



NAST

---

Dokumentation

<b>Dok.-Version:</b>	2.0.1	<b>Dok.-Status:</b>	Final
<b>Dok.-Stand:</b>	01.12.2015	<b>Dok.-Name:</b>	NAST-Dokumentation-DNS Services-V2.0.1-2015-12-01

## Impressum

Autor(en)	Abteilung	Telefon	E-Mail
DNS Services	DNS Services	+49-69-27 235 272	info@denic.de

## Dokument-Freigabe

Dokument-Version	Freigegeben von	Freigegeben am
2.0.1	DNS Services	01.12.2015

## Verteiler

Name
Mitglieder
Öffentlichkeit

---

Inhalt

<b>1.</b>	<b>Allgemeines.....</b>	<b>4</b>
1.1	Über dieses Dokument .....	4
1.2	Motivation.....	4
<b>2.</b>	<b>Inhalt des Softwarepakets.....</b>	<b>5</b>
<b>3.</b>	<b>command-line interface.....</b>	<b>6</b>
3.1	Systemvoraussetzungen .....	6
3.2	Installation .....	6
3.3	Konfiguration .....	7
3.4	Optionen.....	7
3.4.1	Domain .....	7
3.4.2	Nameserver.....	7
3.4.3	Resolver .....	9
3.4.4	Port.....	9
3.4.5	Ausgabeformat.....	9
3.4.6	Hilfe .....	9
3.4.7	DNSSEC-Optionen.....	10
<b>4.</b>	<b>NAST-Webservice.....</b>	<b>11</b>
4.1	Systemvoraussetzungen.....	11
4.2	Erstinstallation .....	11
4.3	Starten und Stoppen .....	12
4.4	Update des Web-Service.....	12
4.5	Deinstallation .....	12
4.6	Konfiguration .....	13
4.7	Logging.....	13
4.8	HTTP-Request.....	14
4.8.1	URI .....	14
4.8.2	HTTP-Methoden .....	15
4.8.3	HTTP-Response.....	15
4.8.4	HTTP-Statuscodes .....	15
4.8.5	HTTP-Header.....	16
4.8.6	HTTP-Body.....	16
4.9	XML-Schema-Dateien.....	18
<b>5.</b>	<b>Programm-Quelltext.....</b>	<b>19</b>
<b>6.</b>	<b>Issue-Codes.....</b>	<b>20</b>
<b>7.</b>	<b>Dokumenthistorie.....</b>	<b>25</b>

# 1. Allgemeines

## 1.1 Über dieses Dokument

Die vorliegende Dokumentation beschreibt die Software (Quellcode, REST-Service und command-line interface) sowie den Web-Client, die DENIC zur Validierung von Nameservern hinsichtlich korrekter Delegationsdaten für Secondleveldomains unterhalb .de und 9.4.e164.arpa zur Verfügung stellt.

Wichtiges wird separat schattiert dargestellt.

Neuerungen sind durch rote Balken gekennzeichnet.

Der NAST-Web-Client (NAST = NAME-Server-Tester) ist erreichbar unter <http://nast.denic.de>.

### Nameserver Predelegation Check Webinterface

Mit dem NAST Webinterface können Sie einen Nameserver Predelegation Check durchführen. Die Nameserver Ihrer Zone (Domain) werden dabei verschiedenen Tests unterzogen, um sicherzustellen, dass sie korrekt konfiguriert sind und die Domain sicher und einfach delegiert werden kann. Damit wird ein hoher Grad an Qualität für die Domain erzielt.

Bitte tragen Sie im Formular Ihre Domain ein. Die Angabe der Nameserver ist optional. Sind die Nameserver nicht angegeben, so werden diese automatisch aus dem DNS ermittelt. Dies ist natürlich nur bei bereits registrierten Domains möglich.

Domain:

Nameserver 1:	<input type="text" value="ns1.domain.de"/>	IPs:	<input type="text"/>
Nameserver 2:	<input type="text" value="ns2.domain.de"/>	IPs:	<input type="text"/>
Nameserver 3:	<input type="text"/>	IPs:	<input type="text"/>

#### Hinweis

Die Zeit, die benötigt wird um eine komplette Zone zu verifizieren, kann bis zu 35 Sekunden betragen.

## 1.2 Motivation

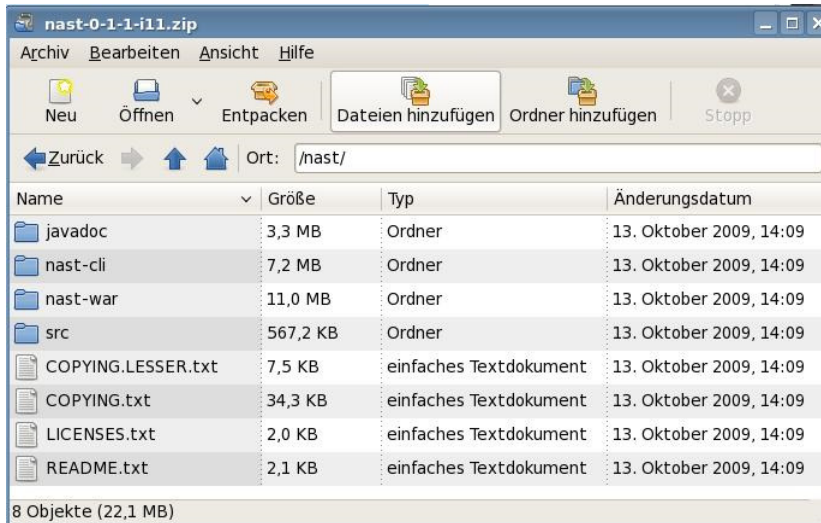
Das Softwarepaket NAST (NAME Server Tester) dient dazu, die Tests welche die DENIC im Rahmen des Nameserver Predelegation Checks durchführt, selbst durchzuführen, um initiale Setup-Fehler zu vermeiden und die Delegation von Secondleveldomains unterhalb .de bzw. 9.4.e164.arpa zu erleichtern.

Die Anforderungen finden Sie in der Dokumentation DENIC-23 – Nameserver Predelegation Check unter <http://nast.denic.de>

Unter <http://nast.denic.de> stellt die DENIC ein Softwarepaket zur Verfügung.

## 2. Inhalt des Softwarepakets

In dem ZIP-Archiv von NAST ist folgendes enthalten:



CHANGELOG.txt	Changelog (Übersicht über alle Versionsänderungen)
COPYING.LESSER.txt	enthält die GNU Lesser General Public License unter der DENIC die Software zur Verfügung stellt
COPYING.txt	GNU General Public License
LICENSES.txt	listet weitere verwendete 3 <sup>rd</sup> party libraries auf
<i>nast-war</i>	den REST-Service zum Einbinden in eine Servlet Engine
<i>javadoc</i>	API-Dokumentation für NAST im HTML-Format
<i>src</i>	Programm-Quellcode von NAST, der unter der LGPL <sup>1</sup> (GNU Lesser General Public License) Version 3.0 zur Verfügung gestellt wird.
<i>nast-cli</i>	command-line interface

<sup>1</sup> <http://www.gnu.org/licenses/>

### 3. command-line interface

Das command-line interface (Kommandozeilen-Interface) von NAST finden Sie im Verzeichnis *nast-cli*.



Datei	Bedeutung
nast-cli.jar	Standalone-Anwendung
config.properties.example	Konfigurationsdatei (Optional)

#### 3.1 Systemvoraussetzungen

Das System, auf dem NAST installiert werden soll, muss folgende Voraussetzungen erfüllen:

- Es wird ein Java Runtime Environment ab Version 1.7,
- der Zugriff auf einen Default-Resolver,
- ein Zugang zum DNS über IPv4 und IPv6 benötigt.

#### 3.2 Installation

Das command-line interface kann direkt von der Kommandozeile aus mit dem Befehl `java -jar nast-cli.jar <Option>` aufgerufen werden.

### 3.3 Konfiguration

In der Datei `config.properties` können folgende technische Parameter konfiguriert werden:

Parameter	Bedeutung	Wertebereich	Obligatorisch/ Optional
<code>default.resolver.interface</code>	Hostname des Default-Resolvers (Default: <i>localhost</i> )	"localhost", Interface-Name oder IP-Adresse	Optional
<code>default.resolver.port</code>	Portnummer des Default-Resolvers (Default: <i>53</i> )	0 – 65535	Optional
<code>global.unicast.ipv6.url</code>	URL zum Laden der "IPv6 Global Unicast Address Assignments"; (Default: <i>http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml</i> ) URL kann auch <code>file://&lt;datei&gt;</code> sein	Gültige URL gemäß RFC3986	Obligatorisch

Falls Sie keinen Default-Resolver konfigurieren, versucht NAST automatisch den Default-Resolver der Plattform zu ermitteln. Die Anwendung geht dabei nach dem hier dokumentierten Verfahren vor: <http://www.dnsjava.org/dnsjava-current/doc/org/xbill/DNS/ResolverConfig.html>

### 3.4 Optionen

Nachfolgend finden Sie eine Übersicht über die Optionen des command-line interface.

#### 3.4.1 Domain

Mit `-d <domain>` wird die zu prüfende Domain übergeben.

Diese Option kann nur in Verbindung mit der Option `-n <Nameserver>` verwendet werden.

#### 3.4.2 Nameserver

Mit der Option `-n` werden die zu prüfenden Nameserver übergeben.

Als Trenner zwischen Hostname und IPV4 und/oder IPv6 Adresse wird ein Leerzeichen verwendet.

Als Trenner zwischen den Hostnamen wird ein Komma „`,`“ verwendet.

Beispiel: Parameter für -n

```
java -jar nast-cli.jar -d denic.net -n "ns1.denic.de,ns2.denic.de"
```

```
java -jar nast-cli.jar -d denic.de -n "ns1.denic.de 81.91.170.1  
2001:608:6:6:0:0:0:11,ns5.denic.net"
```

```
java -jar nast-cli.jar -d denic.de -n "ns1.denic.de 81.91.170.1  
2001:608:6:6:0:0:0:11,ns2.denic.de 193.171.255.36,ns3.denic.de  
87.233.175.19,ns4.denic.net,ns5.denic.net"
```



### 3.4.3 Resolver

Mit „-i <Resolver>“ können Sie einen optionalen Resolver für die Abfrage mit angeben:

Beispiel: Optionale Resolver

```
java -jar nast-cli.jar -d denic.net -n "ns1.denic.de,ns2.denic.de" -i resolver.denic.de -p 9053
```

Bei Angabe von -i <Resolver> wird der in der Konfiguration angegebene Resolver nicht verwendet. Eine Eingabe von -p <Port> ist zwingend erforderlich.

### 3.4.4 Port

Mit -p <Port> können Sie einen anderen Port als den Defaultport 53 für die Abfrage mit angeben:

Beispiel: Optionaler Resolverport

```
java -jar nast-cli.jar -d denic.net -n "ns1.denic.de,ns2.denic.de" -i resolver.denic.de -p 9053
java -jar nast-cli.jar -d denic.net -n "ns1.denic.de,ns2.denic.de" -p 9053
```

Bei Angabe von -p <Port> wird der in der Konfiguration angegebene Port nicht verwendet.

### 3.4.5 Ausgabeformat

Mit -f XML oder -f JSON können Sie das Ausgabeformat festlegen.

Mit -f XML erfolgt die Ausgabe im XML-Format:

Beispiel: Ausgabe im XML-Format.

```
[lorelai@starshollow]$ java -jar nast-cli.jar -d denic.net -n "ns1.denic.de,ns2.denic.de" -f XML
Version: 1-0--1-i18 (2010-01-08 12:43)
Applying JVM/OS stub as default recursive resolver
<predelegation-check success="true" xmlns="http://schema.denic.de/rng/nast"/>
```

Mit -f JSON erfolgt die Ausgabe im JSON-Format:

Beispiel: Ausgabe im JSON-Format.

```
[lorelai@starshollow]$ java -jar nast-cli.jar -d denic.net -n "ns1.denic.de,ns2.denic.de" -f JSON
Version Version: 1-0--1-i18 (2010-01-08 12:43)
Applying JVM/OS stub as default recursive resolver
{"success":true}
```

### 3.4.6 Hilfe

Mit -h kann ein allgemeiner Hilfetext abgefragt werden.

Beispiel: Ausgabe im XML-Format.

```
[lorelai@starshollow]$ java -jar nast-cli.jar -d denic.net -n "ns1.denic.de,ns2.denic.de" -f
JSON
No config file "config.properties" found. Use example "config.properties.example" file to
build one if applicable.
EXAMPLE: java -jar nast.jar -h -d DOMAIN -f FORMAT -h -n NAMESERVER -i RESOLVER -
p PORT -v
-d DOMAIN    : Name of domain to execute tests against (like "test.xx")
-f FORMAT    : Optional format of output ("XML" or "JSON")
-h           : Print this command line parameter help
-k           : Comma-separated list of dnskey parameters (like "256 3 5
              DnskeyAsBase64, 257 3 5 DnskeyAsBase64")
-n NAMESERVER : Comma-separated list of nameserver parameters (like "ns1.test.x
              x 192.168.1.1, ns2.other.xx")
-D DNSSEC    : Name of optionals predelegation check
-i RESOLVER  : Overrides configured interface of default resolver (like "192.1
              68.1.1" or "myresolver.xx")
-p PORT      : Overrides configured port of default resolver (like "53")
-v           : Be verbose
```

### 3.4.7 DNSSEC-Optionen

Mit `-D DNSSEC` werden auch DNSSEC-spezifische Predelegation-Checks durchgeführt.

Wenn `-D` nicht angegeben wird wird der Default angenommen und es werden keine DNSSEC-spezifischen Checks durchgeführt.

Beispiel: Predelegation-Check mit DNSSEC

```
java -jar nast-cli.jar -d denic.net -n "ns1.denic.de,ns2.denic.de" -D DNSSEC -k "257 3 5
AwEAAAdDECajH
aTjfSoNTY58WcBah1BxPKVIHBz4IfLjfqMvium4IlgKtKZLe97DgJ5/NQrNEGGQmr6fKvUj67
cfrZUojZ2cGRizVhgkOqZ9scaTVXNuXLM5Tw7VWOVlceeXAuuH2mPliEV6MhJYUsW6d
vmNsJ4XwCgNgroAmXhoMEiWEjBB+wjYZQ5GtZHBFKVXACSWTiCtddHcueOeSVPi5W
H94VlubhHfiytNPZLrObhUCHT6k0tNE6phLoHnXWU+6vpsYpz6GhMw/R9BFxW5PdPFI
WBgoWk2/XFVRSKG9Lr61b2z1R126xeUwww46RVy3hanV3vNO7LM5HniqaYcIBhk="
```

## 4. NAST-Webservice

Der NAST-Webservice ist eine Webanwendung die in einem Servlet Container abläuft und als RESTful Webservice implementiert ist. REST (REpresentational State Transfer) ist ein Architekturstil, der für die Kommunikation zwischen Client und Server die Semantik und Methoden des HTTP-Protokolls verwendet. Eine Einführung zu REST finden Sie zum Beispiel in der Dissertation von Roy Thomas Fielding<sup>2</sup>. Für die Verwendung des Webservice sind nähere Kenntnisse über REST nicht erforderlich. Wie ein Request aufgebaut ist und welche Antworten der Webservice zurückliefert, wird im Folgenden erläutert.

Client und Server kommunizieren über HTTP, Version 1.1.

### 4.1 Systemvoraussetzungen

Das System, auf dem der NAST-Webservice installiert werden soll, muss folgende Voraussetzungen erfüllen:

- Es wird ein Servlet-Container, der die Servlet-API 2.4 implementiert hat (z.B. Tomcat ab Version 5.5.x),
- ein Java Runtime Environment ab Version 1.7,
- der Zugriff auf einen Default-Resolver,
- ein Zugang zum DNS über IPv4 und IPv6 benötigt.

### 4.2 Erstinstallation

1. Entpacken Sie die Datei `nast.war` aus dem Archiv
2. Editieren Sie die in `nast.war` enthaltene Datei `web.xml` (siehe Kapitel 4.6) und speichern Sie die geänderte `web.xml` in der WAR-Datei.
3. Deployen Sie die WAR-Datei anschliessend im Servlet-Container oder verschieben Sie die WAR-Datei in das entsprechende Verzeichnis des Servlet-Containers und starten Sie diesen neu.

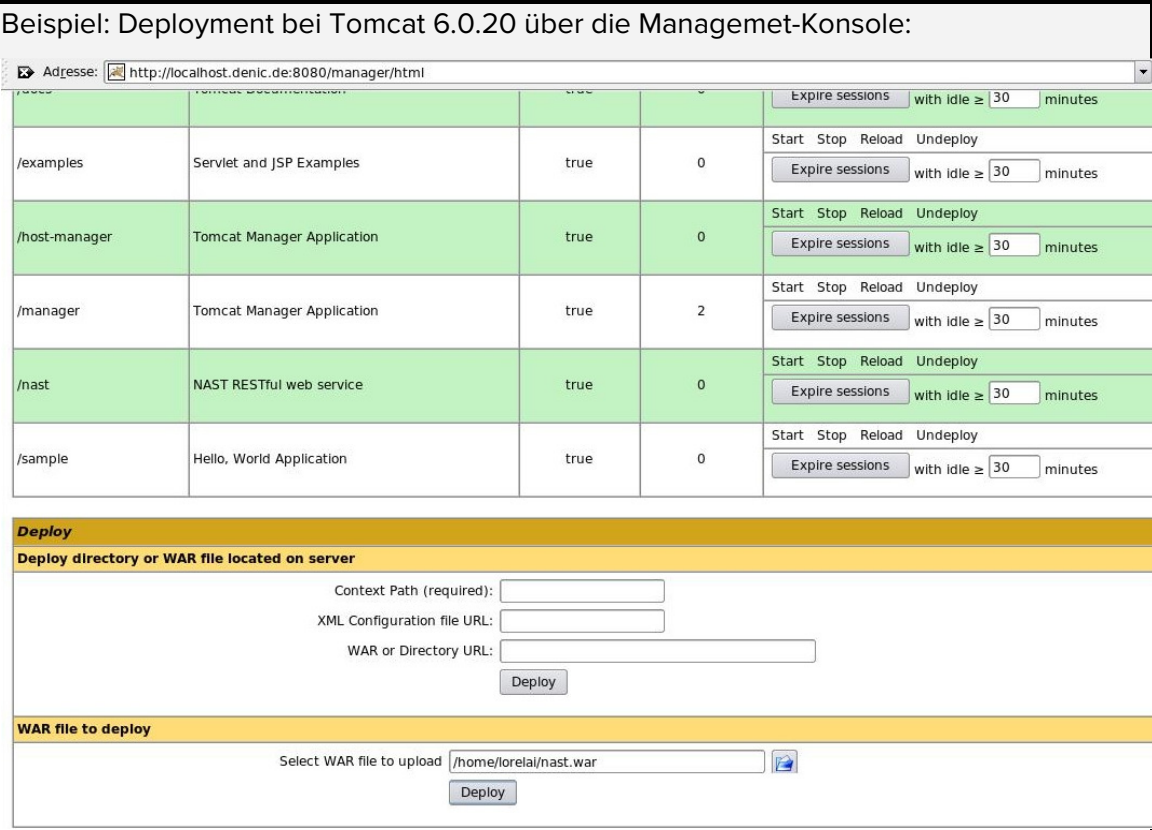
Beispiel: Installation bei Tomcat 6.0.20:

```
[lorelai@starshollow]$ cp nast.war ~/apache-tomcat-6.0.20/webapps/  
[lorelai@starshollow]$ cd ~/apache-tomcat-6.0.20/bin/  
[lorelai@starshollow]$ ./startup.sh  
[lorelai@starshollow]$
```

---

<sup>2</sup> <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

Beispiel: Deployment bei Tomcat 6.0.20 über die Management-Konsole:



Context Path	WAR File	Auto-Deploy	Session Timeout	Actions
/examples	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	Tomcat Manager Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/nast	NAST RESTful web service	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/sample	Hello, World Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

**Deploy**

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

**WAR file to deploy**

Select WAR file to upload

### 4.3 Starten und Stoppen

Starten und stoppen des Web-Service erfolgt mit den vom eingesetzten Servlet-Container bereitgestellten Hilfsmitteln (zum Beispiel Start-/Stop-Skripte oder Management-Konsole).

### 4.4 Update des Web-Service

Um den Web-Service upzudaten stoppen und undeployen Sie die aktuell im Servlet-Container laufende Webanwendung und installieren Sie den Service wie im Kapitel 4.2 beschrieben.

Bitte beachten Sie, dass Sie die Parameter in der `web.xml` neu konfigurieren müssen.

### 4.5 Deinstallation

Um die Software zu Deinstallieren kann entweder der Servlet-Container gestoppt und die Anwendung gelöscht oder aber der Service undeployed werden.

## 4.6 Konfiguration

In der Datei `web.xml` gibt es die folgenden Konfigurations-Parameter:

Parameter	Bedeutung	Wertebereich	Obligatorisch/ Optional
<code>default.resolver.interface</code>	Hostname des Default-Resolvers (Default: <i>localhost</i> )	Interface-Name oder IP-Adresse	Optional
<code>default.resolver.port</code>	Portnummer des Default-Resolvers (Default: <i>53</i> )	0 – 65535	Optional
<code>global.unicast.ipv6.url</code>	URL zum Laden der "IPv6 Global Unicast Address Assignments" (Default: <i>http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml</i> ) URL kann auch <code>file://&lt;datei&gt;</code> sein	Gültige URL gemäß RFC3986	Obligatorisch

Falls Sie keinen Default-Resolver konfigurieren, versucht NAST automatisch den Default-Resolver der Plattform zu ermitteln. Die Anwendung geht dabei nach dem hier dokumentierten Verfahren vor:

<http://www.dnsjava.org/dnsjava-current/doc/org/xbill/DNS/ResolverConfig.html>

## 4.7 Logging

Das Logging der Anwendung kann über die Datei `log4j.properties` im Verzeichnis `WEB-INF/classes` konfiguriert werden.

Der Webservice loggt standardmässig in die Datei `nast.log`.

Jeder Request wird in folgendem Format geloggt:

```
timestamp, second since epoch|policy token|quell-ip-adresse|
Laufzeit, Millisekunden, ganzzahlig|NAST-Status|
Liste der NAST-Warnungscodes|
```

Diese Daten können Sie zum Beispiel statistisch auswerten.

## 4.8 HTTP-Request

Ein HTTP-Request setzt sich aus einem Uniform Resource Identifier (URI) und einer HTTP-Methode zusammen. Der URI identifiziert dabei die Ressource. Die HTTP-Methode gibt an, was mit der Ressource getan werden soll.

### 4.8.1 URI

Im URI müssen im vorliegenden Kontext ein Domainname und mindestens 2 Nameserver (sowie optional deren Glue Adressen) angegeben werden. Weiterhin muss im URI das Darstellungsformat für die Antwortdaten angegeben werden.

Ein gültiger URI hat folgende Syntax (variable Anteile in <spitzen Klammern>):

`http://<Hostname>:<Port>/nast/<Darstellungsformat>/<Domain>?ns1=<Nameserver-1>&ns2=<Nameserver-2>&[nsX=<Nameserver-N>]*`

**Hinweis:** Ab dem dritten Nameservern lautet der URL-Parameter *immer* nsX, wird also nicht mit ns3, ns4 und so weiter hochgezählt.

Erläuterung der URI:

Variable	Bedeutung	Mögliche Werte	Obligatorisch / Optional
Hostname	Name des Hosts, auf dem der Webservice läuft.	localhost, Interface-Name oder IP-Adresse	Obligatorisch
Port	Portnummer des Hosts, auf dem der Webservice läuft.	0 – 65535	Optional
Darstellungsformat	Format, in dem der Client die Antwort dargestellt haben möchte. Mögliche Formate sind XML oder JSON	XML <sup>3</sup> JSON <sup>4</sup>	Obligatorisch
Domain	Name der zu prüfenden Domain	Gültige Domain	Obligatorisch
ns1-nsX	Namen (und optionale Glue-Adressen) der zu prüfenden Nameserver.	Format: Hostname[,IPv4[,IPv6]] Bei den Glue-Adressen sind sowohl IPv4- als auch IPv6-Adressen erlaubt.	Obligatorisch: Mindestens zwei Nameserver
Policy	Angabe, ob auch DNSSEC spezifische Predelegation Checks durchgeführt werden sollen.	DE-DNSSEC-1.0	Optional

<sup>3</sup> <http://de.wikipedia.org/wiki/XML>

<sup>4</sup> <http://de.wikipedia.org/wiki/JSON>

Dnskey	Zu prüfender Dnskey (Flags, Protokoll, Algorithmus und base64-kodierter Public-Key)	Format: Flags,Protokoll,Algorithmus,Public-Key	Optional
--------	---	--	----------

**Beispiel: URI**

Die Antwort soll in XML-Notation dargestellt werden (Kennung "xml" in der URI) :  
`http://host.example:8080/nast/xml  
 /beispiel-eins.de?ns1=ns1.beispiel-eins.de,81.91.170.1,2001:608:6:6:0:0:11&ns2=ns2.denic.de`

Die Antwort soll in JSON-Notation dargestellt werden (Kennung "json" in der URI) :  
`http://host.example:8080/nast/json  
 /beispiel-eins.de?ns1=ns1.beispiel-eins.de,81.91.170.1,2001:608:6:6:0:0:11&ns2=ns2.denic.de`

Im obigen Beispiel wird ein HTTP-Request an den Host "host.example" auf Port 8080 gerichtet. Durch die Angabe "nast" in der URI erkennt der Host, dass der Dokumentation aufgerufen werden soll. Es soll geprüft werden, ob die Delegation der Domain "beispiel-eins.de" an die Nameserver "ns1.beispiel-eins.de" und "ns2.denic.de" den Nameserverdelegations-Anforderungen von DENIC entspricht.

#### 4.8.2 HTTP-Methoden

Der Webservice lässt folgende HTTP-Methoden zu: GET, HEAD und OPTIONS.

- Mit GET wird der Nameserver Predelegation Check ausgeführt.
- HEAD liefert die gleichen HTTP-Header wie GET, jedoch ohne Nutzdaten im Body.
- Mit OPTIONS erfährt der Client, welche HTTP-Methoden vom Server unterstützt werden.

#### 4.8.3 HTTP-Response

Der Webservice antwortet mit einem HTTP-Response, der sich – wie bei HTTP üblich - aus Nachrichtenkopf (Head) und Nachrichtenkörper (Body) zusammensetzt. Im Nachrichtenkopf sind der HTTP-Statuscode und diverse HTTP-Header hinterlegt. Die HTTP-Header liefern u.a. Meta-informationen zum Body (zum Beispiel welche Kodierung verwendet wird). Der Body enthält die Nutzdaten, im vorliegenden Fall also das Ergebnis des Nameserver Predelegation Checks.

#### 4.8.4 HTTP-Statuscodes

In der unten stehenden Tabelle finden Sie die möglichen HTTP-Statuscodes:

Statuscode	Statusmeldung	Erläuterung
200	OK	Der Request war erfolgreich.
400	Bad Request	Der Request konnte vom Server aufgrund einer fehlerhaften Syntax nicht verstanden werden.

405	Method Not Allowed	Die im Request angegebene HTTP-Methode ist nicht erlaubt.
500	Internal server error	Es ist ein unerwarteter Serverfehler aufgetreten. Der Request konnte nicht verarbeitet werden.

Die HTTP-Statuscodes sind im RFC2616<sup>5</sup> definiert.

#### 4.8.5 HTTP-Header

Als HTTP-Header "Content-Type" sind folgende Werte möglich:

- application/xml;charset=ISO-8859-1: Der Inhalt des Body wird in XML-Notation dargestellt.
- application/json;charset=ISO-8859-1: Der Inhalt des Body wird in JSON-Notation dargestellt.

#### 4.8.6 HTTP-Body

Der Body enthält die fachliche Antwort des Servers.

Beispiel: HTTP-Body in JSON-Notation

```
{"success":false,"issues":[{"severity":"warning","code":120,"message":"Recursive queries should not be allowed (resolver)","arguments":["/81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s)"]}, {"severity":"error","code":116,"message":"Authoritative answer regarding SOA record is required (resolver, answer)","arguments":["/81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s)"/, "81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s): RA <No records>"]}, {"severity":"error","code":116,"message":"Authoritative answer regarding SOA record is required (resolver, answer)","arguments":["/81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s)"/, "81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s): RA <No records>"]}, {"severity":"warning","code":120,"message":"Recursive queries should not be allowed (resolver)","arguments":["/81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s)"]}]}]
```

<sup>5</sup> <http://www.ietf.org/rfc/rfc2616.txt>



#### Beispiel: HTTP-Body in XML-Notation

```
<?xml-stylesheet type="text/xsl" href="../meta/nast_answer.xsl"?>
<predelegation-check success="false" xsi:schemaLocation="../meta/nast_answer.xsd"
xmlns="http://schema.denic.de/rng/nast" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
<issues>
<issue code="120" severity="warning">
<message>Recursive queries should not be allowed (resolver)</message>
<parameters>
<parameter>/81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s)</parameter>
</parameters>
</issue>
<issue code="116" severity="error">
<message>Authoritative answer regarding SOA record is required (resolver,
answer)</message>
<parameters>
<parameter>/81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s)</parameter>
<parameter>/81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s): RA &lt;No
records></parameter>
</parameters>
</issue>
<issue code="106" severity="error">
<message>Authoritative answer regarding SOA record is required (resolver,
answer)</message>
<parameters>
<parameter>/81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s)</parameter>
<parameter>/81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s): RA &lt;No
records></parameter>
</parameters>
</issue>
<issue code="120" severity="warning">
<message>Recursive queries should not be allowed (resolver)</message>
<parameters>
<parameter>/81.81.170.12:53 (UDP, Timeout: 5s, Retry: 3 x 5s)</parameter>
</parameters>
</issue>
</issues>
</predelegation-check>
```

In der Antwort gibt es bestimmte Schlüsselwörter, denen entsprechende Werte zugeordnet sind:

Schlüsselwort	Bedeutung	Mögliche Werte und deren Bedeutung
success	Gibt an, ob der Nameserver Predelegation Check erfolgreich war.	true (Der Check war erfolgreich.) false (Der Check war nicht erfolgreich; es gibt mindestens ein Issue mit severity=error.)
issues <sup>6</sup> (Liste)	Eine Liste mit Fehlermeldungen, Sie gibt an, welche im Dokument DENIC-23 – Nameserver Predelegation Check formulierte Empfehlung oder Pflicht-Anforderung nicht erfüllt wurde.	1 – n Fehlermeldungen. Jede Fehlermeldung (Issue) besitzt maximal folgende Attribute: severity (Obligatorisch) code (Obligatorisch) message (Optional) Parameters (Optional).
severity	Gewichtung	warning error
code	Eindeutiger Fehlercode	Numerischer Code
message	Erläuternder Text	String
parameters	0 – n Parameter. Ein Parameter gibt an, auf welches Objekt sich die Fehlermeldung bezieht.	String
arguments	Wird bei JSON-Notation anstelle von „parameters“ verwendet, ansonsten identisch.	String

## 4.9 XML-Schema-Dateien

Über die folgende relative URI können die Schema-Dateien zur XML-Notation heruntergeladen werden:

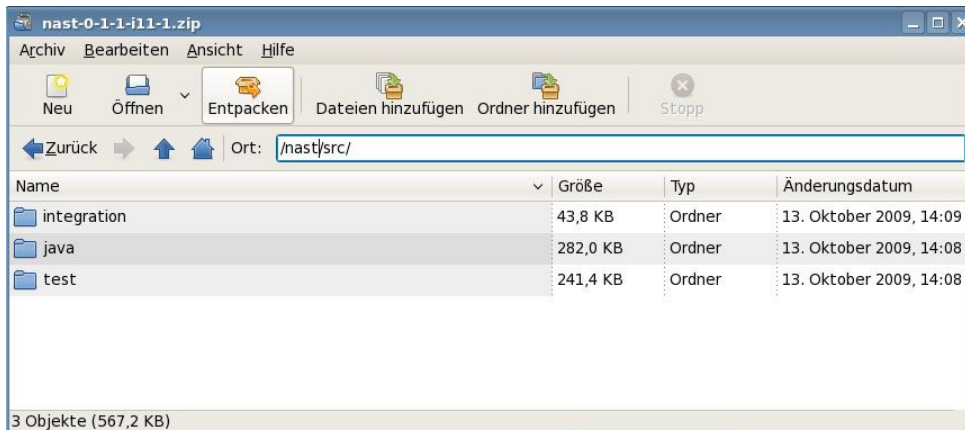
Schema	Relative URI
RELAX NG Schema <sup>7</sup>	/nast/meta/nast_answer.rng
XML-Schema	/nast/meta/nast_answer.xsd

<sup>6</sup> Siehe Kapitel 7. Issue Codes

<sup>7</sup> Siehe <http://relaxng.org/>

## 5. Programm-Quelltext

Im Verzeichnis `src` finden Sie den Programm-Quelltext von NAST:



Bitte beachten Sie, dass wir den Quelltext unserer selbstentwickelten Bibliotheken nicht weitergeben Ihnen jedoch die kompilierten Bibliotheken zur Verfügung stellen.

Damit ist es möglich, den Quelltext von NAST zu kompilieren. Die entsprechenden JAR-Files finden Sie im Verzeichnis `WEB-INF/lib` der Implementierung des NAST-Webservice.

In der mitgelieferten Javadoc finden Sie den zentralen fachlichen Einstiegspunkt im Interface `de.denic.zone.nast.business.PredelegationCheck` mit seinen Implementierungsklassen, insbesondere `PredelegationCheckImpl`.

Der erste Konstruktor dieser Klasse erwartet drei Objekte, von denen die Implementierung abhängig ist:

- Ein `ExecutorService` aus dem JDK.
- Eine Implementierung des Interfaces `de.denic.techsupport.dns.DnsQuery`, das die Abwicklung von DNS-Anfragen kapselt. Die Implementierung `de.denic.techsupport.dns.impl.DnsQueryDnsJavaImpl` finden Sie im JAR `techsupport-dns-<VERSION>.jar`. Diese Implementierung basiert ihrerseits auf der `dnsjava`-Bibliothek. Sie bietet neben einem Default-Konstruktor die Möglichkeit, den rekursiven Default-Resolver zu konfigurieren; der Default-Konstruktor nutzt dazu den Stub-Resolver des OS bzw. der VM. Um sich eine Implementierung des Interfaces `de.denic.techsupport.dns.Resolver` zusammenbauen zu können, empfehlen wir den `de.denic.techsupport.dns.ResolverBuilder`.
- Eine Implementierung des Interfaces `de.denic.techsupport.ipaddress.Ipv6GlobalUnicastAddressCheck`, das dafür zuständig ist zu prüfen, ob eine IPv6-Adresse zu einem von der IANA zugewiesenen Adressblock gehört. Die Implementierung `de.denic.techsupport.ipaddress.impl.Ipv6GlobalUnicastAddressCheckIanaPageParsingImpl` finden Sie im JAR `techsupport-ipaddress-<VERSION>.jar`. Diese wiederum erwartet einen URL, der auf die Text- oder XML-Quelle verweist, von wo diese Informationen bezogen werden können. Weiterhin wird ein Flag erwartet, das angibt, welches Ergebnis zurückgeliefert werden soll für den Fall, dass irrtümlich eine IPv4- und nicht eine IPv6-Adresse zur Prüfung übergeben wird. Optional können Sie einen `ScheduledExecutorService` übergeben, mit dem Sie bestimmen, wie häufig diese Informationen aus der konfigurierten URL aktualisiert werden.

Bitte beachten Sie, dass die DENIC keine Hilfestellung bei Fragen rund um die Implementierung des Quelltext gibt. Sollten Sie Anregungen und Verbesserungsvorschläge haben, so können Sie diese gerne an [nast-feedback@denic.de](mailto:nast-feedback@denic.de) senden.

## 6. Issue-Codes

Die unten angegebenen Issuecodes bleiben unverändert auch wenn der Text der Meldung („Message“) mit der Zeit variiert werden kann. Es wird daher ausdrücklich empfohlen, auf die Issue-Codes zu parsen. Aktuell sind folgende Issue-Codes definiert:

<b>Issue-Code</b>	<b>Message</b>	<b>Bezug auf Kapitel / Absatz in Dokumentation DENIC-23</b>
101	Missing glue record for the nameserver (NS)	Kap. 2.1.3, Absatz 1a
102	Provided glue records not applicable (NS)	Kap. 2.1.3, Absatz 1b
103	Nameservers having IPv6 glue records should have IPv4 glue records too (NS, # of IPv4 glues, # of IPv6 glues)" RECURSION_AVAILABLE => "Recursive queries should not be allowed (resolver)	Kap. 2.1.3, Absatz 2
104	Calculated referral response larger than allowed (length in octets)	Kap. 2.1.5, Absatz 3
105	All IPv6 Addresses must be Global Unicast dedicated, allocated and routable	Kap. 2.1.6, Absatz 1
106	Inconsistent set of nameserver IP addresses (NS, provided glues, determined glues)	Kap. 2.1.3, Absatz 3
106	Received response does not provide expected records directly (resolver, NS, RR)	Kap. 2.1.3, Absatz 3
106	Received response not authoritative (resolver, NS, RR)	Kap. 2.1.3, Absatz 3
107	Insufficient diversity of nameserver' s IP addresses (expected, found)	Kap. 2.1.2
107	Insufficient diversity of nameserver' s IPv4 addresses (expected, found, IPs found)	Kap. 2.1.2
107	Insufficient number of nameservers reachable via IPv4 (expected, found)	Kap. 2.1.2
107	Insufficient number of nameservers reachable (expected, found)	Kap. 2.1.2
108	Refresh value out of range	Kap. 2.1.4, Absatz 2a
109	Retry value out of range (expected, found)	Kap. 2.1.4, Absatz 2b
110	Retry value out of range (expected, found)	Kap. 2.1.4, Absatz 2c
111	Expire value out of range (expected, found)	Kap. 2.1.4, Absatz 2d

<b>Issue-Code</b>	<b>Message</b>	<b>Bezug auf Kapitel / Absatz in Dokumentation DENIC-23</b>
112	Minimum TTL out of range (expected, found)	Kap. 2.1.4, Absatz 2e
113	Primary Master (MNAME) inconsistent across SOA records	Kap. 2.1.5, Absatz 4
114	Inconsistent serial number across SOA records (serial number)	Kap. 2.1.4, Absatz 1
115	SOA record response must be direct (resolver, answer)	Kap. 2.1.5, Absatz 2
116	SOA record response must be authoritative (resolver, answer)	Kap. 2.1.1
118	Inconsistent set of NS RRs (IP, NS host names)	Kap. 2.1.5, Absatz 1
118	NS query response is empty	Kap. 2.1.5, Absatz 1
119	Some Nameservers not reachable via TCP	Kap. 2.1.6, Absatz 3
120	Recursive queries should not be allowed (resolver)	Kap. 2.1.6, Absatz 2
121	Received a truncated response (resolver, answer)	Kap. Keines
200	DNSKEY RR ZONE flag (bit 7) must be set	Kap. 3.6.1.1, Absatz 1
201	DNSKEY RR REVOKE flag (bit 8) must not be set	Kap. 3.6.1.1, Absatz 2
202	DNSKEY RR SEP flag (bit 15) should be set	Kap. 3.6.1.1, Absatz 3
203	DNSKEY RR RSA key modulus length in bits out of range	Kap. 3.6.1.4, Absatz 1
204	DNSKEY RR RSA public key exponent length in bits must not exceed 128 bits	Kap. 3.6.1.4, Absatz 2
205	DNSKEY RR DSA public key parameter T out of range	Kap. 3.6.1.4, Absatz 3
206	DNSKEY RR DSA public key has invalid size	Kap. 3.6.1.4, Absatz 4
207	DNSKEY RR public key must be BASE64 encoded	Kap. 3.6.1.4
208	Duplicate DNSKEY RR	Kap. 3.6.1
209	At least one DNSKEY RR must be specified in request	Kap. 3.6.1.4
210	At most 5 DNSKEY RR allowed	Kap. 3.6.1 und 3.6.2
211	Inconsistent DNSKEY RR in nameserver response	Kap. 3.6.2

<b>Issue-Code</b>	<b>Message</b>	<b>Bezug auf Kapitel / Absatz in Dokumentation DENIC-23</b>
212	Did not find DNSKEY RR from request in nameserver response	Kap. 3.6.2, Absatz 2
213	No DNSKEY RR from request found in nameserver response	Kap. 3.6.2, Absatz 2
214	Some nameservers not reachable via EDNS0 with sufficient packet size	Kap. 3.6.5, Absatz 2
215	Timeout after switching from UDP to TCP - switch to TCP due to truncation	Kap. 3.6.5, Absatz 3
216	No visible DNSKEY found signing the DNSKEY RR obtained in response	Kap. 3.6.3
217	No visible DNSKEY found in signing directly or indirectly the SOA RR obtained in response	Kap. 3.6.4
218	Received invalid answer to a DO-Bit query	Kap. 3.6.5, Absatz 2
219	Unable to retrieve DNSKEY RR with TCP or EDNS0	Kap. 3.6.5, Absatz 3
220	DNSKEY RR has invalid algorithm	Kap. 3.6.1.4., Absatz 4
221	Unknown flags in DNSKEY RR are set	Kap. 3.6.1.1
222	Some nameservers not reachable via EDNS0 because of occurred timeout	Kap. 3.6.5, Absatz 2
223	Timeout after switching from UDP to TCP - switch to TCP due to timeout	Kap. 3.6.5, Absatz 3
224	Some nameservers not reachable via EDNS0	Kap. 3.6.5, Absatz 2
225	Timeout after switching from UDP to TCP	Kap. 3.6.5, Absatz 3
226	DNSKEY RR ECDSA public key has invalid size	Kap. 3.6.1.4, Absatz 5 und 6
227	DNSKEY RR GOST public key has invalid size	Kap. 3.6.1.4, Absatz 7
901	Unexpected RCODE (target, entity, RCODE)	Ohne Bezug zu einem Kapitel. Der RCODE der Antwort ist ungleich NOERROR.
902	Timeout	Ohne Bezug zu einem Kapitel.

<b>Issue-Code</b>	<b>Message</b>	<b>Bezug auf Kapitel / Absatz in Dokumentation DENIC-23</b>
		Timeout bei der Kommunikation mit dem angegebenen Server
903	Timeout with recursive resolver	Ohne Bezug zu einem Kapitel. Interner Timeout – Bitte versuchen Sie es erneut
904	Port unreachable	Ohne Bezug zu einem Kapitel. Der Server ist nicht auf UDP Port 53 zu erreichen.
905	Invalid DNSKEY RR public key - conversion problem	Vergleich des DNSKEY RR fehlgeschlagen - Bitte wenden Sie sich an Business Services
906	Invalid DNSKEY RR DSA public key - conversion problem	Vergleich des DNSKEY RR fehlgeschlagen - Bitte wenden Sie sich an Business Services
907	DNSKEY RR from nameserver response cannot be compared with DNSKEY RR from request - conversion problem	Vergleich des DNSKEY RR fehlgeschlagen - Bitte wenden Sie sich an Business Services
908	TCP connection refused	Ohne Bezug zu einem Kapitel. „TCP connection refused“ vom Server
909	Socket error	Ohne Bezug zu einem Kapitel. Socket kann nicht aufgebaut werden, zum Beispiel weil das Netzwerk nicht erreichbar ist.
999	Unexpected exception	Ohne Bezug zu einem Kapitel. Der Code

<b>Issue-Code</b>	<b>Message</b>	<b>Bezug auf Kapitel / Absatz in Dokumentation DENIC-23</b>
		bedeutet, dass die DNS-Abfrage fehlschlug. Bitte wenden Sie sich an Business Services.



## 7. Dokumenthistorie

Version	Datum	Kapitel	Änderung
0.1	17.11.2009	<Alle>	Dokument erstellt
0.2	23.11.2009	3.3 & 4.6	URL aktualisiert
0.3	24.11.2009	<Alle>	Dokument aktualisiert
0.4	30.11.2009	3.4.2 & 4	Typos
0.5	12.01.2010	1.1, 1.2, 6, 3.4.5, 3.4.6, 5	Update auf NAST-Version 1.0.0
0.6	22.01.2010	4.7	Kapitel hinzugefügt
0.7	29.01.2010	6	Issue-Code 120 hinzugefügt
0.8	10.02.2010	6	Typo bei Issue-Code 107; Issue-Code 117 entfernt
0.9	23.02.2010	6	Issue-Codes 200-206 hinzugefügt
1.0	28.04.2010	1.2 4.8.1 3.4.6 3.4.7	NAST-Version aktualisiert Policy und Dnskey hinzu Kapitel ergänzt Hinzugefügt
1.1	14.06.2010	6	Issue-Codes 200-225, 905-907 hinzugefügt
1.2	06.07.2010	6	Text von Issue-Code 103 und 105 geändert Issue-Code 904 hinzugefügt
1.3	16.09.2010	1.1, 1.3. 4.6.1	Anpassungen für ENUM
1.4	23.5.2011	6	121: Received a truncated response (resolver, answer) 207: Kap. 3.6.1.4 Absatz 4 -> Kap. 3.6.1.4 208: Kap. 3.6.5 -> Kap 3.6.1 209: Kap. 3.6.1.4 Absatz 4 -> Kap. 3.6.1.4 210: Kap. 3.6.5 -> 3.6.1 und 3.6.2 212: Kap. 3.6.2 -> Kap 3.6.2 Absatz 2 213: Kap. 3.6.2 -> Kap 3.6.2 Absatz 2 215: -> Timeout after switching from UDP to TCP - switch to TCP due to truncation 218: Kap. 3.6.2 -> Kap. 3.6.5 Absatz 2 223: -> Timeout after switching from UDP to TCP - switch to TCP due to timeout
1.5	20.6.2011	3.4	Ab v1-3-2 (Siehe NAST-55) sind Änderungen in der Optionen für die CLI vorgenommen worden. 1) '-ri' ist '-i' geworden 2) '-p' ist mit '-D' ersetzt worden 3) '-rp' hiesst jetzt '-p' 4) '-?' entfällt (es bleibt aber '-h' für Help).

---

1.6	20.6.2011	3.4.3	Ergänzt: Eine Eingabe von „-p <Port>“ ist zwingend erforderlich.
1.7	10.03.2014	6	Issue Code 107 überarbeitet
1.8	01.12.2014	3.1, 4.1, 4.7	Systemvoraussetzungen und Informationen zum Logging aktualisiert
1.9	20.08.2015	6	Issue Codes 226 und 227 hinzugefügt