

Installationsanleitung / Technische Dokumentation - DicomViewer

Inhaltsverzeichnis

Einleitung	1
Features des DicomViewer	1
Installationsanleitung.....	2
Softwareaufbau.....	4
Schichten.....	4
Verzeichnis-Struktur	5
Beschreibung der einzelnen Dateien.....	6
Kritik / Verbesserungsvorschläge.....	8
Verwendete Programme / Bibliotheken	9
Genauere Versionsnummern.....	9
Beschreibung der Programme / Bibliotheken.....	9

Einleitung

Als Beispielanwendung wurde ein webbasierter DicomViewer auf PHP-Basis erstellt, da dies die einleuchtendste Beispielanwendung ist. Aufgrund der bereits enormen Anzahl konventioneller DicomViewer für alle gängigen Betriebssysteme, haben wir uns für ein webbasiertes Projekt entschieden. Für das Verwenden unseres DicomViewers ist keine Installation notwendig, es wird lediglich ein moderner Browser und eine Internetverbindung benötigt.

Der Viewer kann unter <http://test.ragersweb.de/dicom> erreicht werden. Für die Nutzung empfehlen wir einen Browser auf Webkit Basis (Chrome oder Safari). Der Viewer ist mit einer für Touchscreens optimierten Oberfläche auf Basis von jQuery Mobile ausgestattet, dadurch kann er auch mit Tablet-Computer ideal genutzt werden. Eine mögliche Nutzung des Viewers wäre die einfache webbasierte Weitergabe von Dicom-Daten an den Patienten oder eine medizinische Einrichtung. So könnten anstatt der bisher üblichen CDs einfach ein passwortgeschützter Link an den Patienten oder den Arzt weitergegeben werden. Alle zu dem Fall gehörenden Dicom-Daten könnten dann einfach über den Browser gesichtet werden. Bei unserem DicomViewer handelt es sich um einen ersten „Proof of Concept“. Er ist noch nicht mit allen notwendigen Features ausgestattet. Trotzdem lässt sich schon in dieser frühen Phase erkennen, welches Potential in webbasierten Systemen für den medizinischen Einsatz stecken.

Features des DicomViewer

- Webbasierter DicomViewer mit touchscreenoptimierter Oberfläche
- Laden von Dicom-Dateien
- Parsen und Anzeigen von beliebigen Dicom-Attributen
- Live-Volltextsuche über alle Attribute
- Anzeigen von Dicom-Bildern in den Formaten JPEG, JPEG2000, JPEGLL, TIFF
- Clientseitiges Anpassen von Gamma- und Helligkeitswerten sowie Invertieren des Bildes

Installationsanleitung

Es wird davon ausgegangen, dass bereits ein Webserver mit PHP auf dem System installiert ist. Eine Anleitung für Ubuntu finden sie z.B. unter http://www.ragersweb.de/coding/php/ubuntu_als_webserver_einrichten.html. Für die Installation des DicomViewers ist Image-Magick zwingend notwendig. Auf Debian basierten Systemen kann es über den Befehl:

```
aptitude update && aptitude install imagemagick && aptitude install php5-imagick
```

installiert werden. Zur Installation werden Root-Rechte benötigt. Je nach System muss eventuell der Apache neu gestartet werden. Dies kann über

```
/etc/init.d/apache2 restart
```

geschehen. Nach dem Neustart kann über eine „Info.php“ (Name der Datei ist egal) getestet werden, ob Image-Magick korrekt installiert wurde. Dazu muss einfach eine PHP Datei mit folgendem Inhalt auf dem Webserver abgelegt werden.

```
<?php phpinfo() ?>
```

Wird diese Datei dann über den Browser aufgerufen, werden alle PHP-Einstellungen aufgelistet. Unter dem Punkt „Imagick“ sollte dann folgendes aufgelistet sein:

imagemagick

imagemagick module	enabled
imagemagick module version	2.1.1-rc1
imagemagick classes	ImageMagick, ImageMagickDraw, ImageMagickPixel, ImageMagickPixelIterator
ImageMagick version	ImageMagick 6.5.1-0 2010-12-02 Q16 OpenMP http://www.imagemagick.org
ImageMagick copyright	Copyright (C) 1999-2009 ImageMagick Studio LLC
ImageMagick release date	2010-12-02
ImageMagick Number of supported formats:	198
ImageMagick Supported formats	A, AI, ART, ARW, AVI, AVS, B, BGR, BMP, BMP2, BMP3, BRG, BRG, C, CAPTION, CIN, CIP, CLIP, CMYK, CMYKA, CR2, CRW, CUR, CUT, DCM, DCR, DCX, DDS, DFONT, DJVU, DNG, DOT, DPS, DPX, EPDF, EPI, EPS, EPS2, EPS3, EPSF, EPSI, EPT, EPT2, EPT3, ERF, EXR, FAX, FITS, FRACTAL, FTS, G, G3, GBR, GIF, GIF87, GRADIENT, GRAY, GRB, HISTOGRAM, HRZ, HTM, HTML, ICB, ICO, ICON, INFO, INLINE, IPL, ISOBRL, JNG, JP2, JPC, JPEG, JPG, JPC, JPC, K, K25, KDC, LABEL, M, M2V, M4V, MAP, MAT, MATTE, MIFF, MNG, MONO, MOV, MP4, MPC, MPEG, MPG, MRW, MSL, MSVG, MTV, MVG, NEF, NULL, O, ORF, OTB, OTF, PAL, PALM, PAM, PATTERN, PBM, PCD, PCDS, PCL, PCT, PCX, PDB, PDF, PDFA, PEF, PFA, PFB, PFM, PGM, PGX, PICON, PICT, PIX, PJPEG, PLASMA, PNG, PNG24, PNG32, PNG8, PNM, PPM, PREVIEW, PS, PS2, PS3, PSD, PTIF, PWP, R, RADIAL-GRADIENT, RAF, RAS, RBG, RGB, RGBA, RGBO, RLA, RLE, SCR, SCT, SFW, SGI, SHTML, SR2, SRF, STEGANO, SUN, SVG, SVGZ, TEXT, TGA, THUMBNAIL, TIFF, TIFF64, TILE, TIM, TTC, TTF, TXT, UBRL, UIL, UYVY, VDA, VICAR, VID, VIFF, VST, WBMP, WMF, WMV, WMZ, WPG, X, X3F, XBM, XC, XCF, XPM, XPS, XV, XWD, Y, YCbCr, YCbCrA, YUV

Directive	Local Value	Master Value
imagemagick.locale_fix	0	0

Abbildung 1: Auszug aus der info.php

Wichtig ist hier, dass das Image-Magick aktiviert ist und bei „ImageMagick Supported formats“ die Formate PNG, TIFF, JPEG, JPG und vor allem DCM vorhanden sind.

Ist die Installation von Image-Magick erfolgreich abgeschlossen, kann der DicomViewer per FTP oder SSH auf den Server übertragen werden. Nach der Übertragung der Daten muss darauf geachtet werden, dass alle Dateien die benötigten Dateiberechtigungen haben (644 → rw-r--r--). Wurden die Berechtigungen gesetzt, sollte der DicomViewer unter dem angelegten Verzeichnis per Browser erreichbar sein.

Hinweise:

Bei der eingesetzten PHP-Version und dem eingesetzten Webserver sind auch andere Versionen/Programme einsetzbar. Allerdings wurde die Software ausschließlich unter PHP5 mit dem Apache2 getestet. Der Einsatz von alternativen Webservern wie Nginx oder Lighttpd ist ebenfalls möglich, wurde aber nicht getestet.

Softwareaufbau*Schichten*

Die Komponenten des DicomViewer sind in insgesamt sieben Schichten aufteilbar. Dabei sind die Schichten eins bis vier auf dem Server und die Schichten fünf bis sieben auf dem Client.

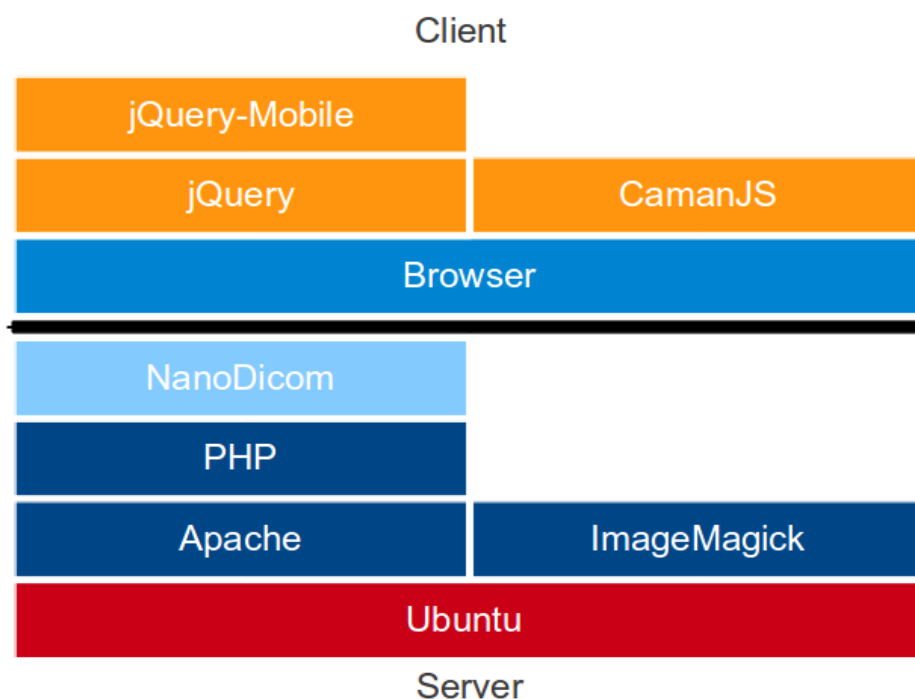


Abbildung 2: Schichten-Modell

Die unterste Schicht bildet das eingesetzte Betriebssystem, auf dem dann der Webserver läuft. In der von uns eingesetzten Konfiguration verwenden wir das Apache-Modul mod_php, das eine performantere Einbindung von PHP erlaubt, als wenn auf fastCGI gesetzt wird. Es ist auch der Einsatz von fastCGI möglich, was PHP dann auf die selbe Schicht wie den Apache bringen würde. Für den Zugriff auf Image-Magick wird die entsprechende Image-Magick-PECL-Extension verwendet. Zwar ist es auch möglich, Image-Magick von PHP aus über die Konsole zu bedienen (exec), allerdings werden dabei keine Nebenläufigkeiten bei parallelen Zugriffen abgefangen. Diese müssen dann

vom Programmierer selbst behandelt werden um sicherzustellen, dass immer die richtigen Bilder ausgeliefert werden. Ein weiterer Vorteil der PECL-Extension ist, dass die erzeugten Bild-Daten nicht erst auf die Festplatte geschrieben werden müssen, sondern dass diese direkt als Stream von PHP verarbeitet werden können. NanoDicom ist ein Dicom-Parser/Writer, der unter PHP läuft.

Auf der Client-Seite ist wie bereits beschrieben lediglich ein moderner Web-Browser erforderlich. Dieser dient als Thin-Client für die JavaScript-Bibliotheken jQuery und CamanJs. jQuery-Mobile ist eine Extension für jQuery, die für die Umsetzung der Oberfläche zuständig ist.

Verzeichnis-Struktur

- **dicomViewer**
 - **classes** - Ordner für zusätzliche Klassen (momentan nicht in Verwendung)
 - **css** - Ordner für CSS (momentan nicht in Verwendung)
 - **dicom** - Ordner mit den Dicom-Dateien (Bildern)
 - **images** - Ordner für Bilder des User-Interfaces (momentan nicht in Verwendung)
 - **js** - Ordner für Javascript
 - **images** - Standard-Bilder von jQuery Mobile
 - **caman.full.min.js** - Javascriptbibliothek zur Bildbearbeitung
 - **dicom.js** - Javascript für DicomViewer
 - **jquery.mobile-1.0b1.min.js** - Javascriptbibliothek GUI
 - **jquery.mobile-1.0b1.min.css** - CSS für GUI
 - **jquery-1.6.1.min.js** - Javascriptbibliothek
 - **nanodicom** - NanoDicom
 - **dict** - Dictionary für Dicom-Attribut-Namen usw.
 - **tools** - Tools von NanoDicom → erben von NanoDicom-Core-Klasse
 - **file_view.php** - Hauptansicht des Viewers
 - **image.php** - Rendert das Bild als PNG
 - **index.php** - Startseite mit „Bild öffnen“-Button
 - **info_dialog.php** - Impressum
 - **nanodicom.php** - Include für file_view (so von NanoDicom vorgegeben)
 - **open_dialog.php** - Öffnen Ansicht für die Bilder

Beschreibung der einzelnen Dateien

In jeder Datei werden die benötigten komprimierten Javascript-Dateien eingebunden.

index.php

Die index.php ist der Einstiegspunkt in die Anwendung. Hier wird nur das Layout der Startpage definiert. Bei einem Klick auf „Datei öffnen“ wird per Ajax zur Seite open_dialog.php gewechselt. Es ist kein PHP-Code vorhanden.

open_dialog.php

In der Datei ist der Bild-Öffnen-Dialog umgesetzt. Die File wird per Ajax aus der index.php oder der file_view.php geladen. Dabei werden alle Dateien und Verzeichnisse beim ersten Laden der Seite ausgelesen. Das heißt, nach einem Klick auf einen Ordner ist kein erneuter Ajax-Call notwendig, um die Dateien zu laden. Die Funktion format_bytes(\$size) sorgt für eine ansprechend formatierte Ausgabe der Dateigröße. Die Dateien werden über die Funktion list_dir(\$startpath) rekursiv ausgelesen und ausgegeben. Für die Dateiliste wird eine mit jQuery Mobile formatierte Liste verwendet.

Bei einem Klick auf eine Datei wird der Pfad der Datei an die file_view.php weitergegeben. Dies kann nicht per Ajax geschehen, da es mit CamanJs bei einem erneuten Ajax-Call nicht mehr richtig initialisiert würde.

file_view.php

Das file_view.php implementiert die Hauptansicht. Auf der linken Seite werden alle Attribute des Dicom-Images angezeigt. Auf der rechten Seite sieht man das Bild.

Die Attribute werden in eine jQuery Mobile Liste geschrieben. Die Dicom-Dumper-Instanz wird über eine Factory geholt. Anschließend wird die Liste mit dem Inhalt aus dem NanoDicom-Dumper befüllt, der wiederum ein spezielles Array mit Format-Informationen übergeben bekommen hat. Durch die Angabe des HTML-Attributs data-filter="true" bei der Liste wird jQuery Mobile angewiesen, den Listeninhalt filterbar zu machen. Dabei wird die gesamte Liste auf dem Client live durchsucht. Es ist kein Klick auf einen Suchen-Button notwendig.

Das Bild wird auf dem Server über die image.php in eine PNG gewandelt. Danach wird es auf der Client-Seite über CamanJs in ein Canvas konvertiert. Dies ist notwendig, um die Helligkeit und den Gamma-Wert im Browser per Javascript anpassbar zu machen. Unter dem Bild werden noch zwei Slider und ein Switch-Button für die Invertierung ausgegeben. Die Events und der notwendige Javascript-Code für die Bildanpassungen befinden sich in der dicom.js.

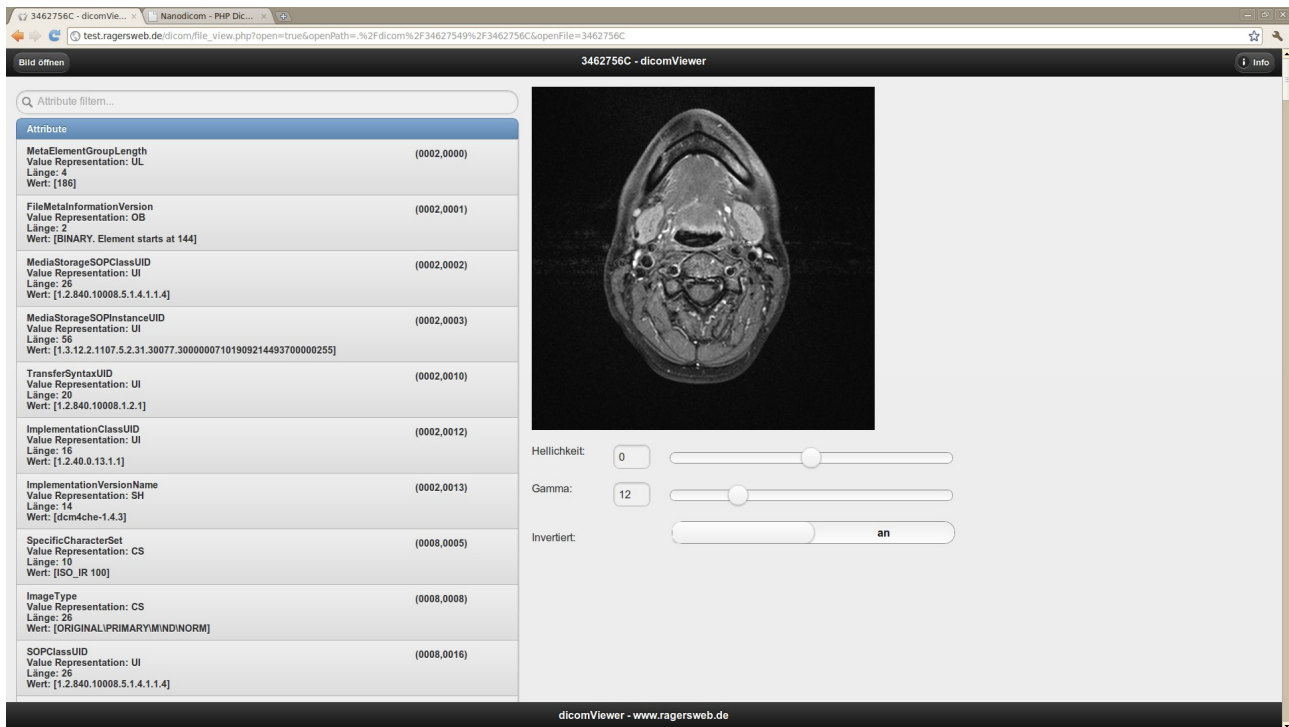


Abbildung 3: Hauptseite des DicomViewers

image.php

Das image.php konvertiert das Dicom-Image in eine PNG und sendet sie an den Browser. Der übergebene Pfad wird auf „..“ gefiltert, damit das Verzeichnis nicht verlassen werden kann. Zum Konvertieren wird eine Image-Magick-Objekt erzeugt und bei der Ausgabe ein entsprechender PNG-Header gesendet.

info_dialog.php

Im info_dialog.php steht das Impressum und Informationen über den Viewer. Es ist kein relevanter Code enthalten.

dicom.js

In einem Intervall von einer Sekunde wird die Funktion renderImage() aufgerufen. Über diese Funktion wird das Bild angepasst. Wenn sich einer der Werte der Slider geändert hat, werden die am Bild vorgenommenen Änderungen zurückgenommen und mit den neuen Werten erneut ausgeführt. Dabei wird erst der Gamma-Wert und dann die Helligkeit angepasst. Ist der InvertModus aktiviert, wird das Bild zusätzlich invertiert.

Es werden bisher nur Standard-Funktionen von CamanJs verwendet. Es ist auch möglich, eigene Filter zu implementieren.

Es wurde für diese Funktion bewusst auf Events verzichtet, da es durch zu schnelles Ziehen an den Slidern zu Fehlberechnungen bei der Bildanpassung kam, wenn die vorherige Berechnung noch nicht abgeschlossen war.

Kritik / Verbesserungsvorschläge

Wie in der Einleitung bereits beschrieben handelt es sich bei unserer Anwendung um einen ersten „Proof of Concept“. Für den produktiven Einsatz wäre es sinnvoll den Viewer um folgende Features zu erweitern.

- **Session-Management für passwortgeschützte Anmeldung**
- **Speichern der Dateipfade in der Session**

Momentan werden die Pfade zu den Bildern noch per Get-Parameter in der URL übertragen. Zwar wird der Pfad gefiltert, so dass ein Zugriff auf Dateien außerhalb des Dicom-Ordners nicht möglich ist, aber es ist leichtsinnig, die Pfadangaben im Klartext von einer Seite an die nächste zu übergeben, da diese so vom User manipuliert werden können. Werden die Pfade hingegen in der Session gespeichert, ist kein Zugriff von außen mehr möglich.
- **Kontrastfilter und Zoom**

Momentan ist es lediglich möglich, die Helligkeit und die Gammawerte anzupassen. Für einen vollwertigen Viewer wären aber auch noch ein Kontrastfilter oder ein Zoom sinnvoll. Diese könnten über das CamanJs Framework implementiert werden. Dies wurde nicht umgesetzt, da der integrierte Kontrastfilter von CamanJs nicht richtig mit schwarz/weiß Bildern funktioniert und die Implementierung eines eigenen zu aufwändig gewesen wäre. Ebenso ist die Umsetzung eines Bild-Zooms über das bereits eingesetzte Canvas-Element kein Problem.
- **Optimieren des Ladens der Dicom-Attribut-Namen**

Momentan werden immer alle Dicom-Attribut-Namen in ein Globales-Array geladen. Dieser Schritt ist nicht optimiert. Es wäre aber möglich, alle Attribute in eine Memory-SQLite Datenbank auszulagern. Diese könnte dann über mehrere Aufrufe hinweg konsistent im Speicher des Servers gehalten werden und müsste nicht bei jedem Seitenaufruf neu aufgebaut werden.

Verwendete Programme / Bibliotheken

Genaue Versionsnummern

OS:	Ubuntu 6.5 LTS
Webserver:	Apache/2.2.12
PHP:	5.2.10-2 Suhosin Patch 0.9.7
Image-Magick:	6.5.1
NanoDicom:	V1.2 Imperial Cusco - Unstable
jQuery:	1.6.1 min
jQuery Mobile:	1.0.b1
CamanJS:	2.2

Beschreibung der Programme / Bibliotheken

jQuery

jQuery ist eine freie, umfangreiche JavaScript-Klassenbibliothek, welche komfortable Funktionen zur DOM-Manipulation und -Navigation zur Verfügung stellt. Die von John Resig entwickelte Klassenbibliothek wurde im Januar 2006 auf dem BarCamp (NYC) in New York veröffentlicht und wird laufend weiterentwickelt. Im September 2008 haben Microsoft und Nokia angekündigt, jQuery in ihren Produkten zu verwenden. jQuery ist die meistverwendetste JavaScript-Bibliothek weltweit. [jq]

Herunterladen kann man das Framework unter der Website:

http://docs.jquery.com/Downloading_jQuery .

jQuery

Mobile

Frei nach dem Motto „Write less, do more“ soll das neue jQuery Mobile Project für eine Arbeitserleichterung bei den Entwicklern mobiler Anwendungen sorgen. Statt wie bisher für jede mobile Plattform eigene Apps zu entwickeln und dabei auch noch auf Spezifikationen bestimmter Geräte achten zu müssen, sollen Entwickler mit dem neuen Framework eine plattformübergreifende App entwickeln können, die dann auf allen mobilen Geräten läuft. [jqm] Es bietet Werkzeuge zum Erstellen von Touchscreen-optimierten Oberflächen für jedes beliebige Darstellungsformat. Den Download findet man unter: <http://jquerymobile.com/download>.

CamanJS

CamanJS ermöglicht das Manipulieren von Bildern mithilfe von HTML5-Canvas. Es ist ein einfach zu benutzendes Javascript-Framework mit fortgeschrittenen und effizienten Bildbearbeitungsfiltern. Es ist komplett eigenständig, kann aber in Kombination mit anderen Javascript-Bibliotheken wie jQuery, Prototype oder Mootools verwendet werden.

CamanJS steht unter der BSD License und kann unter <http://camanjs.com> heruntergeladen werden.[cjs]

Image-Magick

Image-Magick ist eine Software zum Erstellen, Bearbeiten, Kombinieren oder Konvertieren von Bilddateien. Image-Magick ist eine freie Software und steht unter der Apache 2.0 Lizenz und kann für Windows, Linux und MacOS heruntergeladen bzw. kompiliert werden. Es werden viel mehr Formate wie durch GD unterstützt. So können über 100 Formate wie DPX, EXR, GIF, JPEG, JPEG-2000, PDF, PNG und viele weitere verarbeitet werden.

Die Funktionalität kann sowohl über die Konsole bedient werden, als auch durch ein Programm-API. Dazu stehen APIs für folgende Sprachen zur Verfügung: IMagic (PHP), G2F (Ada), MagickCore (C), ImageMagickObject (COM+), Magick++ (C++), JMagick (Java), L-Magick (Lisp). Image-Magick kann unter <http://www.imagemagick.org> oder über Synaptic bezogen werden. [im]

NanoDicom

NanoDICOM ist aus dem Projekt php-dicom entstanden, das sich ebenfalls zum Ziel gesetzt hatte, einen Dicom-Toolkit für PHP zu erstellen. Aufgrund schlechter Wartungs- und Erweiterbarkeit des php-dicom Codes wurde das Projekt inzwischen eingestellt, so dass NanoDICOM als offizieller Nachfolger von php-dicom gilt. Es ist stabiler, da es unter anderem ein richtiges Exception-Handling hat. Fremde Abhängigkeiten gibt es keine und es ist performanter als sein Vorgänger.

Es unterstützt DICOM 2009 und ist damit zur Zeit auf dem aktuellen Stand. Eine gute Dokumentation ist vorhanden und wird von einer Community unterstützt. Entwickelt wurde es vollständig objektorientiert und ist daher, im Gegensatz zu php-dicom, gut erweiterbar.

Getestet wurde NanoDICOM unter Windows7 32Bit sowie unter den Linux-Distributionen Debian Lenny 64Bit, Ubuntu 10.04 32Bit und Sun Solaris Ultra-Sparc 64Bit. NanoDICOM unterliegt der MIT Lizenz und kann unter <https://github.com/nanodocumet/Nanodicom> heruntergeladen werden.[nd]

Literaturverzeichnis

jq: jQuery, 2011, <http://de.wikipedia.org/wiki/JQuery>

jqm: jQuery Mobile, 2011, <http://t3n.de/news/jquery-mobile-project-web-framework-alle-mobilen-277572/>

cjs: ryan lefevre , 2011,

im: ImageMagick Studio LLC , 2011, <http://www.imagemagick.org/script/index.php>

nd: A lightweight PHP DICOM Toolkit, 2011, <http://www.nanodicom.org/>