

KIH Database

Systemdokumentation for KIH Databasen

1. maj 2013

Indholdsfortegnelse

Indholdsfortegnelse	2
Indledning	3
Systemoverblik	3
KIH Database applikationsserver	5
Forudsætninger før installation	5
Klargøring af database mv. før installation.....	6
Deployment til Apache Tomcat 7	8
Overblik over log-filer for KIH Databasen.....	9
Overvågning af KIH Database server	10
KIH Database monitoreringsnitflade	10
Backup og restore af KIH Databasen.....	11
Backup af Java EE web applikation	11
Backup af database.....	11
Restore af KIH Databasen.....	12
Dokumenthistorik	13

Indledning

Dette dokument udgør den tekniske systemdokumentation for KIH Databasen.

Systemdokumentationen indgår i den samlede KIH Database dokumentationspakke, hvis dele er beskrevet i "KIH Database dokumentationsoverblik".

Formålet med systemdokumentationen er at beskrive hvordan KIH Database platformen installeres, konfigureres og driftes. Dokumentet giver et overblik over konfigurationsfiler, log-filer og andre relevante objekter.

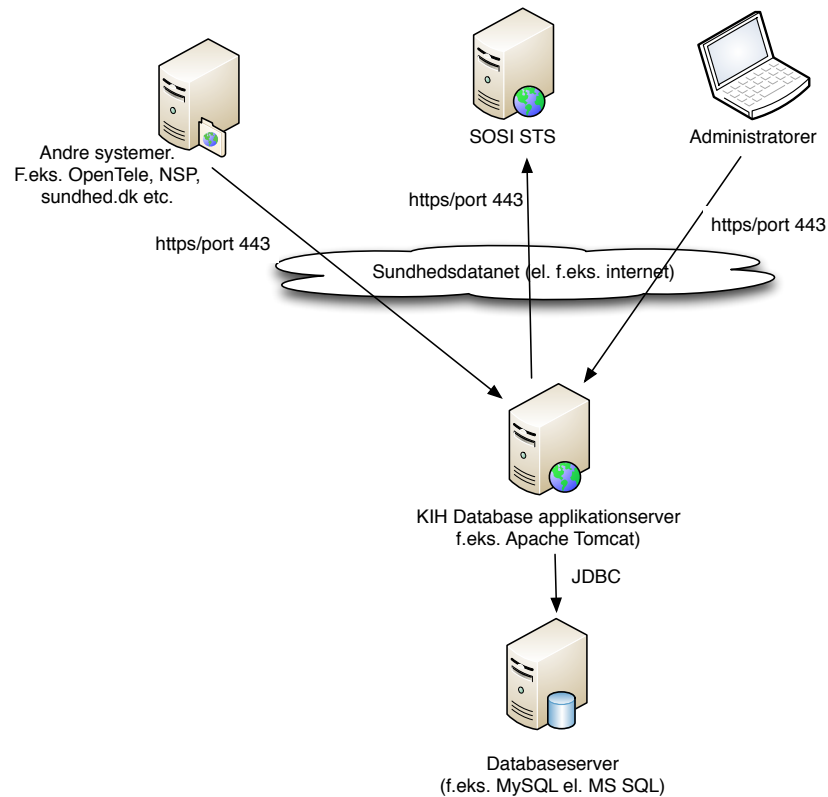
Systemdokumentationen tager udgangspunkt i, at en række komponenter er tilgængelige. De væsentligste af disse er et operativsystem, en database og en Java EE servlet container.

Systemoverblik

Dokumentet "KIH Database Arkitektur og design" beskriver den overordnede arkitektur for KIH Databasen. I den følgende gengives dele af arkitekturbeskrivelsen, med fokus på de elementer der er relevante for systemdokumentationen.

KIH Database platformen består af en Grails baseret serverapplikation, som afvikles i en Java EE web container.

Nedenfor er den overordnede principielle systemarkitektur illustreret.



I illustrationen vises en central KIH Database applikationsserver hvor KIH Database web applikationen er installeret.

Serveren udstiller en web baseret brugergrænseflade til administratorer, som de anvender via en browser. Til systemintegration udstilles en XML baseret web service, som eksterne systemer anvender til at indberette og hente data med.

KIH Database applikationsserveren anvender en standard SQL database som datalager.

Løsningen er testet med H2, MySQL 5 og SQL*Server 2008.

I det følgende antages at en Apache Tomcat 7 web-container anvendes på serversiden. KIH Database web applikationen er også testet på JBoss, og bør fungere uden store problemer i følgende Java EE Containere som er supporteret af Grails 2.1.0¹.

- Tomcat 7
- Tomcat 6
- SpringSource tc Server
- Eclipse Virgo
- GlassFish 3
- GlassFish 2
- Resin 4
- Resin 3
- JBoss 6

¹ Se <http://grails.org/doc/2.1.0/guide/single.html#supportedJavaEEContainers>

- JBoss 5
- Jetty 7
- Jetty 6
- IBM Websphere 7.0
- IBM Websphere 6.1
- Oracle Weblogic 10.3
- Oracle Weblogic 10
- Oracle Weblogic 9

KIH Database applikationsserver

KIH Databasens applikationsserver er en Java EE web applikation, udviklet i frameworket Grails 2.1.0.

I det følgende beskrives hvordan en Java EE web-applikation (WAR-fil) bygges og installeres på en applikationsserver.

Selve installationen af WAR-filen i Java EE applikationsserveren afhænger af hvilket operativsystem, samt hvilken applikationsserver - og databaseteknologi, man vælger at anvende. I det følgende eksempel anvendes version 7 af Apache Tomcat Servlet Container'en som applikationsserver, MySQL 5 som datalager og OS X 10.7 som operativsystem.

Forudsætninger før installation

Før installation af KIH Database server forudsættes følgende:

1. Adgang til en, to eller flere fysiske eller virtuelle maskiner med operativsystem installeret.

Det er op til en konkret designbeslutning i opsætningen af systemet, om man vil afvikle applikationsserver og database på separate maskiner. I det følgende antages at to separate maskiner anvendes.

2. På systemets applikationsserver skal en Java EE webcontainer være installeret. F.eks. JBoss eller Apache Tomcat 7. Applikationsserveren skal anvende Java 6. Java 7 er ikke understøttet.
3. På systemets databaseserver skal f.eks. MySQL eller SQL*Server være installeret.
4. En WAR-fil med KIH Database web-applikationen, bygget til det konkrete miljø skal være tilgængelig

Kompilering- og byg af en WAR applikation er beskrevet nærmere i dokumentet "KIH Database Guide til videreudvikling".

Kort beskrevet, skal et miljø (environment) defineres i web-applikationen (gøres af systemets udviklingsteam). Efterfølgende skal der bygges en WAR fil til det konkrete miljø. Hvis et miljø med navnet "osxtest" f.eks. er defineret, kan war-filen bygges med kommandoen:

```
grails -Dgrails.env=osxtest war
```

Klargøring af database mv. før installation

I det følgende beskrives hvordan KIH Databasen installeres (deploys) til en Apache Tomcat 7 web container. For installation i andre Java EE web containere, f.eks. JBoss, henvises til den sektion i manualen for den konkrete web containere, som beskriver installation af web-arkiver (WAR-filer).

Før installation af web-arkivet skal systemets database være installeret. Hvis MySQL anvendes kan dette f.eks. ske ved at installere MySQL, og efterfølgende oprette en database, og en KIH Database bruger, med kommandoerne i nedenstående boks:

```
create database kihdb character set utf8;

GRANT SELECT, INSERT, UPDATE, CREATE, DELETE, DROP, ALTER, INDEX on
kihdb.* TO ' kihdb@'localhost' IDENTIFIED BY 'kihdb';

flush privileges;
```

Husk at udskifte brugernavn og adgangskode i ovenstående.

Når databasen er installeret skal der oprettes en konfigurationsfil, hvorfra web-applikationen henter sin konfiguration.

På Windows placeres konfigurationsfilen i biblioteket:

```
c:\kihdatamon\settings\kihdb-config.properties
```

På Unix baserede systemer (f.eks. Linux eller OS X) placeres filen i biblioteket:

```
${userHome}/.kih/kihdb-config.properties
```

Hvor \${UserHome} angiver home-biblioteket for den bruger som web-containeren kører under.

Hvis filen ønskes placeret andetsteds, skal dette ændres i systemets kildekode.

Nedenfor to eksempler på KIH Database konfigurationsfiler. Én hvor web-applikationen anvender en MySQL database som datalager, og én hvor web-applikationens datalager er en Microsoft SQL*Server.

Eksempel på KIH Database konfigurationsfil med MySQL database:

```
# Common settings
#dataSource.dbCreate = update
dataSource.pooled = true
#dataSource.logSql = false

## MYSQL DB
dataSource.dialect = dk.silverbullet.kihdb.util.MySQLInnoDBDialect
dataSource.driverClassName = com.mysql.jdbc.Driver
dataSource.username = kihdb
dataSource.password = kihdb
dataSource.url = jdbc:mysql://10.0.0.1:3306/kihdb
```

Eksempel på KIH Database konfigurationsfil med Sql*Server database (OBS: Hvis katalog ikke anvendes på SQL*Server fjernes denne linie):

```
# Common settings
#dataSource.dbCreate = update
dataSource.pooled = true
#dataSource.logSql = false

# SQL Server DB
dataSource.driverClassName = net.sourceforge.jtds.jdbc.Driver
dataSource.username = kihdb
dataSource.password = kihdb
dataSource.url = jdbc:jtds:sqlserver://10.0.0.1:1433:kihdb
dataSource.dialect = dk.silverbullet.kihdb.util.SQLServerDialect
hibernate.default_schema = dbo
hibernate.default_catalog = kihdb
```

Konfigurationsfil parameteren "dataSource.dbCreate", mapper til Grails' standard databaseopsætnings (dbCreate) parameter og kan antage værdierne i nedenstående tabel. I et produktionssetup, anbefales det at fjerne parameteren helt, eller sætte den til værdien "validate", da der ellers er risiko for, at systemet vil forsøge at overskrive dets database når det startes/genstartes.

Hvis man i forbindelse med installationen af KIH Databasen ønsker at oprette tabeller mv. i systemets relationelle database, kan det gøres ved at starte systemet med værdien sat til: `dataSource.dbCreate = create`. Efterfølgende - når systemet er startet og tabelstruktur er oprettet - skal parameteren ændres til f.eks. `dataSource.dbCreate = validate`.

En bedre metode til oprettelse af tabeller er, at systemets udviklere danner et tabel-oprettelsesscript (DDL). Og at systemet enten konfigureres med en tom "dataSource.dbCreate" eller med "dataSource.dbCreate = validate."

dbCreate værdi	Betydning
create	Ved opstart af web-applikationen nyoprettes tabeller og andre databaseobjekter.
update	Ved opstart af web-applikationen oprettes tabeller og andre databaseobjekter som ikke i forvejen eksisterer i databasen.
validate	Ved opstart af web-applikationen validerer applikationen om alle nødvendige databaseobjekter er tilstede. Hvis dette ikke er tilfældet kan applikationen ikke starte (deploys) og logger i stedet en fejl til en dens log-filer.
' '	Tom værdi. Systemet antager at alle nødvendige databaseobjekter er til stede, men gør ikke noget for at undersøge om det er tilfældet.

Deployment til Apache Tomcat 7

Når database- og konfigurationsfiler er klargjort kan web-applikationen (WAR-filen) installeres på systemets applikationsserver.

På Apache Tomcat sker dette ved at kopiere war-filen (f.eks. kih_database.war) til Tomcat-serverens "webapps" bibliotek.

På OS X og Linux kan dette gøres i en shell med kommandoen:

```
cp kih_database.war ${TOMCAT_HOME}/webapps/
```

Afhængigt af hvordan Apache Tomcat serveren er konfigureret, vil ovenstående kommando installere KIH Database web-applikationen, så dens "rod-kontekst" er "kih_database", når man åbner den fra en web-browser, d.v.s.:

```
http://<SERVER_URL>/kih_database
```

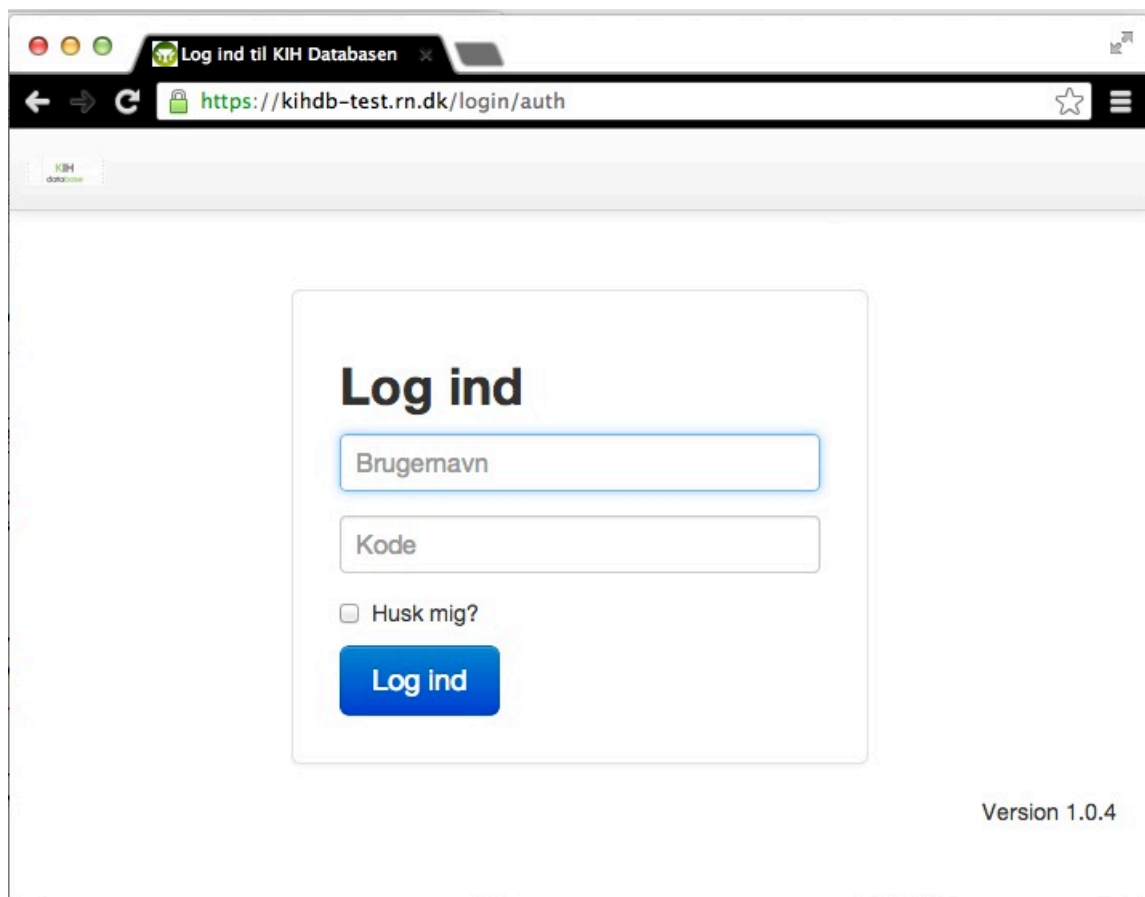
Hvis man i stedet ønsker, at applikationen skal installeres så brugere ikke skal angive en "rod-kontekst" kan det gøres ved at omdøbe war-filen til "ROOT.war" når den kopieres til webapps biblioteket:

```
cp kih_database.war ${TOMCAT_HOME}/webapps/ROOT.war
```


Når applikationen installeres navngivet "ROOT.war", forstår Tomcat det sådan, at den skal udbyde applikationen uden en "rod-kontekst". D.v.s. på adressen:

`http://<SERVER_URL>`

Efter installation, verificeres om serveren er korrekt installeret ved at overvåge om den skriver fejl til sine log-filer, og ved at åbne applikationen i en web-browser. Nedenfor er vist et eksempel på, hvordan applikationens log-ind side ser ud på miljøet "kihdb-test", når den er deployet som "ROOT.war".



Overblik over log-filer for KIH Databasen

KIH Database server er konfigureret til at sende sine logs til Tomcats standard-logbibliotek. Dvs.:

```
<TOMCAT>/logs
```

Hvis systemet skal skrive log-filer til en anden lokation, skal dette konfigureres af udviklingsorganisationen (i filen "Config.groovy").

I standard-opsætningen logger KIH Database webapplikationen til filen "kihdb.log". Som standard er applikationen konfigureret til at danne en ny log-fil på dagsbasis.

D.v.s. nyeste log-fil er filen "kihdb.log". Ældre log-filer omdøbes, så de efter "kihdb.log" tilføjes datoen for de logninger filen indeholder. F.eks.:

Filnavn	Beskrivelse
kihdb.log	Seneste logninger fra KIH Database web-applikationen
kihdb.log.2013-04-30	Logninger fra 30. april 2013
kihdb.log.2013-04-29	Logninger fra 29. april 2013
kihdb.log.2013-04-28	Logninger fra 28. april 2013

Logfilen bør ikke indeholde fejl-logninger. Disse er kendetegnet ved at ordet "ERROR" indgår i en lognings-linie.

Overvågning af KIH Database server

I dette afsnit beskrives den snitflade KIH Database serveren stiller til rådighed for anvendelse i en driftsmæssig overvågning af platformen.

En samlet overvågning af systemet bør også omfatte overvågning af systemets processer samt databaseserver og diskplads m.v. Hvilke processer der skal være kørende på systemet afhænger af hvilken Databaseteknologi og hvilken Java EE webcontainer systemet anvender. Der henvises derfor til dokumentationen for de konkrete produkter.

KIH Database monitorerings-snitflade

KIH Databasen udstiller en simpel web-side, som kan anvendes til overvågning af serverens tilstand fra enten et overvågningssystem eller f.eks. en browser.

Web-siden udstilles på adressen:

```
http(s)://<KIH DATABASE-URL>/<ROD-KONTEKST>/isAlive
```

Når siden kaldes, undersøger systemet om det er korrekt kørende og om det har mulighed for at skabe forbindelse til dets database. I nedenstående skærmbillede vises et eksempel på siden, som den ser ud når alt er fungerende:



Der er fire værdier på siden, som beskrives i følgende tabel:

Tekst	Beskrivelse
responseText	Svarer "I'm alive!" hvis systemets applikationsserver er fungerende.
runningVersion	Angiver hvilken version af KIH Database der er installeret på serveren.
environment	Angiver hvilket "miljø" serveren er konfigureret som (eksempelvis test, devel eller production)
isDatabaseAlive	Angiver om applikationsserveren (f.eks. Tomcat) kan oprette en forbindelse til systemets SQL database. Mulige svar er: "Yes." eller "No.".

For at have et fuldt fungerende system, skal responseText være "I'm alive!" og isDatabaseAlive være "Yes.".

Backup og restore af KIH Database

I det følgende angives anbefalinger for backup/restore strategier for KIH Database applikationen. Den konkrete strategi – og fremgangsmåde – afhænger af hvilke teknologier en konkret installation anvender.

Backup af Java EE web applikation

KIH Database server applikationen gemmer alle data i sin database. Det bør derfor ikke være nødvendigt at lave løbende backup af applikationsserverens filsystem.

Det anbefales derfor, at der efter installation laves en initial backup af maskinen serveren er installeret på, og at der løbende tages backup af serverens konfigurationsfil (beskrevet i tidligere afsnit).

Hvorvidt der udføres backup af web applikationens log-filer, beslutes i forbindelse med den konkrete installation.

Backup af database

Det anbefales at der etableres løbende backup af systemets database, således data ikke kan gå tabt ved evt. systemnedbrud.

Restore af KIH Databasen

Hvis der er sket en hændelse, som medfører at det er nødvendigt at lave restore af hele – eller dele af – KIH Database installationen, anbefales følgende overordnede fremgangsmåde:

- 1) Kontrollér, at alle systemets applikationsservere, og andre systemer som skriver til systemets database, er standsede
- 2) Gendan systemets database
- 3) Start databaseserveren
- 4) Lav restore af systemets applikationsservere (eller start dem igen)

Dokumenthistorik

Version	Dato	Initialer	Ændring
1.0	01-05-2013		Initiel version