Håndbog

Til

CPR services

Bilag 5

Logon og generel brug af CPR-services; programme-ringsvejledning

CPR-kontoret
Finsensvej 15, 2000 Frederiksberg
E-post: cpr@cpr.dk. Hjemmeside: www.cpr.dk

Dokumentoplysninger

Titel:	**Håndbog til CPR services. Bilag 5** **Logon og generel brug af CPRservices, programmeringsvejledning.**
Projekt:	
Placering:	O:\GTS\CPR\CPR-Infoportal\Brugerdokument\CPRMOD_servicehaandbog\Ret\Servicehb-Bilag-4-5\Servicehaandbog_bilag_5_000_v0c.doc
Ikrafttrædelse:	Straks
Forfatter:	Peter N Bilby
Godkendt af:	CPR-kontoret: Jørgen Ø. Møller ved underskrift på følgeseddel CSC: Bo Jystrup ved underskrift på følgeseddel
Fordeling:	
Udskrevet:	18-08-2006

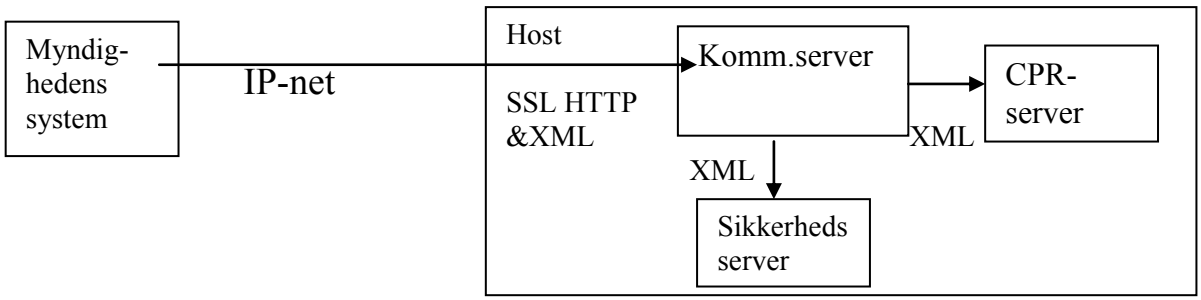
Version/Flgs	Dato	Ændrede sider eller afsnit	Kommentarer
001	09-02-2001	Alle	Første udgivelse
002	15-06-2001	Alle	XML indført
003	23-05-2002	Alle	XML SSL indarbejdet
004/017	20-12-2002	Afsnit 4	Indarbejdet Mulighed for genbrug af sockets (Keep-Alive)
005/023	10-03.2004	Afsnit 3	Ændring af portnr til 683
006/045	18-08-2006	Alle	Tilpasset til HTTP1.1
	04-02--2014	Alle	Tilpasset migrering af CPR drift

Indholdsfortegnelse

1. Indledning

I det følgende beskrives hvordan klienten etablerer forbindelse til og logger på CPR-serveren.

2. Oversigt



Kommunikation

Klientens kommunikation af data i XML-format med CPR-services foregår via en kommunikationsserver på hosten, der lige som klienten er tilsluttet IP-nettet.

Hemmeligholdelse SSL

Kommunikationen foregår på basis af en SSL-socket connection.
På hostsiden supporteres:

Krypteringsmetode til nøgleudveksling:	RsaKeyX 2048
Hashmetode til sikring af integritet:	SHA1
Krypteringsmetode til hemmeligholdelse af gavndata:	Aes128

Headere

HTTP-data udgør en header til de efterfølgende data i XML-format.

På IP-nettet er der yderligere en IP-header og en TCP-header foran HTTP-headeren.

IP	TCP	HTTP	XML

IP

IP-headeren anvendes til at skabe forbindelsen mellem en klient og en kommunikationsserver på hosten, som giver adgang til CPR-serveren og til Sikkerhedsserveren . Hosten er i IP-nettet identificeret ved sin IP-adresse/navn, og kommunikationsserveren lytter på en af hostens porte.

HTTP

HTTP-data anvendes til klientens kommunikation med kommunikationsserverens egne services vedrørende:

- Valg af service (Sikkerhedsservicen Logon og CPR-services).
- Redirection af IP-adresse/navn og port-nr.
- Fejlmeddelelser vedrørende kommunikationen.

XML ved Logon

XML-formatet anvendes af Sikkerhedsserveren. f.eks. ved Logon og ved kommunikation med CPR.

Karaktersæt

Det bemærkes, at kommunikationsserveren anvender karaktersættet ISO-8859-1. En pc kan i praksis være sat op til at anvende karaktersættet cp=850, men der må kun anvendes de karakterer, der er fælles med ISO-8859-1.

3. Etablering af socket-forbindelse

Kald af CPRs server

Inden dialogen kan etableres, skal der over IP-nettet etableres en socket forbindelse mellem klienten og et miljø bag CSC's kommunikationsserver.

Miljø	<IP-navn>:<Port>
Produktion	https://gctp.cpr.dk/cpr-online-gctp/gctp:443
Demo	https://gctp-demo.cpr.dk/cpr-online-gctp/gctp:443

I produktion peget IP-navnet på nuværende tidspunkt på IP-adressen 147.29.101.6.
Tilsvarende peges i Demo på 147.29.101.23.

Logon og hver efterfølgende transaktion anvender hver sin socketforbindelse.

4. HTTP-kommunikationen

4.1 Request

	HTTP-headeren består af: - Startlinie Request og response har forskellig opbygning. - Meddelelseslinier Request og response har samme form. - Slutlinie Request og response har samme indhold. Alle linier i HTTP-headeren indeholder som de sidste 2 karakterer CRLF (HEX '0A0D'), der her skrives som ↵.

Startlinie

ORD	**BESKRIVELSE**
<Type af request>	Skal altid være POST
<Blank>	ASCII-karakteren blank
<Sti til service>	Service Sti til service Logon: /cics/dmwg/cscwbsgn/cpr-online-gctp/gctp CPR-Applikation: /cpcacpra/ajou/xyz/cpr-online-gctp/gctp
<Blank>	ASCII-karakteren blank
<Protokol>/<Version>	<Protokol> Skal altid være HTTP <Version> Skal være mindst 1.0
<CRLF>	HEX '0A0D'

Meddelelseslinier

Linieident	**BESKRIVELSE**
Host:	Indeholder som linieværdi: Hostens internetadresse. Dvs. https://gctp.cpr.dk/cpr-online-gctp/gctp
User-Agent:	Indeholder som linieværdi: "CPR/1.0 " Skal forekomme. (Husk blank efter:)
Content-Length:	Indeholder som linieværdi: Længden i bytes af de data i XML-format, der følger efter HTTP-headeren. Skal forekomme. (Husk blank efter:)
Cookie:	Indeholder som oplysning: TOKEN=ZZZAAAAAAAA Ved Logon valideres indholdet derfor ikke. Ved request af applikation er værdien lig med den token, som blev modtaget ved Logon. Syntaksen for værdien af den token, som bliver modtaget ved logon er (11 pos.) 3 Z'er og 8 små bogstaver. Skal forekomme. Dog ikke ved logon (Husk blank efter:)

Slutlinie

Slutlinien indeholder kun værdien ↵

Som alle andre linier afsluttes den sidste meddelelseslinje også med ↵
De sidste byte i headerenn bliver derfor HEX '0A0D0A0D'

XML

Efter slutlinjen i HTTP-headeren følger XML-delen. Den består af en XML-header, evt commandstatements samt blokken med oplysningen om namespace (xmlns), hvori gctp-blokken befinder sig.

Eksempel

```
POST /cpr-online-gctp/gctp HTTP/1.1↵
Host: gctp.cpr.dk
User-Agent: CPR/1.0↵
Content-Length: 420↵
Cookie: Token=ZZZXXXXXXXX↵
↵
  <?xml version="1.0" encoding="ISO-8859-1"
standalone="yes"?>
<root xmlns="http://www.cpr.dk"><Gctp v="1.0">
…
</Gctp></root>.
```

NB! CPR-klienter skriver XML-strengen som en linje uden ↵ og ekstra blanke. Da der anvendes XML er det dog ikke at krav, at andre også følger den regel.

4.2 Response

Som svar på requestet, returneres et response. Det er opbygget på stort set samme måde som requestet.

Det bemærkes, at der skal reageres på bl.a. redirection.

Startlinie i response

ORD	**BESKRIVELSE**
<Protokol>/<version>	<Protokol>: HTTP <Version>: F.eks.: 1.1 Hostens version.
<Returkode>	HTTP-returkode, hvis requestets kommunikation med kommunikationsserveren gik godt (200) ellers fejl. Der anvendes de i HTTP standardiserede returkoder, samt de supplerende returkoder om Logon, der fremgår af afsnit 3.2.1.
<Returtekst>	Tilhørende standardiseret HTTP-returtekst. Ved returkode 200 er returteksten: OK
<CRLF>	HEX '0A0D'

Det bemærkes, at der efter et ord kan komme et mindre men vilkårligt antal blanke.

Meddelelseslinier i response. De kan komme i andre rækkefølger end den viste og ikke alle behøver at forekomme.

Linieident	**BESKRIVELSE**
User-Agent	Opbygget som ved requestet.
Content-Length	Opbygget som ved requestet. Dvs: Indeholder som linieværdi: Længden i bytes af de data i XML-format, der følger efter HTTP-headeren.
Content-Typ	Content-Typ :text/xml
Pragma	Pragma: no-cache
Date	Date: day, dd month, yyyy hh:mm:ss Timezone Tidspunktet er valgt af hensyn til opdateringern af andre programmer.
Expires	Expires: day, dd month, yyyy hh:mm:ss Timezone Tidspunktet er valgt af hensyn til opdateringern af andre programmer.
Cookie	Strukturen muliggør hostens levering af oplysninger om token og om redirection. Disse

	oplysninger er kun til stede i linien, når værdien er ny eller når den ændres. Struktur af linien er: Set-Cookie:<Oplysningsliste > Enten <Oplysningsliste>::= <Oplysning> eller <Oplysningsliste>::= <Oplysning>; <Oplysningsliste> Token: <Oplysningsliste >::= Token=<token>; Path=/ Er <token> = ZZZzzzzzzz er token ikke accepteret af hostens sikkerhedssystem. Redirection: <Oplysning> ::= Ipaddr=<ipaddres> <Oplysning> ::= Port=<port> <Oplysning> ::= Path=<sti til applikation>
Via:	Via: <Navn på proxyserver> <Navn på proxyserver>::= HTTP/1.0 imainframeweb.csc.dk (IBM HTTP Server)
Connection:	Connection: Keep-Alive Kun når klienten modtager denne meddelelse må klienten forsøge at genbruge den anvendte socket. Der er ingen garanti for, at hosten opretholder viden om denne socket når klienten forsøger at anvende den. For visse applikationer , der ikke håndterer Keep-Alive korrekt kan man prøve at tilføje http-headeren ”`Connection: close`”
Proxy-Connection:	Proxy-Connection : Keep-Alive
<CRLF>	HEX ’0A0D’

Slutlinie

Slutlinien indeholder kun værdien ↵

Som alle andre linier afsluttes også den sidste meddelelseslinje med ↵
De sidste 4 byte i headren bliver derfor: HEX ’0A0D0A0D’

Eksempel

```
HTTP/1.1  200 OK↵
Pragma: no-cache↵
Date:Mon, 21 Mar 2002 15:31:31 GMT↵
Content-Length:2443↵
Content-Type: text/xml ↵
```

```
Expires: Mon, 21 Mar 2002 00:00:02 GMT↵
↵
```
Her følger så de ovennævnte 2443 byte data i XML-format

4.2.1 Data i XML-format ved Logon

XML-dokumenter indledes med en XML-header. Herefter kan der følge Commandstatements med URL til XML-Schema eller DTD, der beskriver strukturen og dens data. Herefter følger en blok med oplysning om ejerskab til GCTP-blokken og dens data. GCTP-blokken er placeret inden i blokken om ejerskab.

Eksempel

```
<?xml version="1.0" encoding="ISO-8859-1"
standalone="yes"?>
<root xmlns="http://www.cpr.dk">
   <Gctp v="1.0">
      …
   </Gctp>
</root>
```

Returkoder

CPR har defineret returkoder fra hostens sikkerhedssystem.
De anvendes i XML-data.

returKode	**Tekst**
900	Signon successful
901	Token kendes ikke
902	Bruger-id er ikke defineret i sikkerhedssystemet
903	Bruger-id er inaktivt i sikkerhedssystemet
904	Ugyldig Bruger-id indtastet
905	Ugyldig kodeord indtastet
906	Dit kodeord er udløbet
907	Begge kodeord skal være ens
908	Det nye kodeord er ikke gyldigt
999	Implementation error

Almindelig Logon

For at blive logget på systemet, skal brugerid og password sendes som data i XML-format med et POST-request.

Eksempel

```
POST /cics/dmwg/cscwbsgn HTTP/1.1↵
User-Agent: CPR/1.0↵
Content-Length: 75↵
↵
<?xml version="1.0" encoding="ISO-8859-1"
standalone="yes"?>
```

```
<root xmlns="http://www.cpr.dk"><Gctp
v=”1.0”><Sik function= "signon" userid="x…x"
password="xxxxxxxx"/></Gctp></root>
```

NB! CPR-klienter skriver XML-strengen som en linje uden ↵ og ekstra blanke. Da der anvendes XML er det dog ikke at krav, at andre også følger den regel.

Skift af password

Eksempel

Når bruger skal skifte password, er HTTP-delen som ovenfor, mens data i XML-format er som følger:

```
<?xml version="1.0" encoding="ISO-8859-1"
standalone="yes"?>
<root xmlns="http://www.cpr.dk"><Gctp
v=”1.0"><Sik function="newpass"  userid=”x…x”
password=”xxxxxxxx” newpass1=”xxxxxxxx"
/></Gctp></root>
```

Svar når det går godt

Eksempel på svaret fra serveren efter Logon og Skift af password:

```
HTTP/1.1  200 OK↵
Content-Length:64↵
Expires: 0↵
Pragma: no-cache↵
Content-Type: text/xml ↵
Set-Cookie: Token= ZZZabcdefgh; Path=/↵
↵
```

Manglende token

```
<?xml version="1.0" encoding="ISO-8859-1"
standalone="yes"?>
<root xmlns="http://www.cpr.dk"><Gctp
v=”1.0"><Sik><Kvit r= ”returKode” t=”Signon
udført” v="900"/></Sik></Gctp></root>
```

Er der ikke (efter Logon) medsendt en token i requestet, eller er der i requestet sendt en udløbet token, kan man f.eks. få returneret følgende data i XML-format:

```
<?xml version="1.0" encoding="ISO-8859-1"
standalone="yes"?>
<root xmlns="http://www.cpr.dk"><Gctp
```

```
v=”1.0”><Sik><Kvit r=”returKode” t=” Token kendes
ikke” v="901"/></Sik></Gctp></root>
```

HTTP-fejlkoden er stadigvæk "200 OK", idet selve kommunikationen gik godt.