



Sicherer Anschlussknoten des Data Providers Integrationsanleitung Bundesverwaltungsamt

Version 1.0.1, Datum 13.11.2025

Inhaltsverzeichnis

Änderungshistorie	1
1. Einleitung	1
2. Überblick	2
2.1. Fachliche Beschreibung	2
2.2. Technische Beschreibung	2
3. Installation der Anwendung	4
3.1. Installation	4
4. Regulärer Wirkbetrieb	8
4.1. Logging	8
4.2. Statusprüfung der Anwendungen	8
4.3. Monitoring	8
4.4. Backup und Recovery	9
4.5. Hinweise für Störungsdiagnose und -behandlung	9
Anhang	10
Anhang A: Technisches Glossar	10
Anhang B: Fachliches Glossar	10
Anhang C: Betrieblich-Relevante Konfigurationsparameter	11
Quellen	15

Änderungshistorie

Version	Änderungen
0.1	• Initiale Erstellung des Dokuments
0.2	• Einarbeitung Review Kommentare
0.9	• Dokument finalisiert
1.0	• Dokument abgenommen
1.0.1	• Anpassungen in der Beispiel-Konfigurationsdatei • Hinzufügen von Konfigurationsparametern für das Rate-Limiting der Anwendung • Hinzufügen von Konfigurationsparametern für einen Truststore mit erlaubten Client-Zertifikaten ohne Zertifikatskette (Pinning) • Anpassung von neuen Links und Referenzen auf OpenCode

1. Einleitung

Das vorliegende Dokument enthält die Integrationsanleitung für die Anwendung Sicherer Anschlussknoten des Data Providers in der jeweils aktuell veröffentlichten Version. Es fungiert als Hilfestellung für den Betrieb der Anwendung.

Das Kapitel [Überblick](#) bietet einen groben fachlichen, technischen sowie betrieblichen Überblick über die Anwendung und liefert Verweise auf weitergehende Dokumente.

Das Kapitel [Installation der Anwendung](#) beschreibt die Installation und Konfiguration der Anwendung in der Laufzeitumgebung.

Relevant für Betrieb und Softwareentwicklung

Das Kapitel [Regulärer Wirkbetrieb](#) liefert Informationen für den Betrieb der Anwendung und gibt Hilfestellung bei der Fehleranalyse

Relevant für Betrieb und Softwareentwicklung

Der [Anhang](#) enthält ein fachliches und technisches Glossar.

2. Überblick

In diesem Kapitel wird aus fachlicher und technischer Sicht ein Überblick über die Geschäftsprozesse gegeben, die mithilfe des SAK-DP durchgeführt werden.

2.1. Fachliche Beschreibung

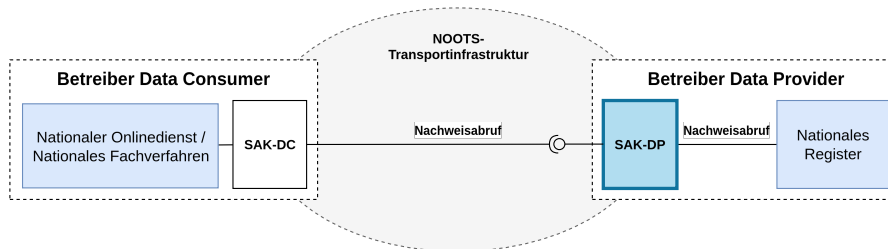


Abbildung 1. SAK-DP: Fachliche Kontextabgrenzung

Die fachlichen Aufgaben von SAK-DP sind:

- Prüfung von Identität und Berechtigung der abfragenden Stelle
- Prüfung und Weitergabe der Abruf-Anfrage an den Data Provider
- Weiterleitung des angefragten Nachweises an die anfragende Stelle

2.2. Technische Beschreibung

SAK-DP stellt Komponenten des NOOTS-Netzwerks einen sicheren Zugang zu einem Data Provider bereit. Hierzu fungiert die Anwendung als "Application Layer Gateway" durch:

- Bereitstellung einer öffentlichen REST-API (Transportprotokoll) für andere SAK-DC im Internet.
- Anbindung des Data Provider via das Anschlussprotokoll.

Zur Absicherung der Zugriffe auf Data Providers via das Anschlussprotokoll kommen "Basic Auth" und "OAuth 2.0" zum Einsatz, zur Absicherung der Netzwerk-Kommunikation "(m)TLS".

Die hierzu benötigten Zugangsdaten und Zertifikate müssen bei der Installation vom SAK-DP bereitgestellt werden.

HINWEIS

- Die Kommunikation des SAK-DP erfolgt direkt mit dessen Data Provider und den SAKs der Data Consumer.
- Der europäische Datenfluss ist nicht Teil des MVP.

Weitere technische Details können im [Anschlusskonzept Data Provider \(AD-NOOTS-02\)](#) gefunden werden.

nachgelesen werden.

3. Installation der Anwendung

Dieses Kapitel beschreibt die Installation der Anwendung.

3.1. Installation

Die nachfolgende Installationsanleitung beschreibt die Erstinstallation auf einem Linux-Server (keine Benutzeroberfläche) mit Zugriff mittels **ssh**.

WICHTIG

Halten Sie sich hier an Ihre betrieblichen Vorgaben für einen sicheren Betrieb. Die hier beschriebenen Schritte dienen nur als Beispiel.

Der SAK-DP wird als ausführbare JAR-Datei (Java Archive) bereitgestellt.

Die JAR-Datei sollte in der Laufzeitumgebung abgelegt werden, in unserem Beispiel ist das **/opt/sak-dp**

Kopieren Sie die JAR-Datei mit der entsprechenden Version auf den Zielrechner, beispielsweise mit **scp**:

```
scp sak-dp-VERSION-TAG.jar admin@laufzeitumgebungshost:~
```

Loggen Sie sich per **ssh** auf dem Rechner ein:

```
ssh admin@laufzeitumgebungshost:~
```

Die Berechtigung sollte nur auf das Nötigste beschränkt werden, damit nur der ausführende User der Serveranwendung Leserechte hat. In unserem Beispiel läuft die Serveranwendung mit dem User **sak-dp** der der Linux Gruppe **sak-dp** zugehört.

```
sudo mkdir /opt/sak-dp
sudo chown root.sak-dp /opt/sak-dp
sudo chmod 750 /opt/sak-dp

sudo mv ~/sak-dp-VERSION-TAG.jar /opt/sak-dp

sudo chown root.sak-dp /opt/sak-dp/sak-dp-VERSION-
TAG.jar
sudo chmod 640 /opt/sak-dp/sak-dp-VERSION-TAG.jar
```

WICHTIG

Die Anwendung sollte unter keinen Umständen über den **root**-User des Systems, sondern idealerweise

mittels eines eigenen Benutzers mit möglichst minimalen Rechten ausgeführt werden.

3.1.1. Systemvoraussetzungen

Die Systemvoraussetzungen für den Betrieb sind abhängig von der Anzahl der gleichzeitigen Zugriffe auf den SAK-DP. Je nach Mengengerüst können diese erheblich von den Mindestanforderungen abweichen.

Tabelle 1. Systemvoraussetzungen

Name	Mindestanforderungen
Java-JRE	JRE 21, getestet mit Adoptium Temurin
Betriebssystem	Linux amd64
Arbeitsspeicher	1024MB
Speicherplatz	512MB
Prozessor	2 GHz Dual Core, 64-Bit
Netzwerk	Internetverbindung (direkt oder über Proxy) auf Port 443

3.1.2. Start und Konfiguration der Anwendung

Die Konfigurationen des SAK-DP findet mittels einer bereitgestellten `application.properties` statt, welche auf die Laufzeitumgebung angepasst werden muss. Die Parameter sind in der Datei beschrieben und enthalten Beispielwerte, an denen sich orientiert werden kann. Die Konfigurationsdatei muss als Parameter übergeben werden. Die Dateirechte auf die Konfigurationsdatei sollten auch hier so gesetzt werden, sodass nur die Anwendung darauf zugreifen kann.

```
java -jar sak-dp-VERSION-TAG.jar
--spring.config.additional
-location=application.properties
```

Weiterführende Hinweise zur Konfiguration

Im oben genannten Beispiel wird davon ausgegangen, dass die `application.properties`-Datei im identischen Verzeichnis wie die Anwendung (.jar) selbst liegt. Wenn die Konfiguration aus einem anderen Verzeichnis geladen werden soll, kann dieses sowohl relational, sowie absolut angegeben werden. Außerdem können mehrere Dateien, auch *optional* angegeben werden. Hier muss die

Reihenfolge beachtet werden. Dateien, welche früher in der Kette angegeben werden, werden durch spätere Dateien überschrieben (je nach Konfigurationsschlüssel)

```
java -jar sak-dp-VERSION-TAG.jar
--spring.config.additional-location=\

    #Konfigurations-Datei im gleichen Ordner; muss
    vorhanden sein
    application.properties,\

    #externe Konfigurations-Datei ; muss vorhanden
    sein
    /opt/default.properties,\

    #externe Konfigurations-Datei ; optional,
    produziert keinen Fehler, wenn nicht vorhanden
    optional:/opt/optional.properties
```

Weitere Informationen zu der Angabe von externen Konfigurationen können der offiziellen [Spring Boot Dokumentation](#); Kapitel "Externalized Configuration"; Abschnitt "External Application Properties" entnommen werden.

Die Anwendung sollte nun gestartet sein und folgende Meldung in der Ausgabe erscheinen:

Komponente SAK-DP erfolgreich gestartet

3.1.3. Endpunkte

In der Standardkonfiguration ist die Serveranwendung über Port **8080** sowie der Managementendpunkt über Port **9000** erreichbar.

WICHTIG

Diese Netzwerkports sollten über geeignete Firewallregeln abgesichert werden, um unbefugten Zugriff zu unterbinden. Halten Sie sich auch hier an Ihre betrieblichen Vorgaben für einen sicheren Betrieb.

In den offiziellen Test- und Produktionsumgebungen von NOOTS ist es verpflichtend, die Endpunkte nicht ungeschützt und unverschlüsselt zu betreiben. Es können hierbei verschiedenen Varianten genutzt werden. Je nach vorhandener Infrastruktur kann auch ein Betrieb hinter einem Proxy sinnvoll sein. In einer lokalen Teststellung ohne Anbindung gegen das NOOTS-System (z.B. mittels Mocks) können die Endpunkte auch weniger

abgesichert betrieben werden.

Tabelle 2. Absicherung der Endpunkte

Art	Erklärung
HTTPS	Die Konfiguration erfolgt über die <code>application.properties</code>
mTLS	Die Konfiguration erfolgt über die <code>application.properties</code>
HTTP Basic-Auth	Die Konfiguration erfolgt über die <code>application.properties</code>
Die obigen Methoden können bei Bedarf durch Apache-, Nginx oder einen anderen geeigneten Reverse Proxy abgebildet werden	Die Konfiguration erfolgt über das externe genutzte System

3.1.4. Start und Stop

Damit die Anwendung nicht immer manuell über die Konsole gestartet werden muss, können Mittel wie zum Beispiel `systemd` oder `init.d` genutzt werden.

Wir empfehlen hierzu die offizielle [Spring Boot Dokumentation; Kapitel "Installing Spring Boot Applications"](#).

4. Regulärer Wirkbetrieb

4.1. Logging

Der SAK-DP verwendet standardmäßig die Konsole für Logausgaben. Das Log-Nachrichtenformat entspricht dem [Elastic Common Schema \(ECS\)](#).

4.2. Statusprüfung der Anwendungen

Der SAK-DP stellt die folgenden Endpunkte über den Management Port zur Statusprüfung bereit:

HINWEIS

Diese Endpunkte sollten nur intern (z.B. hinter einer Firewall) betrieben werden. Eine Freigabe für einen öffentlichen, nicht regulierten Zugriff ist zu vermeiden!

Tabelle 3. Status-Endpunkte

Endpunkt	Beschreibung
<code>/actuator/health</code>	Gesamtstatus der Anwendung. Liefert <code>{"status":"UP DOWN"}</code> als Antwort

4.3. Monitoring

Der SAK-DC kann zusätzliche Metriken liefern. Es stehen folgende Endpunkte zur Verfügung:

- `/metrics`
- `/prometheus` ([Spring Boot Dokumentation](#); Kapitel "Prometheus (prometheus)")

Die Endpunkte können über die `application.properties` aktiviert werden und sind über den Management-Port erreichbar.

Es werden folgende Standard-Metriken bereitgestellt:

HINWEIS

Diese Endpunkte sollten nur intern (z.B. hinter einer Firewall) betrieben werden. Eine Freigabe für einen öffentlichen, nicht regulierten Zugriff ist zu vermeiden!

Tabelle 4. Standard-Metriken

Metriken	Beschreibung	Path
JVM	Metriken zum Speicherverbrauch, zur Garbage Collection, Threads und geladenen Klassen.	<code>/actuator/metrics/jvm/*</code>
System	Metriken zur CPU-Auslastung, dem Dateisystem und der Uptime der Anwendung.	<code>/actuator/metrics/disk/ /actuator/metrics/system/ /actuator/metrics/process/*</code>
Application Startup	Metriken zur Startzeit der Anwendung.	<code>/actuator/metrics/application/*</code>
Logger	Metriken zu Logging-Events.	<code>/actuator/metrics/logback/events</code>
REST Services	Metriken zu aufgerufenen REST-Services.	<code>/actuator/metrics/http/*</code>

4.4. Backup und Recovery

Wir empfehlen mindestens folgende Daten zu sichern:

- Angepasste `application.properties` die für den Betrieb verwendet wird.
- Logs der Anwendung unter Beachtung der geltenden Datenschutzbestimmungen.

4.5. Hinweise für Störungsdiagnose und -behandlung

4.5.1. Debug-Logging

Die Konfiguration wird mit Log-Level **INFO** ausgeliefert, d.h. das Loggen von Debug-Ausgaben ist ausgeschaltet. Es kann aber mittels der `application.properties logging.level.root=DEBUG` eingeschaltet werden.

ACHTUNG

Das Logging sollte auf der Produktivumgebung nie auf **DEBUG** gestellt werden. Aufgrund der umfangreichen Datenausgabe könnte es zu Performance-Problemen kommen. Gleichzeitig ist nicht sichergestellt, dass keine sicherheitsrelevanten Daten ausgegeben werden.

Anhang

Anhang A: Technisches Glossar

HINWEIS

Das hier hinterlegte Glossar ist übergreifend für alle Dokumente des SAKs identisch

Tabelle 5. Technisches Glossar SAK-DP

Begriff	Bedeutung
Laufzeitumgebung	Umgebung, in der ein Service oder eine Anwendung betrieben werden kann.

Anhang B: Fachliches Glossar

HINWEIS

Das hier hinterlegte Glossar ist übergreifend für alle Dokumente des SAKs identisch

Tabelle 6. Fachliches Glossar SAK-DP

Begriff	Bedeutung
Data Consumer	Ein Data Consumer ist ein NOOTS-Teilnehmer, welcher Nachweise aus dem System, konkret den Data Providern, abrufen möchte.
Data Provider	Ein Data Provider ist ein NOOTS-Teilnehmer der den Data Consumern Nachweise zur Verfügung stellt.
Identity Access Management für Behörden (IAM-B)	Das IAM für Behörden ist eine zentrale Komponente des NOOTS. Sie stellt für den Zugriff auf andere NOOTS-Komponenten und Data Provider vertrauenswürdige und zuverlässige Zugriffstoken für Data Consumer aus, die den Zugriff des DC auf die NOOTS-Komponenten und die DPs autorisiert, die Identität des DCs bestätigt und die Berechtigungen des DCs gegenüber den NOOTS-Komponenten und DPs beinhaltet.
Registerdatennavigation (RDN)	Die Registerdatennavigation ist eine zentrale Komponente des NOOTS. Sie liefert auf Anfrage die Information, von welchem technischen Dienst einer Behörde ein gesuchter Nachweistyp abgerufen werden kann. Dazu übermittelt die RDN notwendige Verbindungsparameter an den anfragenden Data Consumer.
Nachweis	Nachweise sind Unterlagen und Daten in elektronischer Form, die zur Ermittlung des Sachverhaltes in Verwaltungsverfahren geeignet sind.

Begriff	Bedeutung
National-Once-Only-Technical-System (NOOTS)	Das NOOTS ist ein gemeinsames informationstechnisches System aus IT-Komponenten, Schnittstellen und Standards, das öffentlichen Stellen den Abruf und die Übermittlung von elektronischen Nachweisen und Daten national und grenzüberschreitend aus Datenbeständen öffentlicher Stellen, zur Erfüllung öffentlicher Aufgaben ermöglicht.
Sicherer Anschlussknoten (SAK)	Ein sicherer Anschlussknoten (SAK) ist eine für Data Consumer - und Provider bereitgestellte NOOTS-Komponente, die dezentral im Verantwortungsbereich der NOOTS-Teilnehmer betrieben wird. Er ermöglicht den NOOTS-Teilnehmern über ein interoperables Anschlussprotokoll einen einfachen und sicheren Zugriff auf die NOOTS Transportinfrastruktur.

Anhang C: Betrieblich-Relevante Konfigurationsparameter

```
# -----
# SAK-DP Transportprotokoll Einstellungen
# -----

# SAK-Server Adresse
server.address = localhost
# SAK-Server Port
server.port = 8443

# max. erlaubte Anzahl von Request, die SAK-DP innerhalb
# des gegebenen Zeitfensters (in Millisekunden) annimmt.
#sakdp.ratelimiter.limitForPeriod = 50
#sakdp.ratelimiter.limitRefreshPeriod = 1000

# SAK-Server TLS
# Hardware-basierte Schlüsselspeicher (empfohlen)
spring.ssl.bundle.jks.server.keystore.type = PKCS11
spring.ssl.bundle.jks.server.keystore.password =
spring.ssl.bundle.jks.server.key.alias = sakdp-server-
crt
# Datei-basierte Schlüsselspeicher (nicht empfohlen,
# bitte nur PKCS12 in diesem Fall verwenden)
#spring.ssl.bundle.jks.server.keystore.type = PKCS12
#spring.ssl.bundle.jks.server.keystore.location = sakdp-
server-cert.p12
#spring.ssl.bundle.jks.server.keystore.password =
```

```
# Separate TLS Server-Zertifikaten, falls der SAK unter
# unterschiedlichen Hostnames exponiert ist
#server.ssl.server-name-bundles[0].server-name = sak-
#dp1.example.com
#server.ssl.server-name-bundles[0].bundle = sakdp1
#spring.ssl.bundle.jks.sakdp1.keystore.type = PKCS11
#spring.ssl.bundle.jks.sakdp1.keystore.password =
#spring.ssl.bundle.jks.sakdp1.key.alias = sakdp1-server-
#cert
#spring.ssl.bundle.jks.sakdp1.options.enabled-protocols
# = ${sakdp.server.ssl.enabled-protocols}
#spring.ssl.bundle.jks.sakdp1.options.ciphers =
# ${sakdp.server.ssl.enabled-ciphers}

# SAK-Server TLS Client-Auth
# Truststore mit vertrauenswürdigen Client-Cert-CAs
spring.ssl.bundle.jks.server.truststore.type = PKCS12
spring.ssl.bundle.jks.server.truststore.location =
# sakdp-server-truststore.p12
spring.ssl.bundle.jks.server.truststore.password =

# Truststore mit erlaubten Client-Zertifikaten ohne
# Zertifikatskette (Pinning)
spring.ssl.bundle.jks.sakdcclientcerts.truststore.type =
# PKCS12
spring.ssl.bundle.jks.sakdcclientcerts.truststore.locati
# on = sakdc-client-certs.p12
spring.ssl.bundle.jks.sakdcclientcerts.truststore.passwo
# rd =

# Truststore-Konfiguration für die NOOTS-Zugriffstoken-
# Validierung
spring.ssl.bundle.jks.iamsigncert.truststore.type =
# PKCS12
spring.ssl.bundle.jks.iamsigncert.truststore.location =
# iam-sign-cert.p12
spring.ssl.bundle.jks.iamsigncert.truststore.password =

# NOOTS IAM Hostname
noots.iam.hostname = iam.noots.gov.de

# -----
# SAK-DP Anschlussprotokoll Einstellungen
# -----

# Einstellung für die Verbindung gegen den hinterlegten
# Data-Provider
```

```
#
-----

# Verbindungsparameter des Data-Providers
# Anfragen gegen den Data-Provider werden mit folgendem
Schema durchgeführt:
# {{dataproducer.url}}{{dataproducer.resource}}/evidence
# Dataprovider Host-URL, ggfs. mit Port
dataproducer.url = https://dp-implementierung-url:8443
# Resource-Path des Request. Dieser wird als Prefix für
die eigentliche URL verwendet.
# Derzeit muss immer ein Prefix-Path angegeben werden.
Leerwerte wie "" oder "/" sind nicht zulässig.
dataproducer.resource = /api

# Optionale Basic-Auth Authentifizierung des Data-
Providers
# Schnittstellen des Data-Provider können laut der
Schnittstellenspezifikation optional via
# Basic-Auth gesichert werden. Sollte dies der Fall
sein, muss hier
# "dataproducer.basicauth.enabled" auf "true" gesetzt,
und Username + Passwort angegeben werden.
dataproducer.basicauth.enabled = true
dataproducer.basicauth.username =
dataproducer.basicauth.password =

# Optionale Proxy-Konfiguration für die Verbindung
zwischen SAK-DP und DP
#
-----

#spring.cloud.gateway.server.webflux.httpclient.proxy.ho
st = ""
#spring.cloud.gateway.server.webflux.httpclient.proxy.po
rt = ""
# Optionale Authentifizierung des Proxies
#spring.cloud.gateway.server.webflux.httpclient.proxy.us
ername =
#spring.cloud.gateway.server.webflux.httpclient.proxy.pa
ssword =

# Client-Zertifikate des SAK
# Hardware-basierte Schlüsselspeicher (empfohlen)
spring.ssl.bundle.jks.client.keystore.type = PKCS11
spring.ssl.bundle.jks.client.keystore.password =
spring.ssl.bundle.jks.client.key.alias = sakdp-client-
```

```
crt
# Datei-basierte Schlüsselspeicher (nicht empfohlen,
# bitte nur PKCS12 in diesem Fall verwenden)
#spring.ssl.bundle.jks.client.keystore.type = PKCS12
#spring.ssl.bundle.jks.client.keystore.location = sakdp-
client-cert.p12
#spring.ssl.bundle.jks.client.keystore.password =

spring.ssl.bundle.jks.client.truststore.type = PKCS12
spring.ssl.bundle.jks.client.truststore.location =
sakdp-client-truststore.p12
spring.ssl.bundle.jks.client.truststore.password =

# -----
# SAK-DP Allgemeine Anwendungskonfiguration
# -----

# Monitoring-Endpunkte der Anwendung
# Die folgenden Endpunkte werden über die URL
"http://{{management.server.address}}:{{management.serve
r.port}}/actuator/{{endpoint}}" auf dem
# konfigurierten Management-Port standardmäßig zur
Verfügung gestellt und können über den
Konfigurationswert
# "management.endpoints.web.exposure.include" selektiv
aktiviert und damit exponiert werden:
# - health: Der Health-Endpunkt stellt Informationen
über die Readiness und Liveness der Anwendung bereit
# - prometheus: Der Prometheus-Endpunkt stellt eine
Vielzahl an Metriken im Prometheus-Format zur Verfügung
# - info: Der Info-Endpunkt liefert interne
Informationen der Anwendung, z.B. Daten zum Build oder
zur aktuellen Laufzeitumgebung
# - sbom: Der SBOM-Endpunkt liefert eine Auflistung
aller Abhängigkeiten und Bibliotheken im SAK-DP
management.endpoints.web.exposure.include =
health,prometheus
# Verbindungseinstellungen für die Management-Endpoints
management.server.address = localhost
management.server.port = 9000

# Logging-Einstellungen
# Mögliche Log-Level: ERROR, WARN, INFO, DEBUG, TRACE
# In einer Produktivumgebung sollte immer min. WARN
gesetzt werden
# Basis-Level für alle Logmeldungen der Anwendung
# (spezifische Log-Level überschreiben die Basis)
```

```
logging.level.root = INFO
```

Quellen

Tabelle 7. Quellen

Name der Quelle	Link	Version
Anschlusskonzept Data Consumer (AD-NOOTS-01)	https://gitlab.opencode.de/noots/public/ad-noots/noots-architektur/ad-noots-release/-/blob/main/AD-NOOTS-XX/AD-NOOTS-01_+Anschlusskonzept+Data+Consumer+_DC_.md	Release Februar 2025
Anschlusskonzept Data Provider (AD-NOOTS-02)	https://gitlab.opencode.de/noots/public/ad-noots/noots-architektur/ad-noots-release/-/blob/main/AD-NOOTS-XX/AD-NOOTS-02_+Anschlusskonzept+Data+Provider+_DP_.md	Release Februar 2025
Spring Boot Dokumentation ; Kapitel "Installing Spring Boot Applications"	https://docs.spring.io/spring-boot/3.5/how-to/deployment/installing.html	Version 3.5.5
Spring Boot Dokumentation ; Kapitel "Externalized Configuration"; Abschnitt "External Application Properties"	https://docs.spring.io/spring-boot/reference/features/external-config.html#features.external-config.files	Version 3.5.5
Spring Boot Dokumentation ; Kapitel "Prometheus (prometheus)"	https://docs.spring.io/spring-boot/api/rest/actuator/prometheus.html	Version 3.5.5

Name der Quelle	Link	Version
Kubernetes Dokumentation ; Kapitel "Pod Lifecycle"; Abschnitt "Types of probe"	https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#types-of-probe	