Konfigurationsdatei webdesk.properties und Umgebungsvariablen

Die Datei webdesk.properties ist die Startkonfiguration von Webdesk 4. Sie enthält die Verbindungsdaten zur Webdesk- und zur Shark-Datenbank, die Einstellungen zur Verschlüsselung dieser Zugangsdaten und einige Schalter, die das Verhalten beim Start des Systems steuern. Diese Seite richtet sich an Systemadministratoren und Berater, die ein System installieren, auf Container umstellen oder Startprobleme analysieren. Fachliche Einstellungen (Mandanten, Jobs, Berechtigungen usw.) werden nicht hier, sondern in der Anwendung vorgenommen, siehe z. B. Systemparameter Portal & Organisation[1].

Ausgewählte Schlüssel der Datei können zusätzlich über JVM-Parameter (-Dschlüssel=wert) oder Umgebungsvariablen (WD_...) übersteuert werden. Das ist die empfohlene Methode für Container-Betrieb, weil dann keine Zugangsdaten in der Datei stehen müssen.

Ablageort und Aufbau

- Webdesk sucht die Datei im Klassenpfad der Webanwendung. In einer Standardinstallation liegt sie daher unter WEB-INF/classes/webdesk.properties der Webdesk-Anwendung im Servlet-Container (Windows-Installer: <Webdesk-Webapp>\WEB-INF\classes\webdesk.properties).
- Die Datei ist eine Java-Properties-Datei: eine Zeile pro Eintrag in der Form schlüssel = wert, Kommentarzeilen beginnen mit #. Die Datei wird beim Start gelesen, Änderungen werden erst nach einem Neustart wirksam.
- Als Ausgangspunkt dienen die vom Installer ausgelieferte Datei bzw. die Vorlage webdesk.properties.template der Webanwendung. Kundenbeispiele finden sich im Quellcode unter webdesk-webapp/src/test/resources/customerprops.
- Ohne diese Datei startet Webdesk nicht (Fehler „Error while trying to load webdesk.properties“), außer es ist die eingebettete Datenbank aktiviert (siehe webdesk.useEmbeddedDatabase). Auch beim Betrieb mit Umgebungsvariablen muss daher mindestens eine kleine Datei vorhanden sein.
- Webdesk schreibt beim Start selbst in die Datei (Verschlüsselung der Zugangsdaten, Schlüssel, Anpassung veralteter Einträge, Zurücksetzen von webdesk.runRegistration). Der Benutzer, unter dem der Container läuft, benötigt daher Schreibrechte, und die Datei muss bei Container-Betrieb auf einem dauerhaften Speicher liegen. Vor manuellen Änderungen ist eine Sicherung anzulegen.

Minimales Beispiel

```
webdesk.runRegistration = 1

webdesk.connection.driver_class = org.mariadb.jdbc.Driver
webdesk.connection.url = jdbc:mysql://<db-host>/webdesk
webdesk.dialect = org.hibernate.dialect.MySQLDialect
webdesk.connection.username = <db-user>
webdesk.connection.password = <db-password>

shark.connection.driver_class = org.mariadb.jdbc.Driver
shark.connection.url = jdbc:mysql://<db-host>/shark
shark.connection.username = <db-user>
shark.connection.password = <db-password>

webdesk.ta.if6020impl.rmi = rmi://<ta-host>:4009/taRmiBroker
```

Beim ersten Start verschlüsselt Webdesk die Benutzernamen und Passwörter in der Datei, erzeugt (bzw. liest) den Verschlüsselungsschlüssel und trägt webdesk.credentials.encrypted, shark.credentials.encrypted, webdesk.encryption.key und webdesk.encryption.key.source selbst ein (siehe „Verschlüsselung der Zugangsdaten“).

Automatische Anpassungen beim Start

Webdesk korrigiert beim Start folgende veraltete Einträge in der Datei und speichert sie danach neu:

- Alte Schlüssel hibernate.connection.driver_class, hibernate.connection.url, hibernate.connection.username, hibernate.connection.password, hibernate.dialect werden in webdesk.connection.* bzw. webdesk.dialect umbenannt (sofern der neue Schlüssel noch nicht existiert).
- webdesk-SessionFactory.executeIndexManager wird in PoStartup.executeIndexManager umbenannt.
- Der Treiber com.mysql.jdbc.Driver wird durch org.mariadb.jdbc.Driver ersetzt (der MySQL-Treiber wird nicht mitgeliefert).
- Bei JDBC-URLs, die mit jdbc:mysql beginnen, werden die Parameter permitMysqlScheme und allowMultiQueries=true ergänzt.
- Bei JDBC-URLs für SQL Server (jdbc:sqlserver) ohne Parameter encrypt wird ;encrypt=false angehängt; der Dialekt org.hibernate.dialect.SQLServerDialect wird zu org.hibernate.dialect.SQLServer2012Dialect.
- Veraltete Cache-Replikations-Einträge werden entfernt bzw. umbenannt (webdesk.cacheReplication.peerProvider.properties wird zu webdesk.cacheReplication.RMI.peerProvider.properties).

Hinweis: Diese Korrekturen betreffen nur die Datei. Werte, die per Umgebungsvariable oder JVM-Parameter geliefert werden, werden nicht korrigiert und müssen bereits die aktuelle Schreibweise haben (z. B. den Treiber org.mariadb.jdbc.Driver).

Datenbankverbindung

Eigenschaft	Erklärung	mögliche Wert(e)
webdesk.connection.driver_class	JDBC-Treiberklasse der Webdesk-Datenbank.	z. B. org.mariadb.jdbc.Driver (MySQL/MariaDB), com.microsoft.sqlserver.jdbc.SQLServerDriver (SQL Server), oracle.jdbc.OracleDriver (Oracle) (Vorlage: com.mysql.jdbc.Driver, wird beim Start ersetzt)
webdesk.connection.url	JDBC-URL der Webdesk-Datenbank.	z. B. jdbc:mysql://<db-host>/webdesk oder jdbc:sqlserver://<db-host>:1433;databaseName=webdesk (Vorlage: jdbc:mysql://localhost/webdesk)
webdesk.dialect	Hibernate-Dialekt passend zur Datenbank.	z. B. org.hibernate.dialect.MySQLDialect, org.hibernate.dialect.SQLServer2012Dialect, org.hibernate.dialect.OracleDialect (Vorlage: org.hibernate.dialect.MySQLDialect)
webdesk.connection.username webdesk.connection.password	Zugangsdaten der Webdesk-Datenbank. Sind sie verschlüsselt, stehen hier Base64-Texte (siehe unten).	Text (Vorlage: Klartext-Beispielwerte)
webdesk.credentials.encrypted	Gibt an, ob Benutzername und Passwort der Webdesk-Datenbank verschlüsselt in der Datei stehen.	true, false (Standard bei fehlendem Eintrag: Webdesk verschlüsselt und setzt true)

	Fehlt der Eintrag, verschlüsselt Webdesk beim Start die Werte und setzt den Eintrag auf true. Der Eintrag kann nicht per Umgebungsvariable übersteuert werden.	
shark.connection.driver_class shark.connection.url	Treiberklasse und JDBC-URL der Shark-Datenbank (Workflow-Engine). Es gelten die gleichen Hinweise wie oben.	(Vorlage: com.mysql.jdbc.Driver, jdbc:mysql://localhost/shark)
shark.connection.username shark.connection.password	Zugangsdaten der Shark-Datenbank. Fehlen die Einträge (System ohne Workflow-Modul), werden sie als leer angenommen.	Text
shark.credentials.encrypted	Wie webdesk.credentials.encrypted, für die Shark-Datenbank. Fehlt der Eintrag und ist ein Shark-Benutzername in der Datei vorhanden, verschlüsselt Webdesk beim Start; fehlt auch der Benutzername, gilt false.	true, false
webdesk-sessionFactory.schemaUpdate	Legt fest, ob das Datenbankschema beim Start automatisch angepasst wird, wenn das Datenmodell neue Spalten oder Tabellen enthält. Bei false wird das Schema nicht geprüft (kürzere Startzeit, aber nach Updates nicht für Produktivsysteme geeignet, solange Schemaänderungen anstehen).	true, false (Standard: true)
webdesk.exitJvmOnConnectionError	Bei true wird die JVM beendet (Exit-Code 1), wenn beim Start keine Datenbankverbindung aufgebaut werden kann und die Fehlermeldung „Access denied“ oder „Communications link failure“ enthält. Nützlich, damit ein Container-Orchestrator den Container neu startet. Der Wert wird aus der Datei webdesk.properties gelesen; bitte dort setzen.	true, false (Standard: false)
webdesk.useEmbeddedDatabase	Bei true startet Webdesk mit einer eingebetteten HSQLDB statt mit den konfigurierten Datenbanken; die Verbindungsdaten der Datei werden ignoriert. Die Datenbank liegt in WEB-INF/work/db (falls	true, false (Standard: false)

	vorhanden), sonst im temporären Verzeichnis der JVM unter db, und bleibt über Neustarts erhalten. Gedacht für Demo- und Entwicklungszwecke. Es wird dabei ein fester Testschlüssel statt eines geheimen Schlüssels verwendet; für Produktivsysteme nicht einsetzen.	

Verschlüsselung der Zugangsdaten

Webdesk verschlüsselt die Zugangsdaten in webdesk.properties mit einem Schlüssel (AES-256). Derselbe Schlüssel wird auch für andere in der Anwendung gespeicherte Passwörter verwendet, etwa in Konnektoren (Datenbank-, REST-, Notes-Konnektor), der Dokumentenverwaltung (Repositories) und im Groupware-Zugriff (Graph/Exchange). Mehr zur Umstellung der Verschlüsselung finden Sie in Neue Features in 4.56[2].

Achtung: Geht der Schlüssel verloren, können die verschlüsselten Zugangsdaten in webdesk.properties und die verschlüsselt gespeicherten Passwörter in der Datenbank nicht mehr entschlüsselt werden. Webdesk kann dann nicht mehr gestartet werden bzw. die Verbindungen funktionieren nicht mehr. Der Schlüssel (enc.key bzw. der Eintrag webdesk.encryption.key) muss daher unabhängig von der Datenbank-Sicherung separat gesichert werden. Ein Schlüssel, der nur in einer Container-Instanz ohne dauerhaften Speicher liegt, geht beim Austausch des Containers verloren.

Eigenschaft	Erklärung	mögliche Wert(e)
webdesk.encryption.key.source	Gibt an, woher der Schlüssel gelesen wird. webdesk_properties: Der Schlüssel steht Base64-codiert im Eintrag webdesk.encryption.key derselben Datei. file: Der Schlüssel steht in der Datei enc.key (nur Base64-Text, erste Zeile), die im Klassenpfad im selben Verzeichnis wie webdesk.properties liegt. webservice ist im Code nicht umgesetzt. Fehlt der Eintrag, sucht Webdesk den Schlüssel in der Datenbank (Tabelle PoFile, FILE_UID = 'encryptionKey'), erzeugt bei Bedarf einen neuen und trägt ihn mit Quelle webdesk_properties in die Datei ein.	file, webdesk_properties (Standard: nicht gesetzt, siehe Erklärung)
webdesk.encryption.key	Der Schlüssel (Base64), wenn die Quelle webdesk_properties ist. Wird von Webdesk selbst erzeugt und eingetragen. Nicht von Hand ändern: andere Werte machen die vorhandenen verschlüsselten Daten unlesbar.	Base64-Text (vom System erzeugt)

Schlüssel in eine eigene Datei auslagern

Empfohlen ist, den Schlüssel nach dem ersten erfolgreichen Start aus der Konfigurationsdatei in eine eigene Datei enc.key auszulagern und diese getrennt zu sichern:

1. Sicherung von webdesk.properties und der Datenbanken anlegen.
2. Den Wert des Eintrags webdesk.encryption.key als einzige Zeile in eine neue Datei enc.key schreiben und diese im selben Verzeichnis wie webdesk.properties ablegen.
3. In webdesk.properties den Eintrag webdesk.encryption.key.source = file setzen und den Eintrag webdesk.encryption.key entfernen.
4. Webdesk neu starten und prüfen, dass das System startet.
5. enc.key getrennt von der Anwendung sichern.

Verhalten im Cluster

Alle Knoten müssen mit demselben Schlüssel arbeiten. Fehlt auf einem Knoten die Angabe webdesk.encryption.key.source, liest er den Schlüssel aus der Datenbank (PoFile, FILE_UID = 'encryptionKey'); beim Anlegen wird die Tabelle gesperrt, sodass nur ein Schlüssel entsteht. Wird der Schlüssel in enc.key ausgelagert, muss die Datei auf jedem Knoten mit demselben Inhalt vorliegen. Der Datenbankeintrag wird danach nicht mehr verwendet und kann (laut Hinweis in Neue Features in 4.56[3]) mit DELETE FROM PoFile WHERE FILE_UID='encryptionKey'; entfernt werden, erst nachdem alle Knoten erfolgreich mit enc.key gestartet sind. Weitere Cluster-Einstellungen siehe Seite „Cluster- und Cache-Betrieb“.

Kommandozeilenwerkzeug CredentialsEncryptor

Mit dem Werkzeug webdesk-encryptor lassen sich Zugangsdaten in einer webdesk.properties verschlüsseln, ansehen und neu setzen, ohne das System zu starten. Auf Windows-Installationen liegt dafür im Tomcat-Verzeichnis bin das Skript manageWebdeskDBProps.bat, das die Datei und die Programmdatei webdesk-encryptor-*.jar selbst sucht und zwischen „Basic“- und „Expert“-Modus wählen lässt. Direkter Aufruf:

```
java -cp webdesk-encryptor-<version>.jar at.workflow.webdesk.tools.CredentialsEncryptor
 path=<Pfad>/webdesk.properties [webdeskLogin=<user>] [webdeskPassword=<pw>]
 [sharkLogin=<user>] [sharkPassword=<pw>]
```

- Ohne path wird webdesk.properties im aktuellen Arbeitsverzeichnis verwendet.
- Die Parameter webdeskLogin, webdeskPassword, sharkLogin, sharkPassword setzen den jeweiligen Benutzernamen bzw. das Passwort neu (im Klartext angegeben, das Werkzeug verschlüsselt sie).
- Sind die Zugangsdaten noch nicht verschlüsselt, werden sie verschlüsselt und webdesk.credentials.encrypted / shark.credentials.encrypted gesetzt; Werte im alten Verschlüsselungsverfahren werden neu verschlüsselt.
- Danach zeigt das Werkzeug ein Menü: Werte entschlüsselt anzeigen, verschlüsselte Werte anzeigen, alle vier Werte neu eingeben, beenden.
- Enthält die Datei noch keine Schlüsselquelle, erzeugt das Werkzeug einen neuen Schlüssel und trägt ihn mit der Quelle webdesk_properties ein. Das Werkzeug unterstützt nur diese Quelle; bei webdesk.encryption.key.source = file bricht es mit einer Fehlermeldung ab.

Achtung: Wird das Werkzeug auf eine Datei angewendet, die zu einem bereits laufenden System gehört, aber keinen Schlüssel enthält, entsteht ein neuer, anderer Schlüssel als der im System verwendete. Verwenden Sie das Werkzeug nur mit der Original-Datei des Systems samt Schlüssel.

Startverhalten

Eigenschaft	Erklärung	mögliche Wert(e)
webdesk.runRegistration	Legt fest, ob beim Start Aktionen, Texte (Textbausteine) und weitere Artefakte der Module in der Datenbank registriert werden. 0: nie. 1: einmal, danach setzt Webdesk den Eintrag in der Datei auf 0 zurück. 2: bei jedem Start vollständig. 3: bei jedem Start, aber ohne Textbausteine. 4: wie 2, danach wird Webdesk beendet (nur für automatisierte Deployments und Updates in Kubernetes). Ein ungültiger Wert wirkt wie 1. Ist die Datenbank noch leer, wird die Registrierung einmal erzwungen und vor der Nutzung abgeschlossen.	0, 1, 2, 3, 4 (Standard bei fehlendem Eintrag: 1; Vorlage: 1)
webdesk.runRegistrationInThread	Bei false läuft die Registrierung im Vordergrund, d. h. der Start dauert länger, bis sie abgeschlossen ist. Standardmäßig läuft sie in einem Hintergrund-Thread.	true, false (Standard: true)
PoStartup.executeIndexManager	Ob beim Start alle Indizes der Webdesk-Datenbank mit den erwarteten verglichen und fehlende angelegt werden. Kann den Start deutlich verlängern. Für die Shark-Datenbank gilt analog WfStartup.executeIndexManager (Standard: true). Die Vorlage verwendet noch den alten Namen webdesk-SessionFactory.executeIndexManager, der beim Start umbenannt wird.	true, false (Standard: true; Vorlage: false)
webdesk.disableTimeLineChecks	Bei true entfällt beim Start die Konsistenzprüfung der Zeitleisten (Historisierung) der Personen. Die Prüfung kann länger als 30 Sekunden dauern; eine Abschaltung verzichtet auf die Sicherheitsprüfung von Inkonsistenzen in den Stammdaten.	true, false (Standard: false; Vorlage: true)
webdesk.luceneIndexing	Ob der Lucene-Suchindex beim Start aktualisiert wird. Der Index liegt im Cache-Verzeichnis der Anwendung unter lucene/<Unterverzeichnis>.	true, false (Standard: true)

	Das Unterverzeichnis ist standardmäßig eine Zufallszahl, kann aber mit dem JVM-Parameter -Dwebdesk.lucene.index.subdir=<name> fest vorgegeben werden (z. B. wenn mehrere Instanzen denselben Cache-Pfad nutzen).	
webdesk.localLogConfig	Bei true ignoriert der Knoten Aufforderungen, die Log-Konfiguration (log4j2.xml) aus der Datenbank neu zu laden. Änderungen in der Aktion po_editLogSettings („Log-Einstellungen“) auf diesem Knoten werden dann nicht in die Datenbank gespeichert und nicht an andere Cluster-Knoten verteilt. Typischer Einsatz: ein Debug-Knoten.	true, false (Standard: false)
webdesk.collectorHostname.regex	Regulärer Ausdruck für die Hostnamen, die den JavaMelody-Collector verwenden dürfen. Wenn gesetzt und der Tomcat-Filter RemoteHostFilter verfügbar ist, konfiguriert Webdesk den Filter beim Start mit diesem Ausdruck.	regulärer Ausdruck (Standard: nicht gesetzt)

TA-Konnektor (IF6020)

Eigenschaft	Erklärung	mögliche Wert(e)
webdesk.ta.if6020impl.rmi	RMI-Adresse des TA-Konnektors (Broker) zur Zeiterfassungs-Anbindung IF6020. Stehen mehrere Konnektoren zur Verfügung, werden die Adressen durch Strichpunkt getrennt. Fehlt der Eintrag, wird er als leer angenommen.	rmi://<host>:<port>/taRmiBroker, mehrere mit „;“ getrennt (Vorlage: rmi://localhost:4009/taRmiBroker)
webdesk.ta.if6020impl.rmi.timeoutMillis	Timeout in Millisekunden für den Verbindungs-Handshake zum Konnektor.	Zahl (Standard: 2000)
webdesk.ta.if6020impl.rmi.connectTimeoutMillis	Timeout in Millisekunden für den TCP-Verbindungsaufbau zum Konnektor.	Zahl (Standard: 3000)

Beispiel mit zwei Konnektoren:

```
webdesk.ta.if6020impl.rmi = rmi://<ta-host1>:4009/taRmiBroker;rmi://<ta-host2>:4012/
taRmiBroker
```

Eigenschaften von Beans überschreiben

Jeder Eintrag der Form <beanId>.<eigenschaft> = wert setzt beim Start die Eigenschaft der Spring-Bean mit dieser Id (Bean-Namen und Eigenschaften entsprechen den Namen in der Maske „Systemparameter“, siehe Systemparameter Portal & Organisation[4]). Das ist ein Spring-„PropertyOverrideConfigurer“, der webdesk.properties liest.

Beispiele:

```
PoOptions.clusterNode = node1
PoHelpOptions.serviceDeskEmail = <service-desk-adresse>
PoHelpOptions.errorReportEnabled = true
```

Achtung: Einträge mit unbekannter Bean-Id oder unbekannter Eigenschaft werden ohne Fehlermeldung ignoriert. Ein Tippfehler bleibt daher unbemerkt; prüfen Sie nach dem Neustart in der Anwendung, ob der Wert wirksam ist. Per Umgebungsvariable oder JVM-Parameter können nur die in der Tabelle unten genannten Schlüssel übersteuert werden, darunter PoOptions.clusterNode, WfStartup.engineName und SecurityHeadersOptions.forceSameSiteNoneCookieAttribute; alle anderen Bean-Einträge müssen in der Datei stehen.

JVM-Parameter und Umgebungsvariablen

Bestimmte Schlüssel können unabhängig von der Datei übersteuert werden. Für diese Schlüssel gilt die Rangfolge (höchste zuerst):

1. Umgebungsvariable WD_...
2. JVM-Parameter -Dschlüssel=wert (z. B. in CATALINA_OPTS / JAVA_OPTS)
3. Wert in webdesk.properties

Namensregel

Der Name der Umgebungsvariable entsteht aus dem Schlüsselnamen so: alle Buchstaben in Großbuchstaben umwandeln, jeden Punkt durch einen Unterstrich ersetzen und WD_ voranstellen. Andere Zeichen (z. B. Unterstriche) bleiben unverändert. Beispiele:

- webdesk.connection.driver_class wird zu WD_WEBDESK_CONNECTION_DRIVER_CLASS
- webdesk.ta.if6020impl.rmi wird zu WD_WEBDESK_TA_IF6020IMPL_RMI
- PoOptions.clusterNode wird zu WD_POOPTIONS_CLUSTERNODE

Eine leere oder nur aus Leerzeichen bestehende Umgebungsvariable wird ignoriert (der Wert aus der Datei gilt dann weiter). Nicht in der Liste enthaltene Schlüssel können nicht per Umgebungsvariable gesetzt werden, auch wenn eine passend benannte Variable existiert.

Überschreibbare Schlüssel

Schlüssel	Umgebungsvariable	Bemerkung
webdesk.connection.driver_class	WD_WEBDESK_CONNECTION_DRIVER_CLASS	Webdesk-Datenbank
webdesk.connection.url	WD_WEBDESK_CONNECTION_URL	Webdesk-Datenbank
webdesk.connection.username	WD_WEBDESK_CONNECTION_USERNAME	Webdesk-Datenbank
webdesk.connection.password	WD_WEBDESK_CONNECTION_PASSWORD	Webdesk-Datenbank
webdesk.dialect	WD_WEBDESK_DIALECT	Hibernate-Dialekt
shark.connection.driver_class	WD_SHARK_CONNECTION_DRIVER_CLASS	Shark-Datenbank
shark.connection.url	WD_SHARK_CONNECTION_URL	Shark-Datenbank

shark.connection.username	WD_SHARK_CONNECTION_USERNAME	Shark-Datenbank
shark.connection.password	WD_SHARK_CONNECTION_PASSWORD	Shark-Datenbank
webdesk.ta.if6020impl.rmi	WD_WEBDESK_TA_IF6020IMPL_RMI	TA-Konnektor(en), mehrere mit „;“
webdesk.runRegistration	WD_WEBDESK_RUNREGISTRATION	Registrierung beim Start
webdesk.disableTimeLineChecks	WD_WEBDESK_DISABLETIMELINECHECKS	Zeitleistenprüfung
webdesk.useEmbeddedDatabase	WD_WEBDESK_USEEMBEDDEDDATABASE	eingebettete Datenbank
webdesk.exitJvmOnConnectionError	WD_WEBDESK_EXITJVMONCONNECTIONERROR	siehe auf den Liste, die Prüfung liest den Wert jedoch aus webdesk.properties; dort setzen
PoOptions.clusterNode	WD_POOPTIONS_CLUSTERNODE	Knotenname (Cluster)
WfStartup.engineName	WD_WFSTARTUP_ENGINENAME	Name der Workflow-Engine (Standard: webdeskshark)
SecurityHeadersOptions.forceSameSiteCookieAttribute	WD_SECURITYHEADERSOPTIONS_FORCESAMESITECOOKIEATTRIBUTE	SameSite-Attribut erzwingen
isDistributed	WD_ISDISTRIBUTED	Cluster, siehe unten
useJmsReplication	WD_USEJMSREPLICATION	Cluster, siehe unten
webdesk.cacheReplication.RMI.peerProvider.properties	WD_WEBDESK_CACHEREPLICATION_RMI_PEERPROVIDER_PROPERTIES	Cluster, siehe unten
webdesk.cacheReplication.JMS.peerProvider.properties	WD_WEBDESK_CACHEREPLICATION_JMS_PEERPROVIDER_PROPERTIES	Cluster, siehe unten
redis.useAsSessionStore redis.connection.url redis.connection.password redis.clusterNodes redis.session.cookie.jvmRoute redis.credentials.encrpted	WD_REDIS_USEASSESSIONSTORE WD_REDIS_CONNECTION_URL WD_REDIS_CONNECTION_PASSWORD WD_REDIS_CLUSTERNODES WD_REDIS_SESSION_COOKIE_JVMROUTE WD_REDIS_CREDENTIALS_ENCRPTED	Redis als Session-Speicher, siehe unten. Der Schlüssel redis.credentials.encrpted heißt tatsächlich so (ohne „y“).

Beispiel für den Container-Betrieb

In der Datei webdesk.properties im Container genügt ein Eintrag, der festlegt, dass die über die Umgebung gelieferten Zugangsdaten nicht verschlüsselt sind (der Eintrag lässt sich nicht per Umgebungsvariable setzen). Ohne diesen Eintrag versucht Webdesk beim Start, die (fehlenden) Werte der Datei zu verschlüsseln, und bricht mit der Meldung ab, dass webdesk.connection.username in webdesk.properties keinen Wert hat:

```
webdesk.credentials.encrypted = false
webdesk.runRegistration = 1
```

Die Umgebungsvariablen (hier als Liste, wie in einer Docker-Compose-Datei):

```
WD_WEBDESK_CONNECTION_DRIVER_CLASS=org.mariadb.jdbc.Driver
WD_WEBDESK_CONNECTION_URL=jdbc:mysql://<db-host>/webdesk?
permitMysqlScheme&allowMultiQueries=true
WD_WEBDESK_CONNECTION_USERNAME=<db-user>
WD_WEBDESK_CONNECTION_PASSWORD=<db-password>
WD_WEBDESK_DIALECT=org.hibernate.dialect.MySQLDialect
WD_SHARK_CONNECTION_DRIVER_CLASS=org.mariadb.jdbc.Driver
WD_SHARK_CONNECTION_URL=jdbc:mysql://<db-host>/shark?
permitMysqlScheme&allowMultiQueries=true
WD_SHARK_CONNECTION_USERNAME=<db-user>
WD_SHARK_CONNECTION_PASSWORD=<db-password>
WD_WEBDESK_TA_IF6020IMPL_RMI=rmi://<ta-host>:4009/taRmiBroker
WD_WEBDESK_EXITJVMONCONNECTIONERROR=true
```

Hinweis: Möchten Sie im Container verschlüsselte Zugangsdaten verwenden, setzen Sie webdesk.credentials.encrypted = true in der Datei und liefern in den Variablen die mit dem Systemschlüssel verschlüsselten Base64-Werte. Da WD_WEBDESK_EXITJVMONCONNECTIONERROR die Prüfung nicht erreicht (siehe Tabelle), tragen Sie webdesk.exitJvmOnConnectionError = true in der Datei ein, wenn der Container bei Verbindungsfehlern beendet werden soll.

Cluster und Redis (Überblick)

Für den Betrieb mit mehreren Knoten kennt webdesk.properties unter anderem folgende Schlüssel; Bedeutung und Zusammenspiel siehe Seite „Cluster- und Cache-Betrieb“.

Eigenschaft	Erklärung	mögliche Wert(e)
isDistributed	Schaltet den Cluster-Betrieb (Cache-Replikation) ein.	true, false (Standard: false)
useJmsReplication	Cache-Replikation über JMS statt RMI.	true, false
PoOptions.clusterNode	Name des aktuellen Knotens. Ist isDistributed per Umgebung/ JVM auf true gesetzt und kein Knotenname angegeben, verwendet Webdesk den Wert der Umgebungsvariable HOSTNAME. Im Cluster hängt Webdesk den Knotennamen an den Namen der Workflow-Engine an (webdeskshark_<knoten>), solange WfStartup.engineName nicht geändert wurde.	Text, z. B. node1
webdesk.cacheReplication.RMI.peerProvider.properties webdesk.cacheReplication.peerListener.properties webdesk.cacheReplication.JMS.peerProvider.properties	Parameter der Cache-Replikation (RMI bzw. JMS).	Beispiele siehe Vorlage
redis.*	Redis als Session-Speicher (redis.useAsSessionStore, redis.connection.url, redis.connection.password, redis.clusterNodes, redis.session.cookie.jvmRoute).	siehe Seite „Cluster- und Cache-Betrieb“

Fehlersuche

- **Webdesk startet nicht, „Error while trying to load webdesk.properties“:** Die Datei fehlt im Klassenpfad (WEB-INF/classes).
- **Start bricht ab mit „... in webdesk.properties has no value“:** Die Zugangsdaten stehen nicht in der Datei, webdesk.credentials.encrypted fehlt jedoch (siehe Container-Beispiel).
- **Entschlüsselung schlägt fehl bzw. System startet nicht nach Wechsel der Datei oder des Servers:** Schlüssel (enc.key bzw. webdesk.encryption.key) stimmt nicht mit dem im System verwendeten Schlüssel überein.
- **Eine Umgebungsvariable zeigt keine Wirkung:** Schlüssel in der Tabelle „Überschreibbare Schlüssel“ prüfen, Namensregel prüfen (Punkte werden zu Unterstrichen, Großbuchstaben), Variable darf nicht leer sein.

- **Änderung in der Datei wirkt nicht oder ist nach dem Start verschwunden:** Die Datei wird nur beim Start gelesen; Webdesk schreibt sie dabei teilweise neu (siehe „Automatische Anpassungen beim Start“) und setzt webdesk.runRegistration = 1 nach der Registrierung auf 0.
- **Start ist sehr langsam:** PoStartup.executeIndexManager, webdesk.disableTimeLineChecks, webdesk-sessionFactory.schemaUpdate, webdesk.runRegistration und webdesk.luceneIndexing bestimmen, welche Prüfungen beim Start laufen.

1. /daisy/personalwolke-admin/5655-dsy.html
2. /daisy/personalwolke-admin/10823-dsy.html
3. /daisy/personalwolke-admin/10823-dsy.html
4. /daisy/personalwolke-admin/5655-dsy.html