**Installazione**

**Manuale**

**Servizi Scolastici**

Sistema Palermo Innovazione SpA

|  |  |
| --- | --- |
| **Titolo Documento:** | Manuale: Installazione |
| **Ambito** | Servizi Scolastici |
| **Data** | 31/01/2022 |
| **Revisione** | 1.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Revisioni** | | | |
| **Versione:** | **Data** | **Modifiche** | **Autore** |
| 1.0 | 31/01/2022 | Prima versione | SISPI |

**Sommario**

[1 INTRODUZIONE 3](#_Toc97776434)

[1.1 Scopo 3](#_Toc97776435)

[1.2 Soluzione tecnica 3](#_Toc97776436)

[2 ARCHITETTURA 4](#_Toc97776437)

[2.1 Architettura tecnologica e funzionalità applicative 4](#_Toc97776438)

[2.2 Architettura logica 6](#_Toc97776439)

[2.1 Clustering e Scalabilità 7](#_Toc97776440)

[3 INSTALLAZIONE e configurazione 8](#_Toc97776441)

[3.1 zookeeper 9](#_Toc97776442)

[3.2 solr 10](#_Toc97776443)

[3.3 activemq 11](#_Toc97776444)

[3.4 gotenberg 12](#_Toc97776445)

[3.5 ks-api-docer 13](#_Toc97776446)

[3.6 ks-api-bpm 14](#_Toc97776447)

[3.7 ks-api-portal 17](#_Toc97776448)

[3.8 ks-api-payment 19](#_Toc97776449)

[3.9 ks-api-sispi 21](#_Toc97776450)

[3.10 ks-api-bl 23](#_Toc97776451)

[3.11 ks-rpc-docer 24](#_Toc97776452)

[3.12 ks-spa-form 25](#_Toc97776453)

[3.13 ks-web-desktop 27](#_Toc97776454)

[3.14 ks-web-flowdesigner 29](#_Toc97776455)

**Lista Figure**

[Figura 1 - Architettura logica 6](#_Toc97737539)

# INTRODUZIONE

## Scopo

Il presente documento ha lo scopo di riepilogare la soluzione tecnica del progetto Servizi Scolastici, descrivere brevemente l’architettura logica, e focus sulle specifiche tecniche di installazione dei servizi applicativi che ne fanno parte.

## Soluzione tecnica

La soluzione prevede l’installazione e la configurazione della piattaforma tecnologica della piattaforma KeySuite Freeware, nelle versioni dei software implementate ed in uso al Comune di Palermo, sugli ambienti Cloud messi a disposizione dall’Amministrazione per la gestione e l’erogazione della piattaforma per il periodo richiesto dal progetto di Servizi Scolastici.

Assunzioni:

* Non sono previste le componenti di data protection (es. backup) assumendo che siano già in essere presso il Cliente
* Non sono state stimate le licenze del software applicativo (es. middleware, db, ecc) in carico al Cliente
* Non sono previste le componenti core di rete e di sicurezza (es. bilanciatori, firewall, log management, ecc.)

L’architettura infrastrutturale proposta è basata su soluzioni software convergenti basate su architetture di riferimento modulari e altamente scalabili ed è in grado di:

* Soddisfare i requisiti di capacità futuri senza limitare prestazioni o funzionalità.
* Implementare un’infrastruttura di virtualizzazione efficiente, con vantaggi chiari di gestione e di consolidamento
* Supportare il consolidamento dell'IT con le architetture scalabili personalizzate in linea con i requisiti delle applicazioni

# ARCHITETTURA

## Architettura tecnologica e funzionalità applicative

L’architettura della piattaforma Servizi Scolastici è una architettura a servizi implementata nel rispetto del massimo disaccoppiamento tra le diverse funzionalità disponibili: ogni servizio della piattaforma svolge il proprio specifico ruolo rendendo disponibile le sue funzionalità alle altre applicazioni.

In particolare, i principali servizi applicativi sono:

* Sistema di gestione documentale (Document Management System, DMS) basato su Solr Cloud 8 opportunamente esteso per rispettare i dettami di una corretta gestione archivistica dei documenti.
* Sistema per l’orchestrazione di procedimenti amministrativi e flussi documentali (Business Process Management System, BPM) basato su JBPM 7 esteso con funzionalità ad-hoc per fornire una soluzione completa per l’automazione dei processi.
* Console di Back-Office a supporto degli uffici preposti che fornisce tutti gli strumenti per una completa interazione con i diversi procedimenti amministrativi e flussi documentali.
* Portale dei Servizi On-Line, lo strumento facile, veloce e diretto, rivolto a cittadini, imprese ed utenti esterni per interagire con gli uffici di Enti pubblici o privati tramite sportelli virtuali polifunzionali.
* Designer BPMN 2.0, strumento web based molto intuitivo che consente di modellare un qualunque processo/procedimento evidenziando la struttura e la scomposizione di questo in attività elementari. Il Designer permette inoltre, attraverso l’integrazione con il servizio Form Manager, di disegnare i form accessibili di interazione con l’utente previsti nel procedimento. Queste caratteristiche consentono di utilizzare la terminologia comune alla PA, rendendo il disegno dei processi facilmente leggibile e riusabile, ma al tempo stesso conforme alla notazione standard BPMN2.

Le tecnologie JBPM e Solr Cloud rappresentano i maggiori punti di forza della piattaforma Keysuite Freeware.

Per il servizio di repository documentale si è scelto di utilizzare la tecnologia Open Source Solr Cloud in quanto presenta una architettura estremamente flessibile che opportunamente configurata e personalizzata, gestisce indicizzazione, memorizzazione dei file, sicurezza applicata alla ricerca, modelli flessibili e personalizzabili di definizione dei metadati da applicare ai documenti.

Il motore JBPM, invece, è una suite Java per la gestione dei processi di business rilasciata da Jboss Company. Il prodotto permette di modellare, eseguire e monitorare i processi durante l’intero ciclo. Supporta l’ultima versione (2.0) del linguaggio di definizione BPMN, che rappresenta lo standard definito dalla OMG (Object Management Group) con cui descrivere graficamente e funzionalmente i processi di business per facilitarne la comprensione a tutti gli stakeholder (user, analyst, business administrator, developer).

## Architettura logica

Il diagramma seguente illustra nel dettaglio l’architettura logica di tutti i servizi e i layer che compongono la soluzione KeySuite (Figura 1).

Graphical user interface

Description automatically generated

Figura - Architettura logica

## Clustering e Scalabilità

Ogni servizio di KeySuite è deployabile in cluster e scalabile indipendentemente al fine di implementare bilanciamento del carico (Load Balancing) e alta disponibilità (High Availability) e fornire una soluzione robusta e performante, dimensionata a seconda dei requisiti di carico previsti.

Tutti i servizi, ad esclusione di Solr, sono stateless, per cui possono essere scalati orizzontalmente col necessario numero di repliche. Il servizio BPM API non è puramente stateless e necessita di un volume configurato e condiviso dalle repliche per il tracciamento dei log delle transazioni.

Per scalare il servizio Solr, è invece necessario configurare opportunamente le repliche per l’interazione tramite Zookeeper.

La configurazione in cluster di Solr attraverso Zookeeper, così come il clustering degli altri servizi stateless, non è trattata in questo documento, così come le piattaforme e/o i componenti per la gestione del load balancing non sono inclusi nella distribuzione software rilasciata.

# INSTALLAZIONE e configurazione

KeySuite è una architettura a servizi indipendenti che astrae dall’architettura fisica di deploy.

I servizi KeySuite possono essere deployati su una o più macchine fisiche, virtuali (VM) o containerizzati su cluster (e.g. Kubernetes, Openshift). Quest’ultima opzione è comunque fuori dagli obiettivi del seguente documento.

Nei pacchetti di installazione forniti, la KeySuite è preconfigurata per una istallazione base standalone di tutti i servizi applicativi, infrastrutturali, e dei database necessari, sulla una stessa macchina Linux (fisica o virtuale) sotto la directory di installazione **/opt/ks6**. Tutti i servizi sono dunque preconfigurati per essere mutualmente raggiungibili all’host **localhost** sulla porta di default esposta dal servizio (diversa per ciascun servizio).

Naturalmente, configurando opportunamente i vari servizi come illustrato in seguito, è possibile il deploy di ciascun servizio o gruppo di servizi su una VM dedicata (scelta raccomandata) e/o variare il path di installazione o la porta esposta (scelta sconsigliata).

Segue per ogni servizio uno schema sintetico riassuntivo per l’installazione e la configurazione.

## zookeeper

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **9983** (tcp client)
  + 2888 (tcp cluster peer)
  + 3888 (tcp cluster leader)
* Database: N/A

**Installazione**

Scompattare il pacchetto **zookeeper.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nel file conf/zoo.cfg

* dataDir=**/opt/ks6**/zookeeper/data

Sostituire la cartella di installazione se diversa da /opt/ks6.

**Start/Stop**

bin/zkServer.sh start

bin/zkServer.sh stop

## solr

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **8983** (tcp search)
  + 8981 (tcp indexing)
* Database: Utente sispi
  + Oracle 12c+

**Installazione**

Scompattare il pacchetto **solr.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nei file di definizione dei report server/reports/\*.properties

* datasource.sispi=jdbc:oracle:thin:@**localhost**:1521/**sid**?user=**username**&password=**password**

Configurare la connessione al db Oracle (la stessa del servizio ks-api-sispi).

**Start/Stop**

bin/start.sh

bin/stop.sh

## activemq

L’nstallazione di un Message Broker è opzionale. Si rende necessaria solo nel caso in cui i processi deployati ed eseguiti dal servizio BPM abbiano la necessita di sottoscrivere/pubblicare eventi (messaggi) da/verso sistemi esterni alla piattaforma KeySuite, caso non contemplato nella soluzione a riuso proposta.

Lo standard per la gestione dei messaggi è JMS e l’implementazione di riferimento adottata è **Apache ActiveMQ.** Si veda la documentazione ufficiale per il download, l’installazione e la configurazione.

<https://activemq.apache.org/components/classic/documentation>

Il servizio risponde alla porta di default **61616.**

## gotenberg

Il servizio di Document Conversion è implementato attraverso il motore di conversione PDF open source **Gotenberg**. Si veda la documentazione ufficiale per il download, l’installazione e la configurazione.

<https://gotenberg.dev/docs/about>

Il servizio risponde alla porta di default **3000**.

## ks-api-docer

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **8080** (http)
* Database: N/A

**Installazione**

Scompattare il pacchetto **ks-api-docer.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nel file config/resources/system.properties

* client.tmp=**/opt/ks6**/ks-api-docer/tmp/ws
* DISK\_BUFFER\_DIRECTORY=**/opt/ks6**/ks-api-docer/tmp/buffer
* tempfiles=**/opt/ks6**/ks-api-docer/tmp/files
* tempfiles.upload=**/opt/ks6**/ks-api-docer/tmp/files/upload
* zkHost=**localhost**:9983

Sostituire la cartella di installazione se diversa da /opt/ks6.

Sostituire l’host/IP di puntamento interno dei servizi

* zookeeper (localhost:9983)

**Start/Stop**

run.sh

stop.sh

## ks-api-bpm

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **8083** (http)
* Database: Utente/Schema bpm
  + MySql 5.7+
  + Oracle 12c+
  + PostgreSQL 10+
  + SqlServer 2017+

**Installazione**

Scompattare il pacchetto **ks-api-bpm.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nel file config/application.yml

* spring.datasource:
  + url: **jdbc:mysql://localhost:3306/bpm**?serverTimezone=UTC&rewriteBatchedStatements=true
  + username: **username**
  + password: **password**
  + driver-class-name: **com.mysql.cj.jdbc.MysqlXADataSource**
* spring.jpa.properties:
  + hibernate.dialect: **org.hibernate.dialect.MySQL57InnoDBDialect**
* spring.jta:
  + log-dir: **/opt/ks6**/ks-api-bpm/tx/log
* quartz.datasource:
  + url: **jdbc:mysql://localhost:3306/bpm**?serverTimezone=UTC
  + username: **username**
  + password: **password**
  + driver-class-name: **com.mysql.cj.jdbc.Driver**
* email.smtp:
  + defaultFrom: **email@example.com**
  + sources:
    - from: **email@example.com**
* personal: **personal**
* host: **smtp.host**
* port: **587**
* authentication: **true**
* username: **username**
* password: **password**
* protocol: **smtp**
* tls: **true**
* jms:
  + brokerUrl: **vm://embedded-broker?broker.persistent=false**
  + brokerUser:
  + brokerPassword:
* env.systemProperties:
  + name: docer.url
* value: http://**localhost**:8080
  + name: tempfiles.upload
* value: **/opt/ks6**/ks-api-docer/tmp/files/upload
* rpc.docerActionService:
  + url: http://**localhost**:8089
* form.service:
  + url: http://**localhost**:8086/form-manager
* application.backoffice:
  + baseUrl: **http://ks-backoffice:8082**
* application.document:
  + baseUrl: http://**localhost**:8080/files/upload
  + basePath: **/opt/ks6**/ks-api-docer/tmp/files/upload
* application.datastore.jdbc.datasources:
  + name: local
* platform: **MySQL5**
* schema: **bpm**
  + name: sispi
* platform: Oracle10
* schema: **username**
* url: jdbc:oracle:thin:@**localhost**:1521/**sid**
* username: **username**
* password: **password**
* driverClassName: oracle.jdbc.driver.OracleDriver
* loader.path: **/opt/ks6**/ks-api-bpm/libs

Sostituire la cartella di installazione se diversa da /opt/ks6.

Configurare il datasource bpm spring (XA) e quartz (Standard).

Configurare la connessione al db Oracle (la stessa del servizio ks-api-sispi).

Configurare uno o più client SMTP

Configurare JMS (opzionale)

Sostituire l’host/IP di puntamento interno dei servizi

* ks-api-docer (localhost:8080)
* ks-rpc-docer (localhost:8089)
* ks-spa-form (localhost:8086)

Sostituire la base url pubblica (DNS) dell’applicazione web di Back-Office (http://ks-backoffice:8082)

**Start/Stop**

run.sh

stop.sh

## ks-api-portal

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **8084** (http)
* Database: Utente/Schema portal
  + MySql 5.7+
  + Oracle 12c+
  + PostgreSQL 10+
  + SqlServer 2017+

**Installazione**

Scompattare il pacchetto **ks-api-portal.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nel file config/application.yml

* spring.datasource:
  + url: **jdbc:mysql://localhost:3306/portal**?serverTimezone=UTC&rewriteBatchedStatements=true
  + username: **username**
  + password: **password**
  + driver-class-name: **com.mysql.cj.jdbc.Driver**
* spring.jpa.properties:
  + hibernate.dialect: **org.hibernate.dialect.MySQL57InnoDBDialect**
* env.systemProperties:
  + name: docer.url
* value: http://**localhost**:8080
* mail:
  + protocol: **smtp**
  + host: **smtp.host**
  + port: **587**
  + username: **username**
  + password: **password**
  + properties:
    - mail.smtp.auth: **true**
    - mail.smtp.starttls.enable: **true**
* application:
  + mailTo: **email@example.com**
  + baseUrl: **http://ks-portal:8082**
  + bpmBaseUrl: http://**localhost**:8083
  + paymentBaseUrl: http://**localhost**:8091
  + docerBaseUrl: http://**localhost**:8080

Sostituire la cartella di installazione se diversa da /opt/ks6.

Configurare il datasource portal.

Configurare il client SMTP.

Sostituire l’host/IP di puntamento interno dei servizi

* ks-api-docer (localhost:8080)
* ks-api-bpm (localhost:8083)
* ks-api-payment (localhost:8091)

Sostituire la base url pubblica (DNS) dell’applicazione web di Back-Office (http://ks-backoffice:8082)

**Start/Stop**

run.sh

stop.sh

## ks-api-payment

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **8091** (http)
* Database: Utente/Schema payment
  + MySql 5.7+
  + Oracle 12c+
  + PostgreSQL 10+
  + SqlServer 2017+

**Installazione**

Scompattare il pacchetto **ks-api-payment.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nel file config/application.yml

* spring.datasource:
  + url: **jdbc:mysql://localhost:3306/payment**?serverTimezone=UTC&rewriteBatchedStatements=true
  + username: **username**
  + password: **password**
  + driver-class-name: **com.mysql.cj.jdbc.Driver**
* spring.jpa.properties:
  + hibernate.dialect: **org.hibernate.dialect.MySQL57InnoDBDialect**
* env.systemProperties:
  + name: docer.url
* value: http://**localhost**:8080
* application:
* baseUrl: **http://ks-portal:8082**/pagamenti/v1
* portalBaseUrl: **http://ks-portal:8082**
* portalAPIBaseUrl: http://**localhost**:8084
* ksUtilsAPIBaseUrl: http://**localhost**:8085
* docerAPIBaseUrl: http://**localhost**:8080
* expiryTimeInHours: 2
* pagoPAUrl: https://secure.pmpay.it/

Sostituire la cartella di installazione se diversa da /opt/ks6.

Configurare il datasource portal.

Sostituire l’host/IP di puntamento interno dei servizi

* ks-api-docer (localhost:8080)
* ks-api-portal (localhost:8084)
* ks-api-bl (localhost:8085)

Sostituire la base url pubblica (DNS) dell’applicazione web di Portale (http://ks-portal:8082)

**Start/Stop**

run.sh

stop.sh

## ks-api-sispi

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **8088** (http)
* Database: Utente sispi
  + Oracle 12c+

**Installazione**

Scompattare il pacchetto **ks-api-sispi.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Inizializzare lo schema sispi con lo script schema.sql fornito.

Nel file config/application.yml

* spring.datasource:
  + url: jdbc:oracle:thin:@**localhost**:1521/**sid**
  + username: **username**
  + password: **password**
  + driver-class-name: oracle.jdbc.OracleDriver
* spring.jpa.properties:
  + hibernate.dialect: org.hibernate.dialect.Oracle12cDialect
* env.systemProperties:
  + name: docer.url
* value: http://**localhost**:8080
* mail:
  + protocol: **smtp**
  + host: **smtp.host**
  + port: **587**
  + username: **username**
  + password: **password**
  + properties:
    - mail.smtp.auth: **true**
    - mail.smtp.starttls.enable: **true**
* application:
  + mailTo: **email@example.com**

Configurare il datasource sispi.

Configurare il client SMTP.

Sostituire l’host/IP di puntamento interno dei servizi

* ks-api-docer (localhost:8080)

**Start/Stop**

run.sh

stop.sh

## ks-api-bl

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **8085** (http)
* Database: N/A

**Installazione**

Scompattare il pacchetto **ks-api-bl.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nel file config/application.properties

* spring.solr.zkHost=**localhost**:9983
* spring.docer.host=**localhost**:8080
* convert.host=http://**localhost**:3000
* docer.url=http://**localhost**:8080
* tempfiles.upload=**/opt/ks6**/ks-api-docer/tmp/files/upload

Sostituire la cartella di installazione se diversa da /opt/ks6.

Sostituire l’host/IP di puntamento interno dei servizi

* zookeeper (localhost:9983)
* gotenberg (localhost:3000)
* ks-api-docer (localhost:8080)

**Start/Stop**

run.sh

stop.sh

## ks-rpc-docer

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **8089** (http)
* Database: N/A

**Installazione**

Scompattare il pacchetto **ks-rpc-docer.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nel file config/application.yml

* env.systemProperties:
  + name: docer.url
* value: http://**localhost**:8080
  + name: api-bpm-baseUrl
* value: http://**localhost**:8083
  + name: tempfiles.upload
* value: **/opt/ks6**/ks-api-docer/tmp/files/upload

Sostituire la cartella di installazione se diversa da /opt/ks6.

Sostituire l’host/IP di puntamento interno dei servizi

* ks-api-docer (localhost:8080)
* ks-api-bpm (localhost:8083)

**Start/Stop**

run.sh

stop.sh

## ks-spa-form

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **8086** (http)
* Database: Utente/Schema form
  + MySql 5.7+
  + Oracle 12c+
  + PostgreSQL 10+
  + SqlServer 2017+

**Installazione**

Scompattare il pacchetto **ks-spa-form.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nel file config/application.yml

* spring.datasource:
  + url: **jdbc:mysql://localhost:3306/form**?serverTimezone=UTC&rewriteBatchedStatements=true
  + username: **username**
  + password: **password**
  + driver-class-name: **com.mysql.cj.jdbc.Driver**
* spring.jpa.properties:
  + hibernate.dialect: **org.hibernate.dialect.MySQL57InnoDBDialect**
* env.systemProperties:
  + name: docer.url
* value: http://**localhost**:8080
* application:
* preview.externalUrl:
* **http://ks-backoffice:8082**/~form-preview?formId=$formId
* preview.assetsBaseUrl:
* **http://ks-backoffice:8082**/static/

Configurare il datasource form.

Sostituire l’host/IP di puntamento interno dei servizi

* ks-api-docer (localhost:8080)

Sostituire la base url pubblica (DNS) dell’applicazione web di Back-Office (http://ks-backoffice:8082)

**Start/Stop**

run.sh

stop.sh

## ks-web-desktop

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **8082** (http)
* Database: N/A

**Installazione**

Scompattare il pacchetto **ks-web-desktop.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nel file config/system.properties

* host.ks-backoffice=**http://ks-backoffice:8082**
* host.ks-portal=**http://ks-portal:8082**
* zuul.routes.bl.url=http://**localhost**:8085/bl
* zuul.routes.restutils.url=http://**localhost**:8085/rest-utils
* zuul.routes.portal.url=http://**localhost**:8084
* zuul.routes.payment.url=http://**localhost**:8091
* zuul.routes.form.url=http://**localhost**:8086/form-manager
* zuul.routes.docer.url=http://**localhost**:8080
* zuul.routes.bpm.url=http://**localhost**:8083
* zuul.routes.sispi.url=http://**localhost**:8088

e nel file config/resources.backoffice/apps.yaml

* apps
  + link: solr/#/
* targetUri: http://**localhost**:8983
  + link: FlowDesigner/
* targetUri: http://**localhost**:9000
  + link: form-manager/console
* targetUri: http://**localhost**:8086

Sostituire l’host/IP di puntamento interno dei servizi

* solr (localhost:8983)
* ks-api-docer (localhost:8080)
* ks-api-bl (localhost:8085)
* ks-api-portal (localhost:8084)
* ks-api-payment (localhost:8091)
* ks-spa-form (localhost:8086)
* ks-api-bpm (localhost:8083)
* ks-api-sispi (localhost:8088)
* ks-web-flowdesigner (localhost:9000)

Sostituire la base url pubblica (DNS) dell’applicazione web di Back-Office (http://ks-backoffice:8082).

Sostituire la base url pubblica (DNS) dell’applicazione web di Portale (http://ks-portal:8082)

**Start/Stop**

run.sh

stop.sh

## ks-web-flowdesigner

**Prerequisiti**

* OS: Linux, Centos 7+ raccomandato
* Java: JDK 11+
* Porte
  + **9000** (http)
* Database: Schema flowdesigner
  + MySql 5.7+

**Installazione**

Scompattare il pacchetto **ks-web-flowdesigner.zip** nella cartella di installazione (default /opt/ks6)

**Configurazione**

Nel file run.sh

* -Drepository.jdbc.url=
* jdbc:mysql://**localhost**:3306/**flowdesigner**?serverTimezone=UTC
* -Drepository.db.name=**flowdesigner**
* -Drepository.db.username=**username**
* -Drepository.db.password=**password**

Configurare il datasource flowdesigner.