

Anwendung

SDP (vgl. 4566) bietet eine Beschreibung einer Session (z.B. Multimedia Konferenz) mit allen Informationen, die von Clients benötigt werden, um an der Session teilzunehmen:

- ▶ Name und Zweck
- ▶ Zeitraum
- ▶ Medientypen
- ▶ Verbindungsinformationen (Host, Port, Multicast Adressen)
- ▶ Bandbreitenanforderungen

SDP Daten können über verschiedenen Transportprotokolle (z.B. HTTP, EMail mit Content-Type application/sdp, Session Announcement Protocol, SIP) verbreitet werden.

Beispiel (SIP INVITE)

Ein VoIP Client versucht, eine Session zu eröffnen:

```
v=0  
o=username 0 0 IN IP4 10.1.195.159  
s=The Funky Flow  
c=IN IP4 10.1.195.159  
t=0 0  
m=audio 32772 RTP/AVP 97 3 0  
a=rtpmap:0 PCMU/8000  
a=rtpmap:3 GSM/8000  
a=rtpmap:8 PCMA/8000  
a=rtpmap:97 iLBC/8000  
a=fmtp:97 mode=20
```

Beispiel (SIP ACK)

Ein anderer VoIP Client akzeptiert die Session mit leicht veränderten Daten:

```
v=0  
o=depp 123456 654321 IN IP4 127.0.0.1  
s=A conversation  
c=IN IP4 127.0.0.1  
t=0 0  
m=audio 7078 RTP/AVP 3  
a=rtpmap:3 GSM/8000
```

Aufbau einer SDP Nachricht

- ▶ SDP besteht ausschließlich aus UTF-8 Text
- ▶ Aufbau einer SDP Nachricht ist immer
 - ▶ eine Block zur Session
 - ▶ eine möglicherweise leere Liste von Medienbeschreibungen
- ▶ Die Grammatik ist als ABNF (RFC4234) in RFC4566 angegeben und muß strikt befolgt werden.
- ▶ Läßt sich die Nachricht nicht vollständig parsen (unbekannte Felder, falsche Formatierung), ist sie komplett zu verwerfen.

Session Parameter

- v= Protokollversion
- o= Identifikation des Initiators
Name SessionId SessionVersion Netztyp Adresstyp
Adresse
- s= Name der Session
- i= (optionale) Session Information
- u= (optionale) URI der Sesion Information
- e= (optionale) EMali Adresse des Senders
- p= (optionale) Telefonnummer des Senders
- c= (optionale) Verbindungsdaten (üblicherweise im Medienblock)

Session Parameter

- b=** (optionale) Bandbreitenanforderung (üblicherweise im Medienblock)
- ▶ Beschreibung der Übertragungszeit bestehend aus
 - t=** Startzeitpunkt Stopzeitpunkt (als Sekunden seit 1900)
 - r=** (optionale) Wiederholungszeitpunkte
- z=** (optionale) Zeitzeoneninformation (für Offsets zum Tagesanfang bei Wiederholungen)
- k=** (optionale) Schlüssel
- a=** (optionale) Attribute

Medienbeschreibung

- m=** Datentyp (z.B. audio/video), Quellport, Protokoll (z.B.: RTP/AVP oder UDP), zusätzliche Parameter, z.B. RTP Payload Type
- i=** (optional) Titel
- c=** Verbindungsinformationen (optional, falls im Session Block):
 - ▶ Netztyp
 - ▶ Adresstyp (IP4/IP6)
 - ▶ Adresse (IP Adresse / Multicast Adresse)
- b=** (optionale) Bandbreiteninformation(en)
- k=** (optionale) Schlüssel
- a=** (optionale) Medienattribute

Attribute in SDP

Bedeutung und Ursprung finden sich unter

<http://www.iana.org/assignments/sdp-parameters>

Name	Scope	Name	Scope
cat	Session	keywds	Session
tool	Session	ptime	Media
maxptime	Media	rtpmap	Media
recvonly	Either	sendrecv	Either
sendonly	Either	inactive	Either
orient	Media	type	Session
charset	Session	sdplang	Either
lang	Either	framerate	Media
quality	Media	fntp	Media

Einleitung

- ▶ ABNF (Augmented Backus Naur Form) ist eine Möglichkeit, kontextfreie Sprachen zu definieren.
- ▶ Benutzt wird ABNF vor allem in neueren RFCs, um textbasierte Protokolle zu spezifizieren.
- ▶ Die ursprüngliche Version stammt aus RFC733 (EMail Vorläufer, 1977).
- ▶ ABNF erweitert die Backus-Naur Form (BNF) um einfachere Wiederholungsregeln und kompaktere Schreibweise.
- ▶ Aus vergleichbaren Gründen gibt es weitere BNF-Derivate, z.B.EBNF.

Beispiel (aus RFC4566)

IP4-address = b1 3("." decimal-uchar)
b1 = decimal-uchar; less than "224"
integer = POS-DIGIT *DIGIT
POS-DIGIT = %x31-39 ; 1 - 9
; Zahlen 0-255
decimal-uchar = DIGIT
/ POS-DIGIT DIGIT
/ ("1" 2*(DIGIT))
/ ("2" ("0"/"1"/"2"/"3"/"4") DIGIT)
/ ("2" "5" ("0"/"1"/"2"/"3"/"4"/"5"))

Regeln

- ▶ Regeln haben die Form "NAME= ELEMENTE"
 - ▶ NAME ist der Name der Regel, optional in spitzen Klammern (<>), Groß- Kleinschreibung spielt keine Rolle.
 - ▶ ELEMENTE ist eine Folge von Regelnamen oder Terminalsymbolen, verbunden durch Operatoren.
- ▶ Kommentare werden durch ein Semikolon eingeleitet
- ▶ Terminalsymbole werden durch ihre numerische Repräsentation in einem Zahlensystem (d: dezimal, b: binär, x: hexadezimal) angegeben:
 - ▶ CR = %d13 ; ASCII 13
 - ▶ CR = %x0D ; ASCII 13
 - ▶ CRLF = %x0D.0A ; ASCII 13, dann ASCII 10

Alternativ können Zeichenketten in Anführungszeichen gesetzt werden.

Operatoren

ABNF Definiert die Folgenden Operatoren:

- ▶ **Zusammensetzung:** Regelnamen oder Terminalsymbole durch Leerzeichen (Space, HT) getrennt, z.B.:
REGEL = REGEL "a"
- ▶ **Alternativen:** Regelnamen oder Terminalsymbole getrennt durch "/", z.B.:
REGEL = "a" REGEL / "b"
- ▶ **inkrementelle Alternative:** Erweiterung einer Folge von Alternativen durch "/=", z.B.:
REGEL = "A" / "B"
REGEL /= "C"
- ▶ **Zeichenbereiche:** Terminalzeichen inklusive Kodierung, z.B.
ZIFFERN = %x30-39 ; ASCII Zeichen der Ziffern

Operatoren

- ▶ **Gruppen:** Folge von Elementen in runden Klammern, z.B.:
REGEL = ("A" "B") / "C"
- ▶ **Optionale Gruppen:** Folge von Elementen in eckigen Klammern, z.B.
REGEL = ["A"] "B"
- ▶ **Wiederholungen:** Element mit vorangestelltem ANFANG*ENDE, wobei ANFANG und ENDE optional sind. Fehlt ANFANG, wird 0 angenommen, fehlt ENDE, wird ∞ angenommen.
REGEL = *"a" "b" ; beliebig viele a, terminiert durch b
REGEL = 1*"a" ; mindestens ein a
REGEL = *5 ZIFFERN; höchstens 5 Ziffern

Vorrangregeln

Die Liste der Operatoren von höchster zu niedrigster Präzedenz:

1. Zeichenketten, Terminalsymbole, Regelnamen
2. Zeichenbereiche
3. Wiederholung
4. Gruppen, Optionale Gruppen
5. Zusammensetzung
6. Alternativen

Zur Verbesserung der Lesbarkeit sollte im Zweifelsfall ein Gruppe gebildet werden, im Beispiel oben:

REGEL = (“a” REGEL) / “b”

Anwendungen von SIP (vgl. RFC3261)

SIP ist ein Signalisierungsprotokoll auf Anwendungsebene mit den Zielen:

- ▶ Aufbau und Abbau von Sessions, z.B.:
 - ▶ Telefonanrufe
 - ▶ Multimedia Streaming
 - ▶ Multimedia Konferenzen
- ▶ Austausch von SDP Nachrichten
- ▶ Unterstützung diverser Dienste für Endnutzer
- ▶ Unterstützung verschiedenster Infrastruktur (Proxies, Router, ...)

Beispiele

Beispiele für typische Szenarien finden sich z.B. hier:

- ▶ RFC3261 (einfache Beispieldate)
- ▶ RFC3665 (SIP Basic Call Flow Examples, komplette Sequenzdiagramme mit zugehörigen Nachrichten)
- ▶ RFC3666 (SIP to PSTN Dialing)
- ▶ <http://www.tech-invite.com/#mt2>

Transport

- ▶ Alle SIP Implementationen müssen TCP und UDP als Transport unterstützen.
- ▶ Standardport für SIP via TCP und UDP ist 5060.
- ▶ TLS ist optional mit Standardport 5061.
- ▶ Das Nachrichtenformat ist weitgehend identisch mit HTTP 1.1 (RFC2616).
- ▶ HTTP Chunked-Encoding ist nicht erlaubt.
- ▶ Der verwendete Zeichensatz in SIP Nachrichten ist UTF-8.

Transaktionen

Um Request und Response zuordnen zu können, werden auf Client- und Serverseite unterschiedliche Regeln verwendet:

- ▶ Client
 - ▶ Der `branch` im obersten `Via` muß zwischen Request und Response übereinstimmen.
 - ▶ Der `CSeq` Parameter muß übereinstimmen.
- ▶ Server
 - ▶ Der `branch` im obersten `Via` stimmt mit dem `branch` im ersten Request der Transaktion überein.
 - ▶ Der `sent-by` Anteil im obersten `Via` stimmt überein.
 - ▶ Die Methode im Request stimmt überein, ausser bei Methode `INVITE`.

SIP Dialog

Ein Dialog ist eine Beziehung zwischen zwei SIP Endpunkten, die für eine gewisse Zeit aufrechterhalten wird.

Dialoge werden erzeugt durch eine Response mit Statuscode zwischen 100 und 299, deren `To` Header ein `Tag` enthält. Nachrichten werden einem Dialog zugeordnet durch

- ▶ `Call-ID`
- ▶ `From Tag`
- ▶ `To tag`