



dashBIRD Studio 2.2

Administrationshandbuch

Dokumentversion vom 20. 11. 2023

Inhalt

1	Über diese Dokumentation	4
1.1	Änderungshistorie.....	4
1.2	Zielgruppe und Voraussetzungen	4
1.3	Hinweise und Konventionen	4
2	Hard- und Softwarevoraussetzungen	6
2.1	Server.....	6
2.1.1	Hardware.....	6
2.1.2	Software	6
2.2	Client.....	6
2.2.1	Hardware.....	6
2.2.2	Software	7
3	Vorbereitung.....	8
3.1	servBIRD	8
3.2	Heimverzeichnis einrichten.....	8
3.3	Datenbank Schema erstellen	8
4	Installation	9
4.1	Heimverzeichnis einrichten.....	9
4.2	Datenbankschema anlegen.....	9
4.3	WildFly Application Server.....	9
4.4	TRADUI dashBIRD Studio.....	10
4.4.1	Back-End.....	10
4.4.2	Front-End.....	10
4.5	WildFly Datenbankkonfiguration	10
4.6	Datenbank Dialekt und Datenbankschema Name.....	12
4.7	MariaDB	13
4.8	Weitere WildFly Konfigurationsschritte	13
4.9	Weitere TRADUI dashBIRD Studio Konfigurationsschritte.....	15
4.10	Startparameter	17
4.11	Starten des Servers.....	17
4.12	Nachbearbeitung des Datenbankschemas.....	18

4.13 Erste Anmeldung.....	19
4.14 servBIRD Integration.....	19
5 Einstellungen.....	20
5.1 Die folgenden Einstellungen sollten gesetzt sein.....	21
5.2 Die folgenden Einstellungen können gesetzt werden.....	21

1 Über diese Dokumentation

1.1 Änderungshistorie

Datum	Version	Änderungsbeschreibung	Änderung von
08.04.2021	v0.1	Initialer Stand der dashBIRD Studio 1.0 Dokumentation	Team TRADUI
21.07.2021	v0.2	Anpassung an dashBIRD Studio 1.0.3	Team TRADUI
10.02.2022	v0.3	Anpassung an dashBIRD Studio 1.1.0	Team TRADUI
15.02.2023	v0.4	Anpassung an dashBIRD Studio 2.0.0	Team TRADUI
26.09.2023	v0.5	Anpassung an dashBIRD Studio 2.2.0	Team TRADUI

1.2 Zielgruppe und Voraussetzungen

Die vollständige **dashBIRD Studio** Dokumentation ist in zwei Handbücher aufgeteilt:

1. Das **Benutzerhandbuch** ...ist für den Endanwender gedacht.
2. Das **Administrationshandbuch** ... befasst sich mit der Installation, der Einrichtung, der Wartung und dem Betrieb des **dashBIRD Studio**.

1.3 Hinweise und Konventionen

Tipp

Dieser Block hebt Informationen hervor, die bspw. zeit- oder ressourcenschonende Verfahren oder Best Practices erläutern.

Information

In diesem Block werden Informationen von besonderer Bedeutung oder besonderem Interesse hervorgehoben.

Hinweis

Dieser Block weist auf zu beachtende Informationen hin oder warnt vor Stolperfallen. Bitte lesen Sie diese Blöcke aufmerksam!

Achtung

Diese Hinweise deuten auf Informationen hin, die bei Missachtung oder falscher Nutzung zu Fehlfunktion der Software oder Löschung von relevanten Daten führen können.

Beispiel

In diesem Block wird ein Beispiel zum jeweiligen Kontext aufgeführt.

2 Hard- und Softwarevoraussetzungen

2.1 Server

Zum Betrieb eines **dashBIRD Studio** Servers müssen folgende minimale Anforderungen erfüllt sein:

2.1.1 Hardware

CPU	2 GHz Dual Core oder besser
Arbeitsspeicher	8 GB
Festplattenspeicher	20 GB
Netzwerk	100 Mbit/s Ethernetanbindung

Hinweis

Es handelt sich hierbei um Minimalanforderungen, die zum Betrieb nötig sind. Je nach Anwendungsszenario sollten die zur Verfügung gestellten Ressourcen des Systems gegebenenfalls größer dimensioniert werden.

2.1.2 Software

Betriebssystem	Jedes Betriebssystem mit Java 8 Unterstützung	
Java	JDK 11	https://adoptium.net/de/temurin/releases/?version=11 http://oracle.com/java
Application Server	Wildfly 27.0.1	http://wildfly.org/downloads/
Datenbank	MySQL 8, Oracle	
servBIRD	das aktuelle servBIRD Release	https://kundenportal.tradui.net/

2.2 Client

Ein Clientsystem, dass auf die Dienste von **dashBIRD Studio** zugreift, sollte folgende Voraussetzungen erfüllen:

2.2.1 Hardware

CPU	1,5 GHz Single Core oder besser
Arbeitsspeicher	2 GB

Festplattenspeicher	beliebig
Netzwerk	beliebige Netzwerkanbindung
Bildschirmauflösung	1600 x 900 px

2.2.2 Software

Betriebssystem	beliebig
Internetbrowser	aktuelle Version von Mozilla Firefox, Google Chrome oder Microsoft Edge

3 Vorbereitung

3.1 servBIRD

Voraussetzung zum Betrieb des **dashBIRD Studio** ist eine vorhandene **servBIRD** Installation (ab Version 3.18.0) inklusive **TRADUI mooBIRD**. **servBIRD** dient als Authentifizierungsserver und erlaubt das zuweisen von Rollen und Berechtigungen für **dashBIRD Studio** Nutzer.

Den Nutzern, die das **dashBIRD Studio** verwenden sollen, müssen in **servBIRD** über Ihre Rollen Lizenzen für das **dashBIRD Studio**, sowie die gewünschten Berechtigungen zugewiesen werden. Einen Überblick über die verfügbaren Berechtigungen bietet folgende Tabelle:

dashBIRD Studio Core	Öffnen und Verwenden von dashBIRD-Studio
dashBIRD Studio Admin	dashBIRD-Studio Einstellungen einsehen und editieren
dashBIRD Studio Connection Admin	Erstellung und voller Zugriff auf alle Datenanbindungen
dashBIRD Studio Connection Create	Erstellung und voller Zugriff auf eigene Datenanbindungen
dashBIRD Studio Dataset Admin	Erstellung und voller Zugriff auf alle Datensets
dashBIRD Studio Dataset Create	Erstellung und voller Zugriff auf eigene Datensets

3.2 Heimverzeichnis einrichten

Zur Ablage von Konfigurationsdateien und Datenbanktreibern dient das sogenannte **dashBIRD Studio** Heimverzeichnis, welches ein Verzeichnis auf einem Dateisystem darstellt.

Dieses Verzeichnis muss auf einem lokalen Laufwerk erstellt werden. Der Benutzer, der den WildFly Dienst ausführt benötigt Lese- und Schreibrechte auf dieses Verzeichnis.

3.3 Datenbank Schema erstellen

Zu Verwaltung der **dashBIRD Studio** Metadaten wird ein Datenbank Schema benötigt. Die Struktur des Schemas muss vorab per Skript erstellt werden. Das hierzu benötigte Skript wird mit Bereitstellung des Installationspakets ausgeliefert. Welches Datenbanksystem geeignet ist, entnehmen Sie bitte dem Abschnitt **Hard- und Softwarevoraussetzungen**. Weitere Informationen zum Thema Datenbanken entnehmen Sie bitte dem Abschnitt **Installation**.

4 Installation

4.1 Heimverzeichnis einrichten

Legen Sie ein Verzeichnis auf einem lokalen Dateisystem an, das zur Bereitstellung von Dateien für das **dashBIRD Studio** dient. Der WildFly Application Server benötigt Lese- und Schreibrechte auf dieses Verzeichnis

Erstellen Sie einen Unterordner "jdbc" und legen Sie in diesem Verzeichnis die gewünschten JDBC Treiber zur Verwendung durch das **dashBIRD Studio** ab.

Erstellen Sie einen Unterordner "config" legen Sie in diesem Verzeichnis das benötigte Lizenzfile ab.

Beispiel: Windows

```
C:\Daten\dashbirdstudio\  
C:\Daten\dashbirdstudio\jdbc\  
C:\Daten\dashbirdstudio\config\
```

Beispiel: GNU/Linux

```
/var/tradui/dashbirdstudio/  
/var/tradui/dashbirdstudio/jdbc/  
/var/tradui/dashbirdstudio/config/
```

4.2 Datenbankschema anlegen

Installieren Sie ein Datenbankverwaltungssystem (MySQL 5.x bzw. MariaDB, Oracle 10g oder höher). Zur Installation beachten Sie bitte die Angaben des jeweiligen Herstellers.

Erstellen sie ein leeres Datenbankschema und einen neuen Benutzer, der Lese- und Schreibrechte für dieses Schema besitzt.

Hinweis

Bitte beachten Sie, dass das Schema zwischen Groß- und Kleinschreibung unterscheiden muss (case sensitive). Falls Sie beispielsweise eine MySQL Datenbank benutzen, wählen Sie für das Schema bitte als Standardkollation *utf8_bin*.

4.3 WildFly Application Server

Wir empfehlen zur Installation von **dashBIRD Studio** den WildFly Application Server in Version *27.0.1.Final*.

Hinweis

Sie können für das **dashBIRD Studio** selbstverständlich auch die gleiche WildFly Application Server Installation verwenden, auf der **servBIRD** bereitgestellt wird. In diesem Fall brauchen Sie keine zusätzliche Installation von WildFly durchführen.

Das WildFly Application Server Archiv kann in einen Ordner Ihrer Wahl extrahiert werden. **dashBIRD Studio** verwendet die Jakarta EE 10 Spezifikation. Application Server, die nicht der Jakarta EE 10 Spezifikation entsprechen, sind für den Betrieb des **dashBIRD Studio** nicht geeignet.

Als Installationsverzeichnis empfehlen wir:

Windows	Linux etc.
C:\Daten\dashBIRDstudio\wildfly-x.x	/opt/dashbirdstudio/wildfly-x.x

Sie können WildFly prinzipiell in jedem beliebigen Verzeichnis installieren.

Download: <http://wildfly.org/downloads/>

4.4 TRADUI dashBIRD Studio

4.4.1 Back-End

Das WAR-File (DashletEngine_RestService-2.2.x.war) stellt das Back-End des **dashBIRD Studio** und enthält die für den Betrieb des Front-Ends benötigten Schnittstellen. Das WAR-File muss in das Verzeichnis "*standalone/deployments*" von WildFly kopiert werden.

4.4.2 Front-End

Das Front-End besteht aus den für das **dashBIRD Studio** Portal benötigten Ressourcen und Komponenten. Zur Installation kopieren Sie bitte das bereitgestellte Verzeichnis mit dem Namen "*webroot*" in das Hauptverzeichnis der WildFly Installation, sodass es auf gleicher Ebene liegt, wie beispielsweise die Verzeichnisse "*bin*" und "*standalone*".

4.5 WildFly Datenbankkonfiguration

Die Datenbankkonfiguration wird in der zentralen Konfigurationsdatei von WildFly vorgenommen.

Um die Datenbankverbindung zu konfigurieren, öffnen Sie im Verzeichnis "*WILDFLY_HOME/standalone/configuration*" die Datei "*standalone.xml*" und suchen folgenden Eintrag:

standalone.xml

```
<subsystem xmlns="urn:jboss:domain:datasources:7.0">
  <datasources>
    ...
  </datasources>
</subsystem>
```

Innerhalb des Tags `<datasources>` tragen sie Ihre Datenbankverbindung nach folgendem Schema ein:

Beispielkonfiguration: Maria DB

```
<datasource jta="false" jndi-name="java:/datasources/dashbirdstudio" pool-
name="dashbirdstudio" enabled="true" use-ccm="true">
  <connection-url>jdbc:mysql://HOST:PORT/SCHEMANAME</connection-url>
  <driver-class>org.mariadb.jdbc.Driver</driver-class>
  <driver>mariadb-java-client-2.4.0.jar</driver>
  <security>
    <user-name>USERNAME</user-name>
    <password>PASSWORD</password>
  </security>
</datasource>
```

Entfernen sie außerdem, falls vorhanden, die Beispieldatenbank aus dem Tag `<default-bindings>`:

```
<default-bindings context-service="java:jboss/ee/concurrency/context/default"
managed-thread-factory="java:jboss/ee/concurrency/factory/default" managed-
scheduled-executor-service="java:jboss/ee/concurrency/scheduler/default" managed-
executor-service="java:jboss/ee/concurrency/executor/default"/>
```

Beispielkonfiguration: Oracle DB

```
<datasource jta="false" jndi-name="java:/datasources/dashbirdstudio" pool-
name="dashbirdstudio" enabled="true" use-ccm="true">
  <connection-url>jdbc:oracle:thin:@HOST:PORT/SCHEMANAME</connection-url>
  <driver-class>oracle.jdbc.OracleDriver</driver-class>
  <driver>ojdbc6.jar</driver>
  <security>
    <user-name>USERNAME</user-name>
    <password>PASSWORD</password>
  </security>
</datasource>
```

Ersetzen sie den Inhalt folgender Attribute entsprechend Ihrer Umgebung:

- <connection-url>
- <driver-class>
- <driver>
- <user-name>
- <password>

Hinweis

Die gezeigten Beispielkonfigurationen setzen voraus, dass der zu verwendende JDBC Treiber zuvor in das Verzeichnis "*standalone/deployments/*" kopiert wurde.

4.6 Datenbank Dialekt und Datenbankschema Name

dashBIRD Studio verwendet zur Kommunikation mit der Datenbank Hibernate nach der Jakarta EE Spezifikation. Je nach Datenbankhersteller muss der korrekte Hibernate Dialekt definiert sein. Weiterhin muss der Name des Datenbankschemas definiert werden. Um die Einstellungen zu setzen müssen Einträge in der *standalone.xml* eingefügt werden.

Hinweis

Verfügbare Dialekte sind einsehbar unter <https://docs.jboss.org/hibernate/orm/5.1/javadocs/org/hibernate/dialect/package-summary.html>

standalone.xml

```
<server xmlns="urn:jboss:domain:20.0">
  <extensions>
    ...
  <extensions>
  <system-properties>
    <property name="dashbirdstudio.hibernate.dialect" value="org.hibernate.dialect.
MySQL8Dialect"/>
    <property name="dashbirdstudio.hibernate.default_schema" value="SCHEMANAME"/>
  </system-properties>
  <management>
    ...
</server>
```

Weiterhin muss das Ersetzen von Variablen in Java EE Standard Config Files aktiviert werden.

standalone.xml

```
<subsystem xmlns="urn:jboss:domain:ee:6.0">
  <spec-descriptor-property-replacement>true</spec-descriptor-property-replacement>
  ...
</subsystem>
```

4.7 MariaDB

Für Server version: [10.10.2-MariaDB](#)

Client driver: [mariadb-java-client-3.1.2.jar](#)

Dialect: [MariaDBDialect](#)

standalone.xml

```
<system-properties>
  <property name="dashbirdstudio.hibernate.dialect"
value="org.hibernate.dialect.MariaDBDialect"/>
  <property name="dashbirdstudio.hibernate.id.db_structure_naming_strategy"
value="standard"/>
  <property name="dashbirdstudio.hibernate.default_schema" value="SCHEMANAME"/>
  ...
</system-properties>

<datasource jta="false" jndi-name="java:/datasources/dashbirdstudio" pool-
name="dashbirdstudio" enabled="true" use-ccm="true">
  <connection-url>jdbc:mariadb://localhost:3306/SCHEMANAME</connection-url>
  <driver-class>org.mariadb.jdbc.Driver</driver-class>
  <driver>mariadb-java-client-3.1.2.jar</driver>
  ...
</datasource>
```

Nachdem die Tabellen erzeugt wurden, müssen mögliche Spalten mit Typ TinyText zu Typ LongText geändert werden..

4.8 Weitere WildFly Konfigurationsschritte

Der Betrieb des **dashBIRD Studio** Portals erfordert weitere Konfigurationen, welche in der zentralen Konfigurationsdatei von WildFly vorgenommen werden.

Öffnen Sie hierzu im Verzeichnis "*WILDFLY_HOME/standalone/configuration*" die Datei "*standalone.xml*" und suchen folgenden Eintrag:

standalone.xml

```
<subsystem xmlns="urn:jboss:domain:undertow:6.0" default-server="default-server"
default-virtual-host="default-host" default-servlet-container="default" default-security-domain="other">
  ...
  <server name="default-server">
    ...
    <host name="default-host" alias="localhost">
      ...
    </host>
  </server>
</subsystem>
```

Innerhalb des Tags `<host>` ändern Sie den vorhandenen `<location Tag>` und belegen das Attribut `"handler"` mit dem Wert `"webroot"`. Fügen Sie außerdem die aufgeführten `<filter-ref>` Tags hinzu:

standalone.xml

```
<host name="default-host" alias="localhost">
  <location name="/" handler="webroot"/>
  <filter-ref name="dashbirdstudio-redirect" predicate="not equals(%R, '/') and not
equals(%R, '/index.html') and path-prefix('/dashBIRDstudio/') and not path-prefix('/
dashBIRDstudio/assets/') and not path-prefix(/DashletEngine_RestService/) and not
regex('^\/.*.(js|js.map)$') and not regex('^\/.*.(css)$') and not regex('^\/.*.(jpe?
g|png|gif|svg)$') and not regex('^\/.*.(eot|ttf|woff|woff2)$') and regex('^(.*)$')"/>
  <filter-ref name="Access-Control-Allow-Origin"/>
  <filter-ref name="Access-Control-Allow-Methods"/>
  <filter-ref name="Access-Control-Allow-Headers"/>
  <filter-ref name="Access-Control-Allow-Credentials"/>
  <filter-ref name="Access-Control-Max-Age"/>
  <filter-ref name="Cache-Control" predicate="path-prefix('/dashBIRDstudio/')"/>
  <http-invoker http-authentication-factory="application-http-authentication"/>
</host>
```

Suchen Sie nun folgenden Eintrag:

standalone.xml

```
<subsystem xmlns="urn:jboss:domain:undertow:13.0" default-server="default-server"
default-virtual-host="default-host" default-servlet-container="default" default-security-domain="other"
statistics-enabled="{wildfly.undertow.statistics-enabled:$
{wildfly.statistics-enabled:false}}">
  ...
  <handlers>
    ...
  </handlers>
</subsystem>
```

Innerhalb des Tags `<handlers>` ändern Sie den vorhandenen `<file>` Tag (welcome-content) und belegen das Attribut `"name"` mit dem Wert `"webroot"` und das Attribut `"path"` mit dem Wert `"${jboss.home.dir}/webroot"`. Fügen Sie außerdem auf gleicher Ebene des Knotens `<handlers>` einen Knoten `<filters>` und unterhalb dessen, die aufgeführten `<response-header>` Tags hinzu. Ersetzen Sie zudem am Knoten `<response-header>` mit dem Namen `"Access-Control-Allow-Origin"` die Platzhalter für Protokoll, Host und Port des **dashBIRD Studio** im Wert des Attributs `"header-value"` entsprechend Ihrer Konfiguration:

standalone.xml

```
<subsystem xmlns="urn:jboss:domain:undertow:13.0" default-server="default-server"
default-virtual-host="default-host" default-servlet-container="default" default-security-domain="other" statistics-enabled="${wildfly.undertow.statistics-enabled:${wildfly.statistics-enabled:false}}">
    ...
    <handlers>
        <file name="webroot" path="${jboss.home.dir}/webroot"/>
    </handlers>
    <filters>
        <rewrite name="dashbirdstudio-redirect" target="/dashBIRDstudio/index.html"
redirect="false"/>
        <response-header name="Access-Control-Allow-Origin" header-name="Access-
Control-Allow-Origin" header-value="HOST"/>
        <response-header name="Access-Control-Allow-Methods" header-name="Access-
Control-Allow-Methods" header-value="GET, POST, OPTIONS, PUT, DELETE"/>
        <response-header name="Access-Control-Allow-Headers" header-name="Access-
Control-Allow-Headers" header-value="accept, authorization, content-type, x-
requested-with, username, password"/>
        <response-header name="Access-Control-Allow-Credentials" header-name="Access-
Control-Allow-Credentials" header-value="true"/>
        <response-header name="Access-Control-Max-Age" header-name="Access-Control-
Max-Age" header-value="1"/>
        <response-header name="Cache-Control" header-name="Cache-Control" header-
value="max-age=60, must-revalidate"/>
    </filters>
</subsystem
```

4.9 Weitere TRADUI dashBIRD Studio Konfigurationsschritte

Um die Verbindung des Front-Ends mit dem Back-End zu gewährleisten, muss die Adresse des Back-Ends in den Konfigurationsdateien des Front-Ends eingetragen werden.

Öffnen Sie hierzu die Konfigurationsdatei `"WILDFLY_HOME/webroot/config.json"` in einem Texteditor und tragen Sie am Wert des Attributs `"backend_address"` Protokoll, Host und Port entsprechend Ihrer Konfiguration ein, sodass diese auf den WildFly Host mit dem **dashBIRD Studio** Deployment zeigen.

config.json

```
{
  "backend_address": "PROTOCOL://HOST:PORT/DashletEngine_RestService/rest/settings/"
}
```

Alternativ kann die Adresse auch als Header übergeben werden, wenn das Front-End versucht, die Konfigurationsdatei abzurufen. Öffnen sie hierzu im Verzeichnis `"WILDFLY_HOME/standalone/configuration"` die Datei `"standalone.xml"` und suchen folgenden Eintrag:

standalone.xml

```
<subsystem xmlns="urn:jboss:domain:undertow:6.0" default-server="default-server"
default-virtual-host="default-host" default-servlet-container="default" default-
security-domain="other">
  ...
  <server name="default-server">
    ...
    <host name="default-host" alias="localhost">
      ...
    </host>
  </server>
  ...
  <filters>
    ...
  </filters>
</subsystem>
```

Innerhalb des Tags `<host>` fügen Sie *den folgenden* `<filter-ref>` Tag hinzu:

standalone.xml

```
<host name="default-host" alias="localhost">
  ...
  <filter-ref name="Backend-Address" predicate="equals(%R, '/config.json')"/>
</host>
```

Innerhalb des Tags `<filters>` fügen Sie *den folgenden* `<response-header>` Tag hinzu:

standalone.xml

```
<filters>
  <response-header name="Backend-Address" header-name="Backend-Address" header-
value="PROTOCOL://HOST:PORT/DashletEngine_RestService/rest/settings/" />
</filters>
```


4.10 Startparameter

Um die Startparameter des Wildfly Application Servers anzupassen öffnen Sie, je nach Betriebssystem, folgende Datei:

Windows	Linux etc.
WILDFLY_HOME/bin/standalone.conf.bat	WILDFLY_HOME/bin/standalone.conf

Wir empfehlen die folgenden (minimalen) Werte für die Speicherkonfiguration:

Beispiel
<pre>set "JBOSS_JAVA_SIZING=-Xms2048m -Xmx4096m -XX:MetaspaceSize=256M -XX:MaxMetaspaceSize=512m"</pre>

Hinweis

Sollten Sie **dashBIRD Studio** auf dem gleichen WildFly Application Server installieren, auf dem auch **servBIRD** läuft, so berücksichtigen Sie dies bitte bei der Speicherkonfiguration.

Wenn **dashBIRD Studio** auf einem Windows System installiert wird, so ändern Sie bitte das Encoding auf "UTF-8" mittels folgenden zusätzlichen Zeilen:

Beispiel
<pre>rem # Set the JVM encoding explicitly to UTF-8 set "JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8"</pre>

Falls Sie den Pfad zum Java JDK manuell konfigurieren möchten, ändern Sie dazu folgende Abschnitte:

Beispiel
<pre>set "JAVA_HOME=C:\Program Files\Eclipse Adoptium\jdk-11.0.18.10-hotspot" set "JAVA=C:\Program Files\Eclipse Adoptium\jdk-11.0.18.10-hotspot\bin"</pre>

4.11 Starten des Servers

Starten Sie nun den WildFly Application Server. Das **dashBIRD Studio** wird daraufhin das Datenbankschema mit der benötigten Tabellenstruktur füllen.

4.12 Nachbearbeitung des Datenbankschemas

Führen Sie nun das mitgelieferte das mitgelieferte SQL Script "*studio.init.sql*" im angelegten Datenbankschema aus, um die Einstellungstabelle mit vordefinierten Werten zu füllen. Tauschen Sie die Beispielwerte im Anschluss mit den Ihrer Umgebung entsprechenden Werten aus. Eine Übersicht über die vorhandenen Werte finden Sie in nachfolgender Tabelle:

G r o u p	Key	Value	Beschreibung
G l o b a l	BACKEND_AD DRESS	http://HOST:PORT/DashletEngine_RestService/rest/	Protokoll, Host und Port müssen auf den WildFly Host mit dem dashBIRD Studio Deployment zeigen
S y s t e m	STUDIO_HOM E	/PATH/TO/DASHBIRDSTUDIOHOME	Der absolute Pfad zum dashBIRD Studio Heimverzeichnis, der die Unterordner "config" und "jdbc" enthält
S y s t e m	MOOBIRD_AD DRESS	http://HOST:PORT/moobird/service	Protokoll, Host und Port müssen auf den WildFly Host mit dem servBIRD Deployment zeigen
S y s t e m	MOOBIRD_AC CESS_NAME	username	Der Benutzername eines in servBIRD existierenden (technischen) Users für die Verbindung zwischen servBIRD und dashBIRD Studio wenn SAML2 SSO verwendet wird
S y s t e m	MOOBIRD_AC CESS_PASSW ORD	password	Das Passwort des in servBIRD existierenden (technischen) Users für die Verbindung zwischen servBIRD und dashBIRD Studio wenn SAML2 SSO verwendet wird
S y s t e m	SESSION_TIM EOUT	30	Session Timeout für das dashBIRD Studio Portal in Minuten
G l o b a l	SAML_SSO_E NABLED	false	Aktiviert SAML2 SSO für das dashBIRD Studio

G l o b a l	DASHLET_DATA_LIMIT	0	Globale Beschränkung der Anzahl der maximal durch ein DataSet abzurufenden Datensätze
S t y l e	STYLESHEET_NAME	absolution	Der Name des zu verwendenden Themes für das dashBIRD Studio Portal
S y s t e m	LOG_DELETION_INTERVAL	0	Anzahl der Tage nach denen alte Logeinträge gelöscht werden. Muss gesetzt werden, damit Logging stattfindet.
S y s t e m	STUDIO_LANGUAGE	de	Standardsprache für das dashBIRD Studio . Wird verwendet wenn die Browser Sprache nicht festgestellt werden kann.
G l o b a l	PREVIEW_LIMIT	100	Maximale Anzahl an Datensätzen die in der Vorschau von einer Abfrage angezeigt werden.

4.13 Erste Anmeldung

Nach dem erneuten Start des WildFly Application Servers können Sie das Portal des **dashBIRD Studio** im Webbrowser unter einem URL nach folgendem Schema erreichen: **PROTOCOL://HOST:PORT/**

Sie werden nach dem Aufruf der URL (entsprechend Ihrer Browsereinstellungen) automatisch auf die deutsche oder englische Version des **dashBIRD Studio** weitergeleitet. Nun können Sie sich an der Anmeldemaske mit den Authentifizierungsdaten eines im **servBIRD** vorhandenen Nutzers anmelden, sofern dieser über seine Rollen eine Lizenz für das **dashBIRD Studio** und die Berechtigungen für das **dashBIRD Studio** zugeordnet hat.

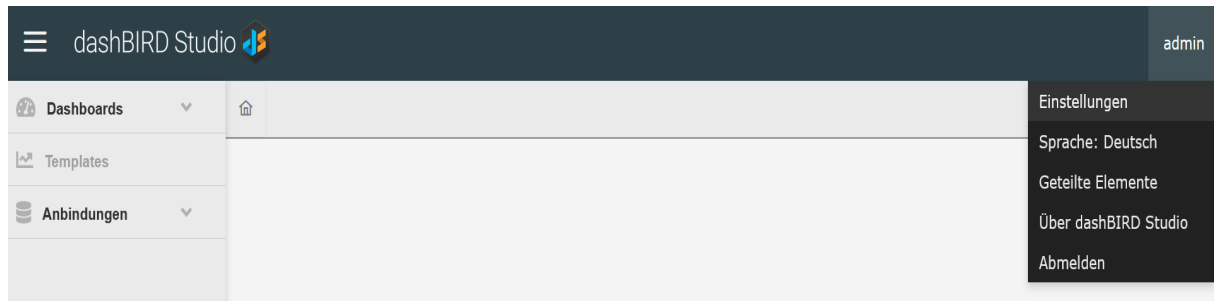
4.14 servBIRD Integration

Das **dashBIRD Studio** kann in **servBIRD** integriert werden, sodass es über einen eigenen Hauptmenüpunkt direkt aus dem **servBIRD** Portal heraus erreichbar ist. Hinterlegen Sie dazu in den **servBIRD** Konfigurationeneinstellungen die URL des **dashBIRD** (nach folgendem Schema: **PROTOCOL://HOST:PORT/**) in der Einstellung "dashBIRD → dashBIRD Studio URL".

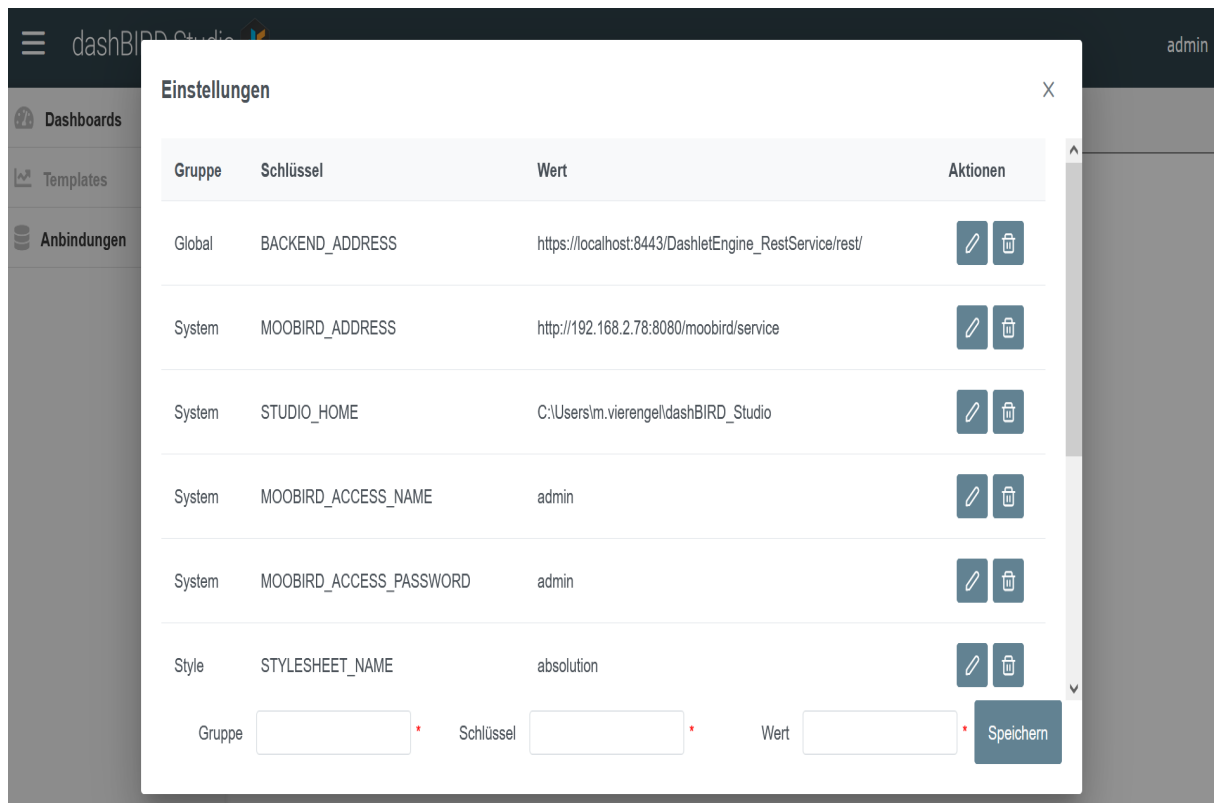
5 Einstellungen



Administratoren können die für das **dashBIRD Studio** vorgenommenen Einstellungen im Studio anzeigen lassen und verändern.

1. Klick auf den Nutzernamen in der oberen rechten Ecke.



2. Klick auf **Einstellungen**.
3. Eine tabellarische Ansicht der Einstellungen erscheint.



- Über das Abfall-Icon () unter 'Aktionen' können Einstellungen gelöscht werden.
- Über das Bleistift-Icon () unter 'Aktionen' können Einstellungen editiert werden.
- Über die Fußzeile können neue Einstellungen hinzugefügt werden.

5.1 Die folgenden Einstellungen sollten gesetzt sein

Gruppe	Schlüssel	Wert
Global	BACKEND_ADDRESS	Adresse der REST-Schnittstelle vom Backend (http://{ip}:{port}/DashletEngine_RestService/rest/)
System	MOOBIRD_ADDRESS	Adresse der angebundenen MooBIRD-REST-Schnittstelle (http://{ip}:{port}/moobird/service)
System	MOOBIRD_ACCESS_NAME	Nutzername für den Zugang zu MooBIRD
System	MOOBIRD_ACCESS_PASSWORD	Passwort für den Zugang zu MooBIRD
System	STUDIO_HOME	Pfad zum Home-Verzeichnis vom Studio, siehe Installationsleitfaden

5.2 Die folgenden Einstellungen können gesetzt werden

Gruppe	Schlüssel	Wert
Global	DASHLET_DATA_LIMIT	Standardbeschränkung der Anzahl der maximal durch ein DataSet abzurufenden Datensätze
Global	PREVIEW_LIMIT	Standardbeschränkung der Anzahl der maximal bei der Vorschau auf ein DataSet abzurufenden Datensätze
System	SESSION_TIMEOUT	Session Timeout für das dashBIRD Studio Portal in Minuten
System	STUDIO_LANGUAGE	Standardsprache für das dashBIRD Studio
System	LOG_DELETION_INTERVAL	Anzahl der Tage für die Logeinträge in der Datenbank gehalten werden
Style	STYLESHEET_NAME	Name von einem für das dashBIRD Studio definierten Stylesheet
Global	SAML_SSO_ENABLED	Ob das dashBIRD Studio SAML SSO unterstützt