

# Erstinstallation

- Datenbank MySQL
  - Anlegen von Datenbank-Schema und -Benutzer
  - Anlegen der Quartz-Tabellen (Job-Scheduling)
  - Import der vorbereiteten Datenbank
  - Anpassen context.xml des Application Server für MySQL
- Alternative zu MySQL: Datenbank Oracle
  - Import des DB-Dump
  - Optional: Ersetzen des "initialuser" Benutzers
  - Anpassen context.xml des Application Server für Oracle
- Alternative zu MySQL: Datenbank MS SQL Server
  - Import des DB-Dump
  - Anpassen context.xml des Application Server für MS SQL Server
- Installation der Software auf dem Application-Server
  - Installation des war-file
  - Nach der Installation
    - Systemparameter und Mailserver-Anbindung
    - Single Sign-On
    - HTTPS
    - Optional: Active Directory Synchronisierung

## Datenbank MySQL

Für die Schritte in diesem Abschnitt schlagen wir vor, die MySQL Workbench zu verwenden (das Root-Passwort zum Einloggen haben Sie bei [Aufsetzen der System-Umgebung](#) Datenbank MySQL während des Punkt 5 vergeben). Sie können aber auch direkt die MySQL Kommandozeile verwenden, wenn Ihnen das lieber ist.

### Anlegen von Datenbank-Schema und -Benutzer

Führen Sie das folgende SQL-Skript mit dem Datenbank-Benutzer root aus (z.B. via MySQL Workbench):

```
-- create schema goco;
create schema goco;
create user goco identified by 'goco';
REVOKE ALL PRIVILEGES,GRANT OPTION from goco;
GRANT SELECT, INSERT, UPDATE, DELETE, EXECUTE, SHOW VIEW, CREATE, CREATE VIEW, ALTER, INDEX,
DROP, REFERENCES ON goco.* TO 'goco';

-- create schema quartz
create schema quartz;
create user quartz identified by 'quartz';
REVOKE ALL PRIVILEGES,GRANT OPTION from quartz;
GRANT SELECT, INSERT, UPDATE, DELETE ON quartz.* TO 'quartz';
```

Belassen Sie die Passwörter für die Benutzer goco und quartz vorerst, bis die Erstinstallation durchgeführt und verifiziert ist. Danach können die Passwörter geändert werden wie beschrieben unter [Wie kann ich das Datenbank-Passwort ändern?](#)

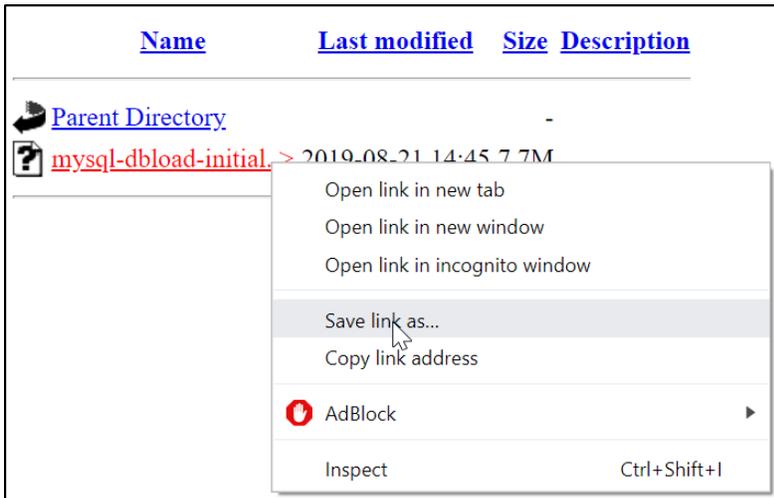
### Anlegen der Quartz-Tabellen (Job-Scheduling)

Führen Sie das folgende Script aus (dies erstellt Tabellen im Schema "quartz", welche für das nächtliche Batchjob-Scheduling zuständig sind):

[create\\_quartz\\_tables.sql](#)

### Import der vorbereiteten Datenbank

1. Laden Sie den initialen DB-Load herunter (rechte Maustaste Save Link As...), den wir Ihnen als Self-Contained File in der Download-Area bereitgestellt haben (mysql-dbload-initial.sql).



2. Importieren Sie dies in MySQL (z.B. via MySQL Workbench, Menü Server --> Import --> Import From Self-Contained File). Wählen Sie als Default Target Schema "goco" aus.
3. Der Import-Screen der MySQL Workbench zeigt nach einigen Sekunden einen grünen Balken zum Zeichen, dass der Import erfolgreich war. Zur Überprüfung aktualisieren Sie auf der linken Sidebar die Schemas, öffnen das Schema "goco", und überzeugen sich, dass es Tabellen enthält (Prefix co\_).

## Anpassen context.xml des Application Server für MySQL

Um auf dem Application Server die Datenbank-Connection zu konfigurieren, öffnen Sie das File context.xml (in <Tomcat-Directory>/conf). Ersetzen Sie den Inhalt mit dem folgendem Codeblock. Dies erstellt die Datenbank Data-Sources für die Schemas "goco" und "quartz":

### context.xml für MySQL

```
<Context>
  <Resource
    name="gocoTenantDs"
    factory="com.gocompliant.encryptedDs.EncryptedDataSourceFactory"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/goco?allowPublicKeyRetrieval=true&useSSL=false"
    username="goco"
    password="922f591c101ab822305be286a532c196"
    maxActive="20"
    minIdle="0"
    maxIdle="0"
    minEvictableIdleTimeMillis="14400000"
    maxWait="-1" />
  <Resource
    name="quartz"
    factory="com.gocompliant.encryptedDs.EncryptedDataSourceFactory"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/quartz?allowPublicKeyRetrieval=true&useSSL=false"
    username="quartz"
    password="2259b33c5588618ee0742b2ced9998e6"
    maxActive="20"
    minIdle="0"
    maxIdle="0"
    minEvictableIdleTimeMillis="14400000"
    maxWait="-1" />

  <Manager pathname="" />
</Context>
```

# Alternative zu MySQL: Datenbank Oracle

## Import des DB-Dump

Wir werden Ihnen in der Download-Area einen DB-Dump zur Verfügung stellen, den Sie via impdp importieren können. Der Dump wird bereits alle benötigten Schemas enthalten.

## Optional: Ersetzen des "initialuser" Benutzers

Im Normalfall liefern wir Ihnen schon initial einige vorkonfigurierte Benutzer mit, deren Benutzer-IDs auf ihre Windows Benutzer-IDs passen, sodass Sie sich sofort mit Single Sign-On via Active Directory einloggen können. Falls dies nicht der Fall ist (wird in der Auslieferungs-Mail explizit erwähnt), müssen Sie die "initialuser" Benutzer-ID mit Ihrer Windows Benutzer-ID ersetzen. Ersetzen Sie im folgenden Script YOUR\_WINDOWS\_USER\_ID mit Ihrer Windows Benutzer-ID und führen Sie es dann aus:

```
update goco.co_employee set user_id = 'YOUR_WINDOWS_USER_ID' where user_id = 'initialuser';
commit;
```

## Anpassen context.xml des Application Server für Oracle

Um auf dem Application Server die Datenbank-Connections zu konfigurieren, öffnen Sie das File context.xml (in <Tomcat-Directory>/conf). Ersetzen Sie den Inhalt mit folgendem Codeblock.

WICHTIG: Ersetzen Sie YOUR\_SID beide Male durch die von Ihnen vorgesehene Oracle SID.

### context.xml für Oracle

```
<Context>
  <Resource
    name="gocoTenantDs"
    factory="com.gocompliant.encryptedDs.EncryptedDataSourceFactory"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@localhost:1521/YOUR_SID"
    username="goco"
    password="922f591c101ab822305be286a532c196"
    maxActive="20"
    minIdle="0"
    maxIdle="0"
    minEvictableIdleTimeMillis="14400000"
    maxWait="-1" />
  <Resource
    name="quartz"
    factory="com.gocompliant.encryptedDs.EncryptedDataSourceFactory"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@localhost:1521/YOUR_SID"
    username="quartz"
    password="2259b33c5588618ee0742b2ced9998e6"
    maxActive="20"
    minIdle="0"
    maxIdle="0"
    minEvictableIdleTimeMillis="14400000"
    maxWait="-1" />

  <Manager pathname="" />
</Context>
```

# Alternative zu MySQL: Datenbank MS SQL Server

## Import des DB-Dump

Wir werden Ihnen in der Download-Area einen DB-Dump zur Verfügung stellen, den Sie als bacpac File importieren können. Der Dump wird bereits alle benötigten Schemas enthalten.

## Anpassen context.xml des Application Server für MS SQL Server

Um auf dem Application Server die Datenbank-Connections zu konfigurieren, öffnen Sie das File context.xml (in <Tomcat-Directory>/conf). Ersetzen Sie den Inhalt mit folgendem Codeblock.

WICHTIG - ersetzen Sie beide Male die folgenden Platzhalter:

- DB\_USER\_ID
- DB\_PASSWORD\_ENCRYPTED
- DB\_SERVER
- DB\_NAME

Kontrollieren Sie auch, ob Sie im url-Attribute andere Einstellungen vornehmen möchten.

### context.xml für Oracle

```
<Context>
  <Resource
    name="goco"
    factory="com.gocompliant.encryptedDs.EncryptedDataSourceFactory"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
    url="jdbc:sqlserver://DB_SERVER;databaseName=DB_NAME;encrypt=true;trustServerCertificate=true"
    username="DB_USER_ID"
    password="DB_PASSWORD_ENCRYPTED"
    initialSize="1"
    maxActive="20"
    minIdle="0"
    maxIdle="0"
    minEvictableIdleTimeMillis="110000"
    maxWait="-1"/>
  <Resource
    name="quartz"
    factory="com.gocompliant.encryptedDs.EncryptedDataSourceFactory"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
    url="jdbc:sqlserver://DB_SERVER;databaseName=DB_NAME;encrypt=true;trustServerCertificate=true"
    username="DB_USER_ID"
    password="DB_PASSWORD_ENCRYPTED"
    initialSize="1"
    maxActive="20"
    minIdle="0"
    maxIdle="0"
    minEvictableIdleTimeMillis="110000"
    maxWait="-1"/>
  <Manager pathname="" />
</Context>
```

## Installation der Software auf dem Application-Server

### Installation des war-file

1. Stoppen Sie den Tomcat-Service.
2. Löschen Sie das Verzeichnis <Tomcat-Directory>/webapps/ROOT.
3. Laden Sie das war-file mittels des Links, den wir Ihnen per Email geschickt haben, aus der Download-Area herunter (rechte Maustaste Save Link As). Eventuell ändert der Browser die Endung des File beim Herunterladen nach .zip, in dem Fall ändern Sie die Endung wieder zurück auf .war. Kopieren Sie das war-file nach <Tomcat-Directory>/webapps, also z.B. C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps
4. Benennen Sie das war-file um nach ROOT.war. Es ist wichtig, dass ROOT komplett in Grossbuchstaben geschrieben ist.
5. Starten Sie den Tomcat-Service neu
6. Öffnen Sie mit einem Browser <http://localhost> oder <http://localhost:8080> und verifizieren Sie, das die GoCompliant Application läuft. Wenn Sie eine SSO Login-Variante gewählt haben, sollten Sie automatisch eingeloggt werden (dazu muss Ihr Windows Benutzername mit einem der auf der Datenbank vorkonfigurierten Benutzern übereinstimmen). Ansonsten sollten Sie eine Login-Seite sehen.

## Nach der Installation

### Systemparameter und Mailserver-Anbindung

Bevor die Applikation für die Fachabteilungen freigegeben wird, sollten die wichtigsten Systemparameter konfiguriert werden, insbesondere diejenigen für die Mailserver-Anbindung. Eine Beschreibung dieser Parameter findet sich im Abschnitt [Systemparameter](#).

### Single Sign-On

Wir empfehlen, unsere Unterstützung für das Single Sign-On via Kerberos-Ticket zu verwenden. Die Applikation erkennt automatisch, dass sie in einer On-Premises Umgebung läuft, und wird ohne explizite weitere Konfiguration diese Art der Authentifizierung versuchen zu verwenden. Damit das funktioniert, ist es nötig, dass ihre Benutzer-IDs den Windows-Benutzernamen entsprechen. Falls Sie von SaaS auf On-Premises wechseln, ist es daher eventuell nötig die Benutzer-IDs anzupassen. Bei Probleme mit dem Single Sign-On verweisen wir auf folgende Hilfeseite: [On-Premises: Single Sign-On \(SSO\) funktioniert nicht](#).

### HTTPS

Wenn Sie die Applikation über https laufen lassen wollen und über ein entsprechendes Zertifikat verfügen, folgen Sie der Anleitung: [Wie kann ich HTTPS einrichten?](#)

### Optional: Active Directory Synchronisierung

Wenn Sie die LDAP (Active Directory) Synchronisierung verwenden möchten, findet sich eine Beschreibung der entsprechenden Parameter im Abschnitt [LDAP \(Active Directory\) Systemparameter](#). Wir empfehlen eine Active Directory Anbindung erst ab einer Benutzerzahl von grösser 50. Darunter lohnt sicher der Aufwand erfahrungsgemäss nicht bzw. ist es einfacher, die Benutzer manuell zu verwalten.



#### Verwandte Seiten

- [Systemvoraussetzungen](#)
- [Systemparameter](#)
- [Informations- und Eskalations-E-mails im IA](#)
- [Informations- und Eskalations-E-mails im IKS](#)
- [Installation eines Updates](#)