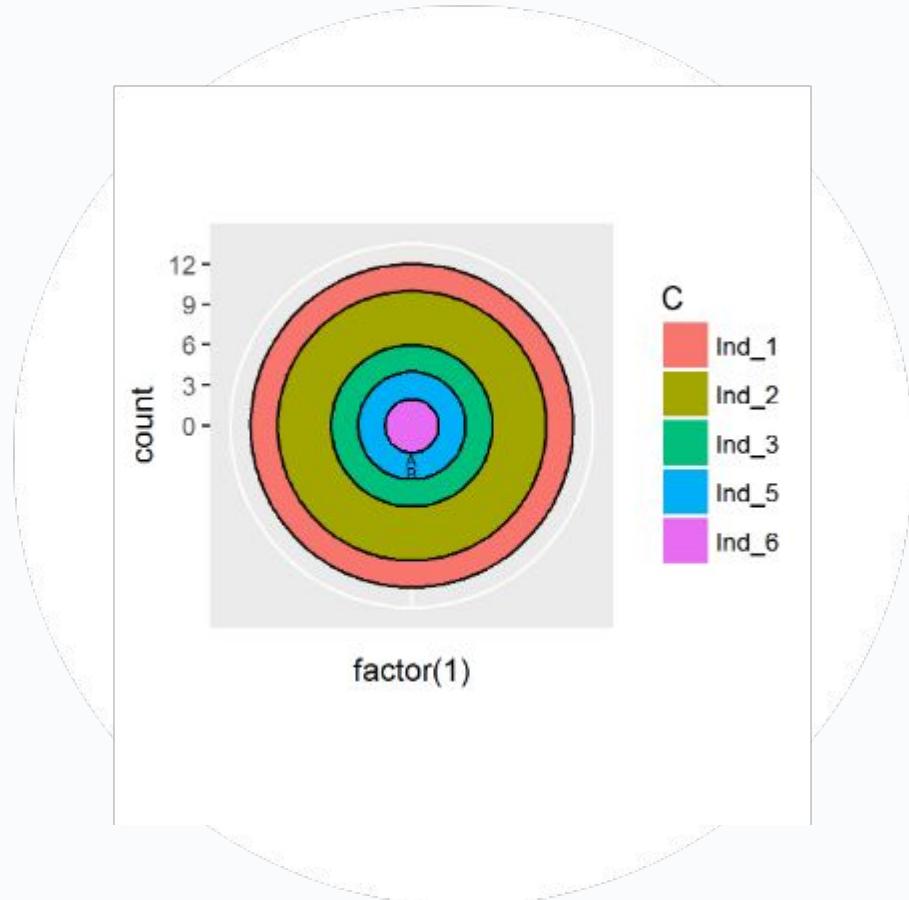
- ✓ Du willst dich nur auf deine **Hauptaufgabe** konzentrieren.
- ✓ Ein **Butler (Proxy)** steht vor deinem Büro.
- ✓ Er nimmt alle Anrufe entgegen.
- ✓ Er prüft Termine, filtert Spam und erinnert dich an Pausen.

**Ergebnis:** Du arbeitest effizienter, ohne dich um "Kleinkram" zu kümmern.



# Die Ring-Metapher

---



## Schutzhülle für deinen Code

Spring baut einen **Ring** um dein Zielobjekt.

- ✓ **Kern:** Dein eigentlicher Java-Code.
- ✓ **Schale:** Der Proxy von Spring.
- ✓ Jeder Aufruf geht erst durch die Schale.

*„Proxies sind wie Ringe um einen Kern.“ –  
Kapitel 4, Buch*

# Was leistet ein Proxy?

---



## Vorverarbeitung

Prüft Parameter, startet Transaktionen oder prüft Rechte.



## Umleitung

Kann Aufrufe blockieren, asynchron auslagern oder wiederholen.



## Nachverarbeitung

Speichert Ergebnisse im Cache oder führt Log-Einträge aus.

# 1. Validierung

## Der Türsteher

Mit **@Validated** und **@Valid** prüft der Proxy deine Daten, bevor dein Code sie sieht.

*Beispiel:* Ein Nutzer darf nur gespeichert werden, wenn er über 18 ist. Der Proxy weist den Aufruf ab, wenn die Regel verletzt wird.



## 2. Caching

### Das Gedächtnis

Mit **@Cacheable** merkt sich der Proxy Ergebnisse teurer Abfragen.

Wenn du das Gleiche nochmal fragst, schaut der Proxy in sein "Notizbuch" und antwortet sofort. Dein Code muss nicht erneut rechnen.



# 3. Asynchronität

## Das Hinterzimmer

Mit **@Async** lagert der Proxy die Arbeit in den Hintergrund aus.

Dein Programm läuft vorne weiter, während der Proxy im Hintergrund die Aufgabe (z.B. E-Mail Versand) erledigt.



**LOREM IPSUM**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ultrices neque orci, sed uktrices  
Leo aliquet non. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ultrices neque  
orci, sed uktrices Leo aliquet non.

## 4. Retry

### Die Beharrlichkeit

Mit **@Retryable** gibt der Proxy nicht sofort auf.

Wenn das Internet kurz hakt, versucht der Proxy es einfach 3-mal neu, ohne dass du eine Fehlermeldung schreiben musst.



# Zwei Arten von Proxies

---

## JDK Dynamic Proxy

Wird genutzt, wenn deine Klasse ein  
**Interface** (Schnittstelle) besitzt.  
Standard-Java-Technik.

## CGLIB Proxy

Wird genutzt, wenn es kein Interface gibt.  
Spring erstellt eine Unterklasse deiner  
**Klasse zur Laufzeit.**



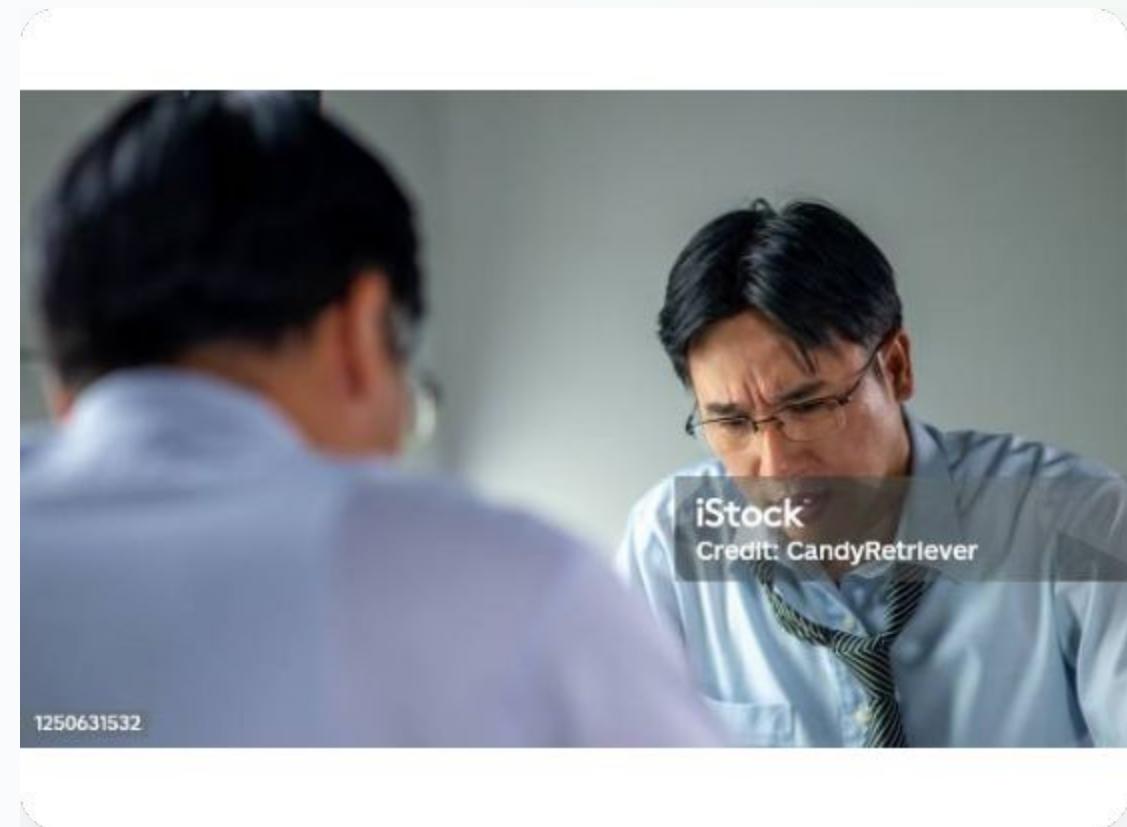
# Die goldene Regel

---

## Vermeide Self-Invocation!

Das ist der häufigste Fehler bei Anfängern:

- ✓ Ein Proxy wirkt nur, wenn der Aufruf **von außen** kommt.
- ✓ Rufst du eine Methode innerhalb der gleichen Klasse auf, wird der Ring übersprungen.
- ✓ **Ergebnis:** @Async oder @Cacheable funktionieren einfach nicht.



# Übersicht: Die wichtigsten Helfer

---

Annotation	Zweck	Proxy-Aufgabe
@Validated	Sicherheit	Prüft Eingabewerte vorab
@Cacheable	Performance	Speichert & liefert bekannte Werte
@Async	Reaktionszeit	Startet Hintergrund-Threads
@Retryable	Stabilität	Wiederholt Aufrufe bei Fehlern



**Alles klar?**

Spring Proxies sind wie Butler: Sie halten dir den Rücken frei.

Bereit für die  
**90-Minuten-Praxis-Session?**

# Image Sources

---



[https://png.pngtree.com/png-vector/20240611/ourlarge/pngtree-green-energy-symbol-with-leaves-illustration-vector-png-image\\_12690357.png](https://png.pngtree.com/png-vector/20240611/ourlarge/pngtree-green-energy-symbol-with-leaves-illustration-vector-png-image_12690357.png)

Source: [pngtree.com](http://pngtree.com)



[https://static.vecteezy.com/system/resources/previews/015/913/423/non\\_2x/senior-butler-icon-isometric-style-vector.jpg](https://static.vecteezy.com/system/resources/previews/015/913/423/non_2x/senior-butler-icon-isometric-style-vector.jpg)

Source: [www.vecteezy.com](http://www.vecteezy.com)



<https://i.sstatic.net/nWCxy.png>

Source: [stackoverflow.com](http://stackoverflow.com)



[https://media.istockphoto.com/id/1200121615/vector/vector-illustration-of-two-security-guards-standing-at-night-at-the-entrance-to-the-half.jpg?s=612x612&w=0&k=20&c=y6taCfPphqbEzSQ13zxJUV98N5SoiqKlbk\\_kBRMwRiU=](https://media.istockphoto.com/id/1200121615/vector/vector-illustration-of-two-security-guards-standing-at-night-at-the-entrance-to-the-half.jpg?s=612x612&w=0&k=20&c=y6taCfPphqbEzSQ13zxJUV98N5SoiqKlbk_kBRMwRiU=)

Source: [www.istockphoto.com](http://www.istockphoto.com)



<https://media.istockphoto.com/id/1224035121/vector/drawer-icon-tray-vector.jpg?s=612x612&w=0&k=20&c=XTWLF-JZrWvO2kh-Gx7BJaO9O8c1DsBzjxxwB4cTNGM=>

Source: [www.istockphoto.com](http://www.istockphoto.com)



[https://static.vecteezy.com/system/resources/thumbnails/014/399/573/small\\_2x/multitasking-and-time-management-concept-working-person-or-business-manager-standing-on-big-clock-doing-effective-multitasking-with-many-hands-productivity-concept-flat-illustration-vector.jpg](https://static.vecteezy.com/system/resources/thumbnails/014/399/573/small_2x/multitasking-and-time-management-concept-working-person-or-business-manager-standing-on-big-clock-doing-effective-multitasking-with-many-hands-productivity-concept-flat-illustration-vector.jpg)

Source: [www.vecteezy.com](http://www.vecteezy.com)

# Image Sources

---



<https://www.iconpacks.net/icons/free-icons-7/free-restart-loop-green-arrow-icon-24696-thumb.png>

Source: [www.iconpacks.net](http://www.iconpacks.net)

---



<https://www.lego.com/cdn/cs/set/assets/blt0b9ab53e6724e980/Balance.jpg?fit=crop&quality=80&width=700&dpr=1>

Source: [www.lego.com](http://www.lego.com)

---



<https://media.istockphoto.com/id/1250631532/photo/stressed-asian-businessman-standing-in-front-of-the-mirror-and-blame-himself-of-mistake.jpg?s=612x612&w=is&k=20&c=d0s2jIL0ZxGTWxsZGVXO5OCd0MJr2hrFlsHjs4751c=>

Source: [www.istockphoto.com](http://www.istockphoto.com)

---



[https://img.freepik.com/free-vector/team-programmers-working-program-code-with-laptops-teamwork-male-female-professional-testers-coders-flat-vector-illustration-software-development-programming-lesson-concept\\_74855-22051.jpg?semt=ais\\_hybrid&w=740&q=80](https://img.freepik.com/free-vector/team-programmers-working-program-code-with-laptops-teamwork-male-female-professional-testers-coders-flat-vector-illustration-software-development-programming-lesson-concept_74855-22051.jpg?semt=ais_hybrid&w=740&q=80)

Source: [www.freepik.com](http://www.freepik.com)