

*enaio*<sup>®</sup>

Softwaredokumentation  
enaio<sup>®</sup> http-service

Version 8.50

Sämtliche Softwareprodukte sowie alle Zusatzprogramme und Funktionen sind eingetragene und/oder in Gebrauch befindliche Marken der OPTIMAL SYSTEMS GmbH, Berlin oder einer ihrer Gesellschaften. Sie dürfen nur mit gültigem Lizenzvertrag benutzt werden. Die Software sowie die jeweils zugehörige Dokumentation sind nach deutschem und internationalem Recht urheberrechtlich geschützt. Das illegale Kopieren und Vertreiben der Software stellt Diebstahl geistigen Eigentums dar und wird strafrechtlich verfolgt. Alle Rechte vorbehalten, einschließlich der Wiedergabe, Übermittlung, Übersetzung sowie Speicherung mit/auf Medien aller Art. Für vorkonfigurierte Testszenarien oder Demo-Präsentationen gilt: Alle Firmennamen und Personen, die in Beispielen (Screenshots) erscheinen, sind frei erfunden. Eventuelle Ähnlichkeiten mit tatsächlich existierenden Firmen und Personen sind zufällig und unbeabsichtigt.

Copyright 1992 – 2016 by

OPTIMAL SYSTEMS GmbH  
Cicerostraße 26  
D-10709 Berlin

01.02.2017  
Version 8.50

# Inhalt

<b>enaio® http-service</b>	<b>5</b>
Über enaio® http-service .....	5
Installation.....	5
Zweck.....	5
Funktionsweise.....	5
Konfiguration.....	6



# enaio® http-service

## Über enaio® http-service

enaio® http-service ist eine Komponente des Dokumentenmanagement-, Workflow- und Archivierungssystems enaio®.

Über enaio® http-service können Sie über das Internet oder ein Intranet Dokumente aus enaio® direkt in Browsern oder mit anderen http-Clients anzeigen lassen. enaio® http-service kann z. B. in Integrationsszenarien eingesetzt werden.

enaio® http-service wird über definierte URL angesprochen und liefert abhängig von den Berechtigungen des Anwenders das angeforderte Dokument.

## Installation

enaio® http-service wird als einzelnes Web-Archiv ausgeliefert. Dieses kann in einem Servlet Container (beispielsweise Tomcat) installiert werden. Mit dem Web-Archiv werden alle abhängigen Komponenten ausgeliefert. Eine Standardkonfiguration wird ebenfalls ausgeliefert. Diese muss an die örtlichen Gegebenheiten angepasst werden.

Das Web-Archiv `osecmhttpservice.war` finden Sie unter den Installationsdaten im Verzeichnis `\components\HTTP-Service\`.

## Zweck

enaio® http-service dient dazu, Dokumente aus dem enaio®-System direkt im Browser anzuzeigen oder an HTTP-Clients zu übergeben. Die Komponente kann beispielsweise von anderen Webanwendungen verwendet werden, um Dokumente direkt im Browserfenster anzuzeigen.

## Funktionsweise

enaio® http-service beinhaltet Funktionalitäten für die Authentifizierung/Autorisierung, Umgang mit unterschiedlich aufgebauten URL und das Bereitstellen von angeforderten Dokumenten.

Derzeit werden für die Authentifizierung HTTP Basic und NTLM unterstützt. Die Anwenderdaten werden dabei jeweils mit dem enaio®-System abgeglichen.

Als URL-Formate werden derzeit drei verschiedene Formate unterstützt.

1. URL-Format, das die IHE-Spezifikation unterstützt.

Format: `http://server:port/Anwendungsname/doc.osid.pdf` oder

`http://server:port/Anwendungsname/doc.osid`

`doc` ist ein fest zu vergebendes Kürzel

`osid` steht für die Objekt-ID aus enaio®

`pdf` ist optional und weist den Service an, explizit ein PDF-Dokument zu übergeben.

2. URL-Format, das die IHE-Spezifikation unterstützt.

Format:

`http://server:port/Anwendungsname/IHERetrieveDocument?requestType=DOCUMENT&documentUID=id&preferredContentType=application%2fpdf`

Für den `requestType` wird derzeit nur `DOCUMENT` unterstützt.

Für die documentUID wird derzeit nur die Angabe der Objekt-ID aus enaio® unterstützt.

Für den preferredContentType wird derzeit nur application%2fpdf unterstützt.

3. URL-Format, das die MS SharePoint Indexintegration von OPTIMAL SYSTEMS GmbH unterstützt.

Format:

```
http://server:port/Anwendungsname/ServiceName.pdf?app.entry.
aspath=3:262144;9477
```

```
http://server:port/Anwendungsname/ServiceName?app.entry.aspa
th=3:262144;9477
```

Die Angabe .pdf bewirkt, dass in jedem Fall ein PDF-Dokument geliefert wird. In dem Parameter app.entry.aspath wird die ID für den Schrank:Objekttyp;Objekt-ID übergeben.

Grundsätzlich arbeitet enaio® http-service nach einem Kettenprinzip. Zuerst wird der Anfragende authentifiziert. Anschließend wird die URL analysiert. Erst dann wird die eigentliche Anfrage ausgeführt.

## Konfiguration

Alle Konfigurationseinstellungen werden in der web.xml der Anwendung vorgenommen.

Es muss jeweils ein Authentifizierungsfilter und ein URL-Filter konfiguriert werden.

Der Authentifizierungsfilter benötigt die Angabe osecm.server. Dort wird entweder die IP oder Name des Servers auf dem enaio® installiert ist, angegeben. Mit osecm.port wird der entsprechende Port konfiguriert.

Als Authentifizierungsfilter stehen folgende Klassen zur Verfügung:

- com.os.http.NtlmHttpFilter – NTLM Authentifizierung
- com.os.http.BasicHttpFilter – HTTP Basic Authentifizierung
- com.os.http.ConfiguredFilter – Konfiguration des Accounts für den Zugriff

Als URL-Filter stehen folgende Klassen zur Verfügung:

- com.os.http.RetrieveDocumentURL-Filter – URL nach OPTIMAL SYSTEMS GmbH Spezifikation
- com.os.http.sharepoint.SpsURL-Filter – SharePoint Portal Server URL
- com.os.http.ihe.IheURL-Filter – URL-Filter nach IHE-Spezifikation.

Im Anschluss sehen Sie ein Beispiel für die Konfiguration eines HTTP Basic Authentifizierungsfilters, einen URL-Filter für SharePoint Portal Server URL und die Konfiguration des Servlets für das Bereitstellen des angeforderten Dokuments.

```
<filter>
    <filter-name>AuthenticationHttpFilter</filter-name>
    <!--filter-class>com.os.http.NtlmHttpFilter</filter-
class-->
    <filter-class>com.os.http.BasicHttpFilter</filter-
class>
    <init-param>
        <param-name>osecm.server</param-name>
        <param-value>127.0.0.1</param-value>
    </init-param>
    <init-param>
        <param-name>osecm.port</param-name>
        <param-value>4000</param-value>
    </init-param>
</filter>
```

```
<!-- URL Filter -->
<filter>
    <filter-name>URL-Filter</filter-name>
    <!--filter-class>com.os.http.RetrieveDocumentURL-
Filter</filter-class-->
    <filter-class>com.os.http.sharepoint.SpsURL-
Filter</filter-class>
</filter>
<filter-mapping>
    <filter-name>AuthenticationHttpFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
<filter-mapping>
    <filter-name>URL-Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
<!-- Define servlets that are included in the example
application -->
<servlet>
    <servlet-name>Retrieving Document Servlet</servlet-
name>
    <servlet-
class>com.os.servlet.http.RetrieveDocumentHttpServlet</servlet-
class>
</servlet>
<servlet-mapping>
    <servlet-name>Retrieving Document Servlet</servlet-
name>
    <url-pattern>/*</url-pattern>
</servlet-mapping>
```