

memYAK

Installationsbeschreibung

Für memYAK Version 2.7.6 und Apache Tomcat.

Stand: 8. Juli 2015

Umgebung.....	3
Dateien	3
Anwendungskonfiguration.....	4
Loggingkonfiguration.....	4
Administrator	4
Ports, Absicherung und Dateigröße.....	5
E-Mail	6
MongoDB	8
Benutzerregistrierung	9
Kennwort-Validator	10
Support Information	10
Text im Anmeldebildschirm.....	11
Master-Instanz	11
Nutzung von Google Maps	12
Tomcat Einstellungen	13
Linux Einstellungen	14
Deployment	14
Erste Anmeldung	14
Neue Versionen installieren.....	15
Backup	15

Umgebung

memYAK setzt die folgende Umgebung voraus:

- Java Version 6

Die Laufzeitumgebung muss mindestens Java 6 sein.

- Apache Tomcat

tomcat.apache.org

- MongoDB

mongodb.org

Benutzen Sie nicht die MongoDB Versionen 2.6.1 bis 2.6.3, diese Versionen sind mit memYAK nicht kompatibel. Verwenden Sie stattdessen MongoDB 2.4.x. oder die MongoDB-Versionen ab 2.6.4.

Die Nutzung von MongoDB in der 64bit Variante wird empfohlen, da die Mongo Datenbanken in der 32bit Version nicht größer als 2GB werden können.

In diesem Dokument wird erläutert, wie memYAK in der zuvor genannten Umgebung installiert wird. Es wird nicht erläutert, wie diese Umgebung herzustellen ist.

Dateien

- memYAK Anwendung

Die gesamte memYAK Anwendung wird in einem WAR-File ausgeliefert.

memyak-[version].war

- memYAK Konfiguration

Die Konfiguration der memYAK Anwendung ist in der ZIP Datei **config-[version].zip** enthalten. Diese Datei enthält den Verzeichnispfad **config/memyak** und dieses Verzeichnis enthält wiederum zwei Dateien:

- **memyakContext.xml**

In dieser Datei wird das Verhalten von memYAK konfiguriert.

- **memyakLogging.xml**

In dieser Datei wird das Verhalten des Logging-Systems von memYAK konfiguriert. Es handelt sich dabei um eine Logback¹ Konfigurationsdatei.

¹ <http://logback.qos.ch>

Anwendungskonfiguration

In der Datei **memyakContext.xml** legen Sie unter anderem fest, welche Sicherheitseigenschaften das System garantiert, welche Ports benutzt werden und wie die Verbindung zum MongoDB-System hergestellt wird, in dem memYAK sämtliche Daten speichert.

Auch der SMTP Mail-Provider, den memYAK zum Versenden von E-Mails verwendet, wird hier konfiguriert.

Diese Datei sollten Sie in einem separaten Verzeichnis des Tomcat ablegen, z.B. in **\${catalina.home}/config/memyak**. Die weiteren hierzu erforderlichen Arbeitsschritte werden im Abschnitt **Tomcat Konfiguration** weiter unten erläutert.

Loggingkonfiguration

memYAK verwendet Logback Logging. Die Konfiguration wird in der Datei **memyak-Logging.xml** vorgenommen.

Auch diese Datei sollten Sie in einem separaten Konfigurationsverzeichnis speichern, z.B. gemeinsam mit der Datei **memyakContext.xml** (siehe oben, **Anwendungskonfiguration** und unten im Abschnitt **Tomcat Konfiguration**).

Die Loggingkonfiguration **memyakLogging.xml** ist für das Deployment im Tomcat vorbereitet und muss normalerweise nicht angepasst werden. memYAK erzeugt eine Logging-Datei mit Namen **memyak.log**, diese Datei wird im **logs** Verzeichnis des Tomcat Servers gespeichert.

Die **memyak.log** Datei wird täglich neu erzeugt und Dateien der vorhergehenden Tage werden nach dem Schema **memyak.yyyy_MM_dd.log** im selben Verzeichnis abgelegt.

Administrator

memYAK wird mit einer Benutzer-Id für den Administrator und zugehörigem Kennwort ausgeliefert. In der Datei **memyakContext.xml** müssen Sie das Kennwort dieses

Administrators ändern. Die Änderung geschieht im Abschnitt **content user directory config**.

```
<!-- content user directory config -->
<bean id="directory"
class="dlab.directory.content.ContentDirectory">
    <constructor-arg ref="repositoryResolver"/>
    <property name="adminUid" value="ADMIN"/>
    <property name="adminPassword" value="secret"/>
</bean>
```

Bitte ändern Sie unbedingt das Kennwort in der Eigenschaft **adminPassword**. Das eingestellte Kennwort wird mit jedem memYAK System ausgeliefert, so dass das Kennwort allen memYAK Administratoren bekannt ist! Sie können auch (müssen aber nicht) die Benutzer-Id in der Eigenschaft **adminUid** ändern.

Der Administrator hat alle Rechte für das installierte memYAK System.

Ports, Absicherung und Dateigröße

Die Sicherheitseigenschaften von memYAK werden in der Datei **memyakContext.xml** im Abschnitt **web config** eingestellt.

```
<!-- web config -->
<bean id="webConfig" class="dlab.web.application.WebConfig">
    <property name="SSLPort" value="443" />
    <property name="publicPort" value="80" />
    <property name="secureAuthenticatedSession" value="false" />
    <property name="secure" value="false" />
    <property name="enforceAuthentication" value="true" />
    <property name="uploadLimitMB" value="100" />
    <!-- the upload limit is in megabytes-->
</bean>
```

1. Ports

Bitte bestimmen Sie die Ports, die der Tomcat für verschlüsselte und unverschlüsselte Kommunikation verwendet. Voreingestellt sind die Standard-Ports. Für SSL Verschlüsselung ist der **SSLPort** mit dem Wert **443** und für unverschlüsselte Kommunikation **publicPort** mit dem Wert **80** konfiguriert.

2. Absicherung

Sie können bestimmen, wann eine verschlüsselte Kommunikation durch memYAK erzwungen wird. In der Voreinstellung wird memYAK keine verschlüsselte Kommunikation erzwingen.

Wenn Sie memYAK im Internet einsetzen, sollten Sie diese Einstellung ändern, da sonst sämtliche Informationen und damit auch alle Anmeldenamen- und Kenn-

worte unverschlüsselt übertragen werden.

Wenn Sie verschlüsselte Kommunikation aktivieren möchten, muss Ihr Tomcat mit einem Zertifikat zur SSL Verschlüsselung ausgestattet sein.

Nachfolgend werden die Sicherheitseigenschaften erläutert:

secureAuthenticatedSession

Wenn diese Eigenschaft den Wert **true** erhält, wird die gesamte Kommunikation zwischen memYAK und den authentifizierten Benutzern verschlüsselt (das schließt den Anmeldevorgang eines Benutzers ein).

secure

Wenn diese Eigenschaft den Wert **true** erhält, wird die gesamte Kommunikation zwischen memYAK und **allen** Benutzern verschlüsselt (das schließt anonyme Benutzer mit ein). Diese Einstellung sollte gewählt werden, wenn memYAK im Internet eingesetzt wird.

enforceAuthentication

Wenn diese Eigenschaft den Wert **true** erhält, erzwingt memYAK eine Anmeldung am System, so dass keine anonymen Benutzer auf memYAK zugreifen können.

3. Begrenzung für das Hochladen von Einzeldateien

Mit der Eigenschaft **uploadLimitMB** begrenzen Sie die Größe von Dateien, die von memYAK verwaltet werden können. Die Größenangabe erfolgt in Megabyte. Die voreingestellte Begrenzung liegt bei **100MB**.

E-Mail

memYAK versendet E-Mails, wenn z.B. Dokumente bearbeitet werden und bestimmte Anwender von dieser Bearbeitung in Kenntnis gesetzt werden sollen. Auch beim Registrieren eines Anwenders oder beim Zurücksetzen des Kennwortes werden E-Mails versendet.

memYAK verwendet einen SMTP Mail-Provider, um diese E-Mails zu versenden. Sie konfigurieren den Zugang zu diesem Provider in der Datei **memyakContext.xml** im Abschnitt **mail config**.

```

<!-- mail config -->
<bean id="mailConfig" class="dlab.mail.MailConfig">
    <property name="host"
        value="... address of your smtp provider ... " />

    <property name="port" value="25" />

    <property name="user"
        value="... complete mail address of your mail user ... " />

    <property name="password"
        value="... password of your mail user ... " />

    <property name="domain" value="... your mail domain ... " />
    <property name="defaultSenderAddress" value="info" />
    <property name="useOnlyDefaultSenderAddress" value="false" />
</bean>

```

1. Host

Unter der Eigenschaft **host** ist die Adresse SMTP-Providers anzugeben, z.B. **smtp.1und1.de**.

2. Port

Die Eigenschaft **port** enthält den zu verwendenden Port. Als Vorgabe ist der Standard-Port **25** eingetragen.

3. E-Mail Anwender

Der E-Mail Anwender wird für die Anmeldung am SMTP Provider verwendet. Unter der Eigenschaft **user** ist der Anwender mit seiner vollständigen E-Mail Adresse und unter der Eigenschaft **password** das Kennwort dieses Anwenders einzutragen.

4. Domain

Die Eigenschaft **domain** enthält den Domännennamen des zuvor angegebenen E-Mail Anwenders, z.B. **datenlabor.net**. Dieser Domännennamen ist ohne "@" einzugeben.

5. E-Mail Absenderadresse für den Versand von E-Mails

Per Vorgabe wird für die Eigenschaft **defaultSenderAddress** die Absenderadresse **info** verwendet. Sie können eine andere Adresse verwenden, indem Sie der Eigenschaft **defaultSenderAddress** einen anderen Wert zuweisen (ohne @ und ohne Ihre Domain).

6. Wenn die gesendeten E-Mails als Absender immer die Vorgabe-Absenderadresse enthalten sollen, setzen Sie den Wert der Eigenschaft **useOnlyDefaultSenderAddress** auf **true**.

MongoDB

An dieser Stelle ist lediglich die Adresse (oder die Adressen) Ihres MongoDB Systems anzugeben. Diese Einstellung wird in der Datei **memyakContext.xml** im Abschnitt **content repository** vorgenommen.

```
<!-- content repository -->
<bean id="contentRepository"
      class="dlab.content.mongo.MongoRepository">
  <constructor-arg ref="repositoryResolver" />
  <constructor-arg ref="currentUserResolver" />
  <constructor-arg>
    <list>
      <bean class="dlab.content.mongo.MongoServer">
        <constructor-arg value="localhost" />
      </bean>
    </list>
  </constructor-arg>

  <!-- the lock timeout: 6 hours -->
  <property name="lockTimeoutInSeconds"
    value="21600"></property>
</bean>
```

Voreingestellt ist in der Bean-Definition **dlab.content.mongo.MongoServer** ein einziger Wert **localhost**. Das bedeutet, Sie verwenden einen MongoDB Server mit dem Standard Port **27017**, der auf dem selben System installiert ist, auf dem Sie auch die memYAK Anwendung (das WAR-File) installieren. Wenn Sie memYAK und MongoDB auf dem selben Rechnersystem betreiben, müssen Sie an dieser Konfiguration nichts ändern.

An Stelle der Adresse **localhost** können Sie auch die IP-Adresse oder den DNS-Namen eines entfernt liegenden MongoDB Servers angeben.

Wenn Sie einen anderen Port verwenden möchten, können Sie ein weiteres Konstruktorsargument angeben. Beispiel:

```
<bean class="dlab.content.mongo.MongoServer">
  <constructor-arg value="mongo.address.net" />
  <constructor-arg value="71072" />
</bean>
```

Wenn Sie memYAK nicht mit einem einzelnen MongoDB Server, sondern mit einem MongoDB Replicaset verbinden möchten, können Sie die Adressen mehrerer MongoDB Server in der Liste zusammenfassen.

Beispiel:

```
<!-- content repository -->
<bean id="contentRepository"
      class="dlab.content.mongo.MongoRepository">
  <constructor-arg ref="repositoryResolver" />
  <constructor-arg ref="currentUserResolver" />
  <constructor-arg>
    <list>
      <bean class="dlab.content.mongo.MongoServer">
        <constructor-arg value="mongo1.server" />
      </bean>

      <bean class="dlab.content.mongo.MongoServer">
        <constructor-arg value="mongo2.server" />
      </bean>

      <bean class="dlab.content.mongo.MongoServer">
        <constructor-arg value="mongo3.server" />
      </bean>
    </list>
  </constructor-arg>

  <!-- the lock timeout: 6 hours -->
  <property name="lockTimeoutInSeconds"
    value="21600"></property>
</bean>
```

Benutzerregistrierung

Die Einstellungen zur Selbstregistrierung und Kennworrücksetzung werden in der Datei **memyakContext.xml** im Abschnitt **navigation config** vorgenommen.

```
<!-- navigation config -->
<bean id="navigationConfig"
      class="dlab.web.navigation.NavigationConfig">

  <property name="allowSelfRegistration" value="false" />
  .
  .
  .
</bean>
```

Die Selbstregistrierung erlaubt Personen, die im memYAK noch nicht bekannt sind, sich als Anwender selbst zu registrieren und damit automatisch einen Personeneintrag im Benutzerverzeichnis zu erstellen.

In der Voreinstellung von memYAK ist diese Funktion deaktiviert. Wenn die Funktion angeboten werden soll, setzen Sie bitte den Wert der Eigenschaft `allowSelfRegistration` auf `true`.

Kennwort-Validator

In der Vorgabeeinstellung müssen die Kennworte, die durch memYAK verwaltet werden, mindestens 5 Zeichen lang sein. Optional kann ein aufwendigerer Kennwort-Validator eingestellt werden, der eine Mindestlänge von 8 Zeichen und kompliziertere Regeln zum geforderten Aufbau der Kennworte enthält.

Die Einstellung wird in der Datei **memyakContext.xml** im Abschnitt **password service** vorgenommen.

Fügen Sie im Abschnitt **passwordService** folgende Eigenschaft ein:

```
<property name="passwordValidator" ref="passwordValidator" />
```

Nun erzeugen Sie direkt unterhalb des Abschnittes **passwordService** eine neue Bean, die wie folgt aussehen muss:

```
<!-- password validator -->
<bean id="passwordValidator"
      class="dlab.security.PasswordValidatorType2" />
```

Sie sollten dann das folgende Bild sehen:

```
<!-- password service -->
<bean id="passwordService" class="dlab.security.PasswordService">
  <property name="directoryService" ref="directory" />
  <property name="mailService" ref="mailService" />
  <property name="supportContact" ref="supportContact" />
  <property name="passwordReservationService" ref="passwordReservationService" />
  <property name="passwordValidator" ref="passwordValidator" />
</bean>

<!-- password validator -->
<bean id="passwordValidator"
      class="dlab.security.PasswordValidatorType2" />
```

Mit dieser Einstellung und nach einem Neustart des Tomcat Servers ist der Kennwort-Validator aktiviert.

Support Information

Sie können den Support Kontakt selbst bestimmen (z.B. mit den Kontaktdaten Ihres Helpdesk). Die Support Kontaktdaten werden den Anwendern zum Beispiel im Anmeldebildschirm und bei bestimmten Fehlermeldungen angezeigt.

Die Einstellung wird in der Datei **memyakContext.xml** im Abschnitt **support contact** vorgenommen.

```

<!-- support contact -->
<bean id="supportContact" class="dlab.util.nameandaddress.NameAndAd-
dress">
    <property name="companyName" value="Datenlabor GmbH" />
    <property name="contactUrl" value="http://datenlabor.net" />
    <property name="mailAddress" value="info@datenlabor.net" />
</bean>

```

Folgende Eigenschaften sind konfigurierbar:

**building, city, cityZip, companyName, contactUrl, country, fax, floor, given-
Name, info, jobTitle, mailAddress, mobilePhone, orgUnit, phone, postBox,
postBoxCity, postBoxZip, room.**

Text im Anmeldebildschirm

Sie können eine Text, wie z.B. Ihren Firmennamen, im Anmeldebildschirm anzeigen.

Diese Konfiguration wird im Abschnitt **login html** vorgenommen.

```

<!-- login html -->
<bean id="loginHtml" class="java.lang.String">
    <constructor-arg>
        <value>
            <![CDATA[<div style="position:fixed;
text-align:right; font-size:300px; line-height:1em;
color:#eee; right:-60px; bottom:-75px;
z-index:-99;">memYAK</div>]]>
        </value>
    </constructor-arg>
</bean>

```

Master-Instanz

Optional kann definiert werden, ob eine memYAK-Instanz als Master betrieben wird.

Ein Master importiert bei jedem Start der Anwendung die Template Dokumentation in das MongoDB-System. In der Vorgabeeinstellung ist jede memYAK-Instanz ein Master. Wenn mehrere memYAK-Instanzen dasselbe MongoDB System verwenden, sollte nur eine einzige memYAK-Instanz als Master betrieben werden, damit die Dokumentation kontrolliert nur von dieser Instanz in das MongoDB-System importiert wird.

Nehmen Sie zur Konfiguration im Bereich **applicationConfig** den folgenden Eintrag vor:

Die Eigenschaft **master** muss den Wert **false** erhalten, damit die aktuelle Instanz kein Master ist. Wenn Sie den Wert auf **true** setzen, oder ganz weglassen, ist die aktuelle Instanz der Master.

```
<!-- application config start -->
<bean id="applicationConfig" class="dlab.web.application.Applica-
tionConfig">

    <property name="master" value="false" />
    .
    .
</bean>
```

Nutzung von Google Maps

Beginnend mit der memYAK-Version 2.6 können bei der Nutzung von Adressformularen Karteninformationen durch die Google Maps API bereitgestellt werden. Sie sehen Dann zum Beispiel in den Profildokumenten der registrierten Benutzer eine Karte, in der die Adresse des Benutzers dargestellt wird.

Um die Karten darzustellen, sind im Bereich **applicationConfig** folgende Einträge vorzunehmen:

Die Eigenschaft **allowMaps** muss den Wert **true** erhalten und in der Eigenschaft **googleMapsAPIKey** muss Ihr **API-Schlüssel** angegeben werden.

Bitte beachten Sie, dass Sie die Google Maps nicht ohne diesen Schlüssel verwenden dürfen.

```
<!-- application config start -->
<bean id="applicationConfig" class="dlab.web.application.Applica-
tionConfig">

    <property name="allowMaps" value="true" />
    <property name="googleMapsAPIKey" value=",,, your google maps
api key ..." />
    .
    .
    .
</bean>
```

Tomcat Einstellungen

1. Machen Sie die memYAK Konfiguration (die Dateien **memyakContext.xml** und **memyakLogging.xml**) dem Tomcat classpath bekannt. Gehen Sie dazu wie folgt vor:

Wenn Sie die Konfiguration z.B. im Verzeichnis **\${catalina.home}/appconfig/memyak** ablegen (das bedeutet Sie speichern die Dateien **memyakContext.xml** und **memyakLogging.xml** in diesem Verzeichnis), fügen Sie in der Datei **catalina.properties** für die Property **shared.loader** den Pfad **\${catalina.home}/appconfig/memyak** hinzu.

Wenn Sie das Verzeichnis eingerichtet und die **catalina.properties** entsprechend angepasst haben, muss der Tomcat neu gestartet werden.

Durch die Speicherung der Konfiguration in einem separaten Verzeichnis können Sie zu späteren Zeitpunkten neue memYAK Versionen installieren, ohne jedes Mal die Konfiguration für Ihre Serverumgebung neu durchführen zu müssen.

2. Das **URIEncoding** muss für alle **Connector** Einträge in der Datei **server.xml** auf **UTF-8** eingestellt werden. Diese Einstellung wird benötigt, damit AJAX-Requests bei der Übermittlung von Sonderzeichen, wie z.B. deutschen Umlauten, richtig codiert sind.

3. Erzeugen Sie die Datei **setenv.sh** oder **setenv.bat** im Verzeichnis **\${catalina.home}/bin**

Hier sollte der **Heapsize** des Tomcat auf **mindestens 1024 MB** eingestellt werden.

Beispiel:

```
export CATALINA_OPTS=„$CATALINA_OPTS -Xms1024m -Xmx1024m“
```

setzt den initialen und maximalen Heap auf 1024 MB. Dabei ist die allgemeine Form:

```
-Xms<initial heap size> -Xmx<maximum heap size>
```

Weiterhin muss Tomcat auf Linux Systemen **unabhängig vom X-Window System** gemacht werden. Das wird durch folgende Konfiguration in der **setenv.sh** erreicht:

```
export CATALINA_OPTS=„$CATALINA_OPTS -server -Djava.awt.headless=true“
```

Wenn Sie alle Einstellungen vorgenommen haben, muss der Tomcat neu gestartet werden.

Linux Einstellungen

Die maximal erlaubte Anzahl geöffneter Dateien und die Anzahl der File-Handles sollte nicht zu gering gesetzt sein. Bitte vergleichen Sie hierzu <http://docs.mongodb.org/manual/reference/ulimit> .

Deployment

Nachdem alle Konfigurationsschritte vorgenommen wurden und die Konfiguration dem Tomcat bekannt gemacht ist (vgl. dazu den Abschnitt Tomcat Einstellungen), kann memYAK deployt werden.

Benennen Sie dazu die Datei **memyak_[version].war** um in **[kontext].war**. **[Kontext]** steht dabei für den Namen des Kontext, unter dem memYAK deployt werden soll. Wenn Sie memYAK im Root-Context (/) des Tomcat Servers deployen möchten, muss die Datei **[kontext].war** in **ROOT.war** benannt werden.

Nun können Sie das WAR-File z.B. mit dem Tomcat Manager deployen.

Erste Anmeldung

Verwenden Sie für die ersten Anmeldung die Administratorkennung, die Sie weiter oben festgelegt haben (vergleichen Sie dazu den Abschnitt **Administrator**).

Wenn Sie sich angemeldet haben, können Sie weitere Benutzer registrieren und neue Arbeitsbereiche anlegen. Insbesondere sollten Sie für sich selbst eine Benutzer-Id registrieren, mit der Sie die tägliche Arbeit im memYAK verrichten. Der Administrationsbenutzer, mit dem Sie sich zum ersten Mal anmelden, sollte nur zum Zweck der ersten Anmeldung und als Rückversicherung verwendet werden.

Wenn Sie für sich selbst eine Benutzer-Id erzeugt haben, können Sie den Administrationsbenutzer verwenden, um ihrer persönlichen Benutzer-Id ebenfalls Administratorenrechte zuzuweisen. Dazu melden Sie sich mit der Administratorkennung an, öffnen über das Benutzerverzeichnis das Benutzerprofil ihres neu angelegten persönlichen Benutzers und aktivieren unter **Sicherheit** Option **Administratorenberechtigung**.

Neue Versionen installieren

Nach der Installation einer neuen memYAK Version muss der Tomcat Server immer gestoppt und wieder neu gestartet werden.

Backup

memYAK speichert sämtliche Daten im MongoDB System. Jeder memYAK Arbeitsbereich ist vollständig durch eine Mongo Datenbank abgebildet, die denselben Namen trägt, wie der memYAK Arbeitsbereich.

memYAK legt automatisch die folgenden MongoDB Datenbanken an:

- AdministrationRequests (für Kennwortrücksetzungen)
- UserProfiles (zur Speicherung der Anmeldedaten und Benutzerprofile)

Für ein Backup einzelner MongoDB Datenbanken oder des gesamten MongoDB Systems wird auf <http://docs.mongodb.org/manual/core/backups> verwiesen.

Die einmalige Sicherung der Dateien **memyakContext.xml** und **memyakLogging.xml** sowie der Dateien, die Sie für die Tomcat Konfiguration angepasst haben, wird empfohlen.