# Inhaltsverzeichnis

[1. Inhaltsverzeichnis 1](#_Toc425186592)

[2. Projekte in Eclipse einbinden 2](#_Toc425186593)

[3. Lokale Testumgebung 3](#_Toc425186594)

[3.1. VM installieren & konfigurieren 3](#_Toc425186595)

[3.2. Karaf installieren 4](#_Toc425186596)

[3.2.1. Pakete nachinstallieren 4](#_Toc425186597)

[3.3. Karaf ausführen 4](#_Toc425186598)

[3.3.1. ActiveMQ starten 4](#_Toc425186599)

[3.3.2. Karaf starten 4](#_Toc425186600)

[3.3.3. Karaf module deployen 5](#_Toc425186601)

[3.3.4. In Karaf einloggen 5](#_Toc425186602)

[3.3.5. Karaf Befehle allgemein 5](#_Toc425186603)

[3.3.6. Karaf Befehle für Camel 5](#_Toc425186604)

[3.3.7. Logs 5](#_Toc425186605)

[3.4. HeidiSQL reintunneln 5](#_Toc425186606)

[3.5. tcp Connection auslesen 6](#_Toc425186607)

[3.5.1. tcpflow installieren & konfigurieren 6](#_Toc425186608)

[3.5.2. tcpflow starten 6](#_Toc425186609)

[3.5.3. tcpflow analysieren 6](#_Toc425186610)

[4. Betrieb 7](#_Toc425186611)

[4.1. ActiveMQ 7](#_Toc425186612)

[4.1.1. Queue ActiveMQ.DLQ 7](#_Toc425186613)

# Projekte in Eclipse einbinden

1. Neues Projekt > Other > SVN
   1. Auswahl svn-Pfad: …/trunk
   2. Dort alle Unterverzeichnisse
2. Für jedes Projekt: als Maven konfigurieren
   1. Rechtsklick auf Projekt > Configure > Convert to Maven Project
3. Für jedes Projekt: JDK 1.6 wählen
   1. Rechtsklick auf JRE System Library > Alternate JRE > Installed JREs > Add > Standard VM > Directory > Pfad zum JDK (z.B. jdk1.6.0\_37)
   2. Auf JDK 1.6 wechseln
   3. Als Workspace default wählen
   4. Ab dem zweiten Projekt kann direkt der Workspace default gewählt werden
4. Für jedes Projekt: Build
   1. Rechtsklick auf Projekt > Run As > Maven build
   2. Goals: „clean install“
   3. Auf „BUILD SUCCESS“ warten
5. Für Schema-Projekte (bei denen CXF auf Basis von XSDs Sourcen baut):
   1. Ggf. zuerst F5 (Projekt aktualisieren)
   2. Rechtsklick auf Projekt > New > Source Folder
   3. Folder name: Browse > target > generated-sources > xjc
   4. Im Projekt sollten diese nun eingebunden sein

# Lokale Testumgebung

## VM installieren & konfigurieren

1. VirtualBox installieren

Image der Karaf VM einbinden

1. root und MySQL Zugangsdaten siehe Keepass
2. In der Konsole eth konfigurieren (als root)
   1. nano /etc/network/interfaces: alle Vorkommen von eth5 auf eth6 ändern
   2. /etc/init.d/networking restart
3. Portweiterleitung
   1. Maschine ändern > Netzwerk > Port-Weiterleitung:
   2. 
   3. Die Gast-IP erfährt man mit ifconfig -a im Gastsystem
   4. MobaXterm: auf 127.0.0.1 verbinden
4. User anlegen (als root)
   1. useradd mmeier
   2. passwd mmeier
   3. mkdir /home/mmeier

chown mmeier:mmeier /home/mmeier

* 1. Bash als Shell festlegen

nano /etc/passwd

* 1. sudo-Rechte anlegen:

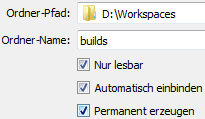
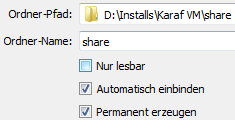
nano /etc/sudoers

mmeier ALL=NOPASSWD: ALL

1. Lokale Verzeichnisse einbinden
   1. Gasterweiterungen installieren (als root)

<http://virtualboxes.org/doc/installing-guest-additions-on-debian/>

* 1. Machine ändern > Gemeinsame Ordner > Ordner hinzufügen

* 1. Verzeichnisse anlegen (als root):

mkdir /mnt/builds

mkdir /mnt/share

* 1. Mounten (als mmeier) (dazu sind sudo-Rechte nötig – s.o.)

sudo mount -t vboxsf builds /mnt/builds

sudo mount -t vboxsf share /mnt/share

* 1. Bash konfigurieren

.bashrc und .bash\_profile vom share nach ~ kopieren

## Karaf installieren

### Pakete nachinstallieren

* In Karaf-Konsole einloggen
* install wrap:mvn:<groupId>/<artifactId>/<version>
* groupId, artifactId und version sieht man z.B. auf dem Archiva
* Falls Meldung erscheint: „Execution Environment: Java SE-1.6 not supported“
  + 1. In zugehöriges bundle.jar gehen (id siehe karaf-Konsole):

apache-karaf-2.2.4-ecotel/data/cache/bundle<id>/version0.0/bundle.jar

* + 1. In META-INF/MANIFEST.MF die Zeile mit JavaSE-1.6 entfernen

Für Vodafone sind dies die Pakete:

* install wrap:mvn:commons-configuration/commons-configuration/1.6
* install wrap:mvn:commons-beanutils/commons-beanutils/1.6
* install wrap:mvn:commons-digester/commons-digester/1.6
* install wrap:mvn:commons-jxpath/commons-jxpath/1.2
* install wrap:mvn:commons-net/commons-net/3.2
* install wrap:mvn:bcpg/bcpg-jdk16-146/1.6-146
  + Java SE-1.6 entfernen
* install wrap:mvn:org.bouncycastle/bcprov-ext-jdk16/1.46
  + Java SE-1.6 entfernen

## Karaf ausführen

### ActiveMQ starten

~/apache-activemq-5.8.0/bin/activemq start

### Karaf starten

~/apache-karaf-2.2.4-ecotel/bin/start

### Karaf module deployen

* Ins Verzeichnis kopieren:

~/apache-karaf-2.2.4/deploy

### In Karaf einloggen

* ssh -l karaf -p 8101 localhost

### Karaf Befehle allgemein

* list
* start <id>
* stop <id>
* restart <id>
* logout (bzw. Ctrl+D)
* refresh

### Karaf Befehle für Camel

<http://camel.apache.org/karaf.html>

|  |  |
| --- | --- |
| **Command** | Description |
| **camel:list-contexts** | Lists the camel contexts available in the current Karaf instance |
| **camel:list-routes** | Displays the list of Camel routes available in the current Karaf instance |
| **camel:info-context** | Displays detail information about a given Camel context |
| **camel:start-context** | Starts the given Camel context |
| **camel:stop-context** | Stops the given Camel context |
| **camel:info-route** | Provides detail information about a Camel route |
| **camel:show-route** | Renders the route in XML |
| **camel:start-route** | Starts the given route. |
| **camel:stop-route** | Stops the given route. |

### Logs

~/apache-karaf-2.2.4/data/log

/data/log

## HeidiSQL reintunneln

1. Portweiterleitung auf der VirtualBox



1. HeidiSQL-Konfiguration

Verbindungstyp: MySQL (TCP/IP)

Hostname / IP: 127.0.0.1

Username + PW: s. Keepass (Karaf MySQL)

Port: 3307

*Bemerkung: Port 3307 wird verwendet, da die lokale MySQL den Port 3306 bereits benutzt!*

## tcp Connection auslesen

### tcpflow installieren & konfigurieren

* sudo apt-get install tcpflow

### tcpflow starten

* sudo ifconfig
* sudo tcpflow -i ethX
  + ethX ergibt sich dabei aus ifconfig
* Test durchführen
* tcpflow beenden

### tcpflow analysieren

1. Im aktuellen Verzeichnis legt tcpflow Dateien an:
   * Quell-IP Quell-Port Ziel-IP Ziel-Port
2. Aus der gewünschten Datei den XML-String extrahieren
3. xmlstarlet fo
4. String pasten, Strg+D

# Betrieb

## ActiveMQ

<http://192.168.1.77:8161/admin>

User: admin

Passwort: admin

### Queue ActiveMQ.DLQ

* Standard DeadLetterQueue von ActiveMQ. Dorthinein werden Elemente geschrieben, für die es keinen Listener gibt.