

DataMatrix

- [Vorbereitung:](#)
- [Datamatrix einbinden über das Image-Objekt](#)
- [Datenquelle erstellen](#)

Ziel in diesem Artikel ist es, mit Hilfe der Java Library [Zxing](#), ein `java.awt.Image` einer gültigen und maschinenlesbaren Datamatrix zu erzeugen. Zxing stellt schon die funktionstüchtige Klasse `DatamatrixWriter` zur Verfügung, diese ist nutzbar für den Hausgebrauch und einfachen Handyscannern, jedoch müssen für die Erzeugung eines gültigen Codes nach GS1-Standard ein paar Kleinigkeiten im Source-Code der Zxing-Library eingefügt werden.

Die Datamatrix ist eine zweidimensionale Abbildung von Information. Jedes Zeichen der Information ist maschinennah in Byte bzw. 8 Bits, gesetzt schwarz, nicht gesetzt weiß, abgebildet. Die Zeichen werden zusammengefasst zu Code-Wörtern, welche, von einer zu Anfang jedes Code-Wortes einleitender, 2-3-stelliger Zeichenfolge identifiziert werden, den sogenannten "Application Identifier". Der Application Identifier gibt an, welche Daten nach der GS1-Spezifikation folgen werden.

Eine gültige Matrix enthält das Codeword FNC1 am Anfang und vor jedem Codeword mit variabler Länge (abhängig vom Application Identifier). Jedoch ist diese FNC1 nicht in den zu verschlüsselnden Daten enthalten und auch nicht in der lesbaren Ausgabe.

Vorbereitung:

Folgende Jars sind notwendig:



Für die Verwendung von Datamatrix über Jasper legt man folgenden Parameter in Nuclios an :

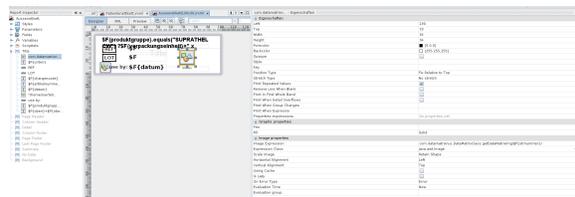
```
nuclios.jasper.reports.compile.classpath.jars.byclasses
```

mit dem Wert:

```
net.sf.jasperreports.engine.JasperReport org.nuclios.server.report.api.  
JRNucliosDataSource com.google.zxing.BarcodeFormat com.google.  
zxing.WriterException com.google.zxing.client.j2se.MatrixToImageWriter  
com.google.zxing.common.BitMatrix com.datamatrixnuc.  
DataMatrixClass
```



Datamatrix einbinden über das Image-Objekt

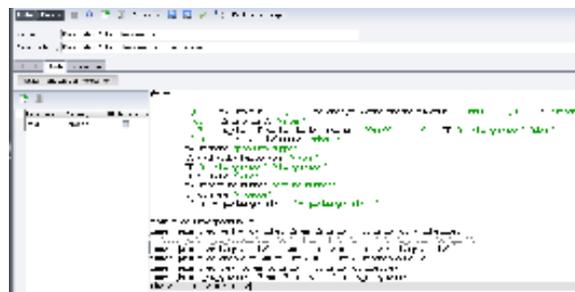


Datenquelle erstellen

Für unten genanntes Beispiel sind hier die "Application Identifier" z.B. (01).

Dies muss individuell je nach Projekt dargestellt werden.

```
'(01)||t2.intgtin13||(17)||to_char(t5.dathaltbarkeitsdatum, 'YYmm00')  
||(10)||T5."strchargencode"||(240)||t2.strartikelnummer "strnummer",
```



Einen Ordner 'extensions/server' anlegen

Dazu zu muss im Vorfeld im Installationspfad von Nuclos ein Ordner "**extensions**" angelegt werden. In Abhängigkeit der darin enthaltenen Funktionalität enthält der Ordner seine eigene Struktur (Unterordner) *client* für clientseitig, *server* für serverseitig, *common* für beides.

Die barbecue-1.5-beta1.jar , **sowie die abhängige jdom.jar** werden nach `~/<Installationspfad-Nuclos>/extensions/server/` geschoben. Das Einbinden der Jasperreport-6.17.0.jar ist nicht notwendig, da dieses Archiv in Nuclos bereits verwendet wird.

Installer ausführen

Nachdem die Extensions-Struktur angelegt und alle Jar's den selben Key besitzen, **muss auf die bestehende Installation ein Update ausgeführt werden**, indem der Installer einfach noch einmal ausgeführt wird. Nach der Installation müssen sich die Jars im Ordner "webapp /WEB-INF/lib" befinden. Soll der Installer nicht gleich wieder ausgeführt werden, reicht das manuelle Einfügen der Jars in den zuletzt genannten Ordner und ein Server-Neustart aus.