

**Forsikring
& Pension**

**Guide for online
Skadehistorik
erhverv, landbrug
og motor**

**Bilag 1
Rest Api**

Version 2.0 Final

Dokumentoplysninger

Titel:	Guide for online Skadehistorik, Bilag 1 Rest Api
Projekt:	EDI kontorets branchekoordinerede dataudveksling
Forfatter:	Morten Lassen, F&P IT-afdelingen
Bidragydere til dokumentet:	
Godkendt af:	Mette Ellermann Jespersen, F&P EDI-kontoret
Dokumentansvarlig:	Mette Ellermann Jespersen, F&P EDI-kontoret
Fordeling:	EDI kontoret, Forsikring & Pension Udleveres til interessenter i dataudvekslingen
Bemærkning:	Dokumentet kan rekvireres hos Forsikring & Pension

Ændringslog

Version	Dato	Forfatter	Ændrede sider eller afsnit
2.0 Draft B	24.09.2018	MLA	Første udgave
2.0 Draft C	22.10.2018	MLA / ADI	Rettelser efter arbejdsgruppemødet den 12. oktober 2018
2.0 Draft C	06.12.2018	MLA / ADI	Rettelser efter teknikermødet den 19. november 2018 samt at Web-løsningen udgår. Beskrivelse af JWT tokens
2.0 Draft D	Juni 2019	MLA / MEJ	Rettelser efter testperiode
2.0 Draft E	September 2019	MLA / MEJ	Rettelser op til implementering
2.0 Final	31. oktober 2019	MLA / MEJ	Rettelser op til implementering

Forkortelser og definitioner:

https	En krypteret udgave af http som benyttes til datakommunikation over internettet
WS, Webservice	Teknologi til at udveksle data via Internettet
Online-selskab	Selskabet udveksler online via Api (integreret)
Web-selskab	Selskabet udveksler via WebEDI (web baseret)
Api	Applikations Program Interface
Rest	Representational State Transfer
Rest Api	Rest baseret Program interface, der benytter http forespørgsler til dataudveksling vha. GET, PUT, POST og DELETE
Token	En dynamisk nøgle/adgangsbillet der identificerer afsenderen
OAuth2	En åben standard for at give adgang til data
JSON	JavaScript Object Notation. Et tekstbaseret format til dataudveksling
JWT	JSON Web Token
JWS	JSON Web Signature

Referencer:

DateTime datatyper følger ISO 8601 standarden

CCYY-MM-DDThh:mm:ss[Z|(+|-)hh:mm]. For eksempel 2016-12-31T20:00:01.

Se https://en.wikipedia.org/wiki/ISO_8601

Datoer uden værdi angives med værdien null

Beskrivelse af Rest Api

https://en.wikipedia.org/wiki/Overview_of_RESTful_API_Description_Languages

Beskrivelse og anbefaling til ClientId og ClientSecret

<https://www.oauth.com/oauth2-servers/client-registration/client-id-secret>

Indholdsfortegnelse:

1.	Indledning	5
2.	Flowdiagram	5
3.	Sikkerhed	6
	Autorisation	6
4.	ClaimService Api	11
	DEMO	11
	Produktion	11
	Online dokumentation	11
	Operationer	13
	Status	13
	Companies	14
	CloseMessage	16
	OpenMessage	16
	HistoryRequest	17
	Serverkontroller	22
	Returkoder og -tekst	23
5.	Selskab Api	24
	Forudsætninger	24
	JWT tokens	24
	Operationer	26
	Status	26
	HistoryRequest	26
	Returkoder og -tekst	27
	Rest Api med Oauth2 skabelon	27
6.	Test	27
	Teknisk testrapport	27
	Krav til test	28
	Api kald til FP-server	28
	Selskabets eget Api	28

1. Indledning

Dette dokument beskriver Rest Api for online skadehistorik.

Rest Api'et er en afløsning til den nuværende SOAP Webservice, som benyttes ved skadehistorik på erhverv og landbrug samt andre ordninger. Rest Api er nemmere at integrere, og er mere dynamisk en SOAP. Endvidere er den ikke så restriktiv som SOAP.

I forbindelse med overgang til Rest Api vil den gamle SOAP Webservice blive udfaset. Det nye Rest Api understøtter både eksisterende og fremtidige branchegrupper på skadehistorik.

Til at sikre Rest Api kaldet benyttes en standardiseret OAuth2 arkitektur, som udsteder en Access Token også kaldet en "adgangsbillet". Denne access token giver selskabet adgang til at kalde Skadehistorik Api'et.

Rest Api returner en http status 200/OK, hvis operationen gik godt. Returdata sendes i http body som en json streng. Ved eventuelle fejl returneres en standard http statuskode – se afsnittet om returkoder – side 23.

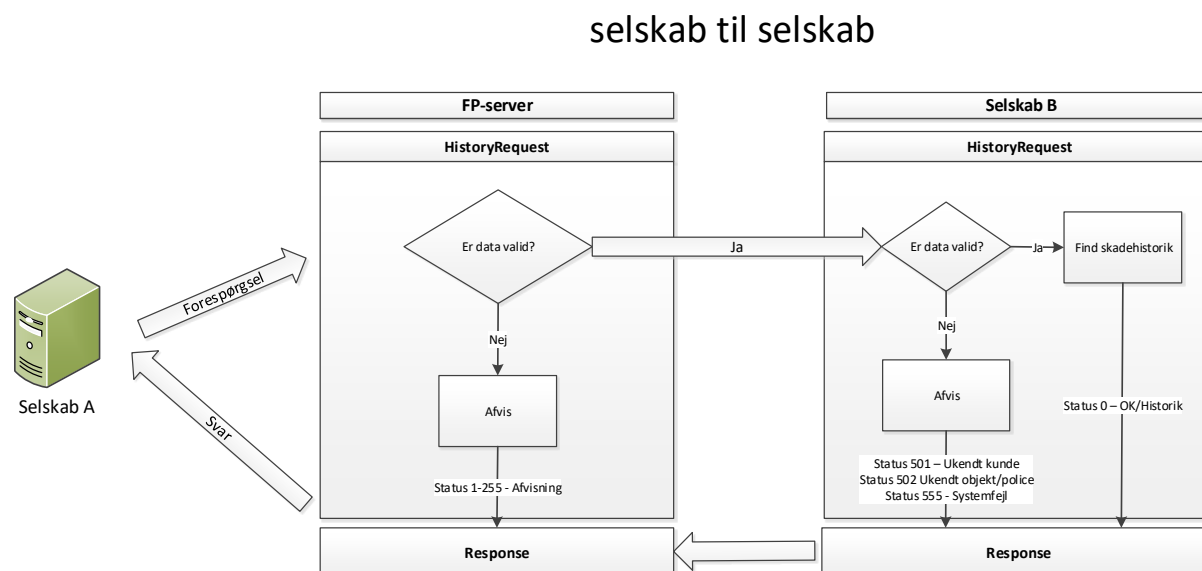
Beskrivelsen af access token og sikkerheden generelt fremgår af kapitel 3.

2. Flowdiagram

I det følgende beskrives flow mellem selskab og Forsikring & Pensions FP-server.

Før selskabet kan kalde Skadehistorik Api'et skal det hente en access token, jf. afsnit 3. Denne token udstedes af Forsikring & Pensions autorisationsserver på baggrund af en klient-id og en klient nøgle, som er unikt for selskabet.

Selskab A sender en forespørgsel ved kald af operationen **HistoryRequest** på FP- serveren. Serveren validerer forespørgslen og afviser ved fejl.



FP serveren sender forespørgslen videre til selskabs Rest Api via operationen **HistoryRequest**. Det svarende selskab B skal svare med det samme, og svaret fra selskabet sendes retur til selskab A. Se afsnit 5.

3. Sikkerhed

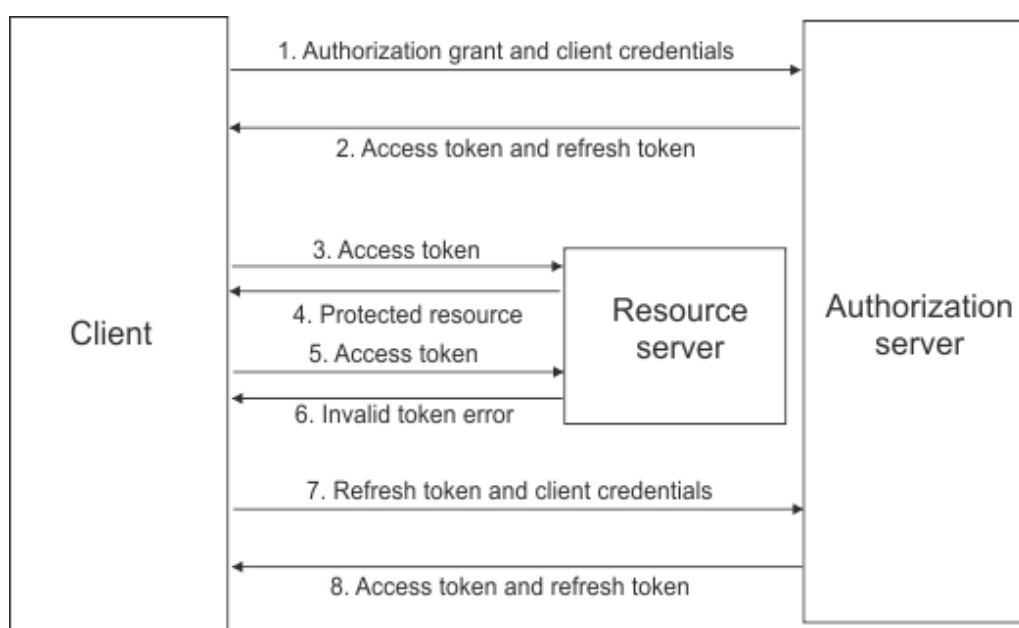
Forsikring & Pensions Skadehistorik Rest Api er beskyttet med flg. teknologier:

- https / SSL / TLS 1.2 der sikrer en krypteret dataforbindelse
- ip whitelist der forhindrer uautoriseret adgang
- Access token der identificerer selskabet

Selskabet får tildelt en klient-id og en klient nøgle, som skal benyttes ved tildeling af en access token – se næste afsnit. Klient nøglen er hemmelig.

Autorisation

Access Token benytter OAuth2 standarden og flowet er vist herunder.



Access token

Før selskabet kan kalde en operation på Forsikring & Pensions Rest Api, skal det anmode om en access token. Denne token udstedes af EDI autorisationsserveren på baggrund af en klient-id (ClientId) samt en klient nøgle (ClientSecret). Sammen med access token returneres også en refresh token, som skal benyttes til at hente en ny access token, når denne er udløbet.

Access token har en levetid på 20 minutter, og skal angives i alle efterfølgende kald til serverens Rest Api.

Access tokens har sin egen livscyclus og der kan derfor være udstedt flere aktive tokens til samme selskab, hvis selskabet har kaldt flere gange samtidigt.

ClientId og ClientSecret

ClientId er unikt for selskabet og genereres af FP-serveren. ClientId er en 32 karakterer hex streng, og kan ses i selskabsadministrationen.

ClientSecret er en hemmelig kryptografisk genereret streng, som kun er kendt af selskabet og autorisationsserveren. Da autorisationsserveren gemmer hash værdien af denne streng, er det ikke muligt at få oplyst ClientSecret efter den er genereret. Selskabet kan selv generere en ny ClientSecret. ClientSecret er valid i 365 dage, og skal skiftes af selskabet inden udløb. Der sendes automatisk en reminder til Api Administratoren 30 dage før udløb.

Anmodning om en token sker på adressen

<https://demoedi.forsikringogpension.dk/authserver/oauth/token>

Fig. oplysninger skal angives som POST parametre:

Grant_type skal have værdien `client_credentials`

Scope skal have værdien `claimhistory`

Client_id og Client_secret

Eksempel på anmodning om en access token:

```
POST /authserver/oauth/token HTTP/1.1
Host: demoedi.forsikringogpension.dk
grant_type=client_credentials
&client_id=a18344f95c694272b6203b35a1ffe059
&client_secret=NPklLbBdD08yymkCABapH3Af1ue17I2gZEKpjYXYhs2
&scope=claimhistory
```

Ved korrekt klient id og -nøgle returnerer serveren en access token og en refresh token.

Eksempel på svar:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache
{
  "access_token": "Vv7gPtgWE4YQ6JqgweOv5k_OFouRsT6Y7J3BKgxClTotPCFSbSUrdT7ieqBN
0vuc1jtgZDk1m1FI7qwn5I4Kh_AcVtOnvJYPZFDViMHjUf07R16cLxPe2VPrc_peGsXQbFncjtd6S1HEGNw
7Zo6eeNkQY1t2ga97PakAWmwAiXOdnjnUsxcbDoG_W8bnQBNEkPEGetXHtvQ9V21hLoxeoL4UQ6pe3Xyrf
uAx-ZmpEvIZ0wKAtdWTSaebWXHnudYJwtXoAA9IHPfSGJWHdD92vIgljxrNEj7Ln6ycW7ICrRIUk2Fxpik6
woPit4szcCdic1_nhVYnySzdjBM7wHeRv-2cV4-IIvz_6u4i7od_zxjh6o1m1pfGoy9-e-5vtXOQUA",
  "token_type": "bearer",
  "expires_in": 1200,
  "refresh_token": "x-GAzugdkRSQUh9akGI1ROYEe9Is0uAE7URgd2P6ezigovhbccpGNggYHA4r
IO_H9aA3Ue7JqHpakl4GtaQbK2RfWEM8kdeUCIMQ6am4p5ASJl8ktxL81-qq7iIGQ1b4pj9sDu5Ddov7Vv
6sZbVbIkJmyx2JfaLuvVElujIrmLE8AGdtFF1340fr9-87r-R7sZaBrXi1qp0AVxTlRPLGLA9A0On5hwlQ
6-qXangZAP6tq_sIbhEp5VnAIbhfPrHMA6h81NKA-MwAs2fx14oRtjP7W1N6oNPrsh-zlsG82HHq3yfbY5
B07WYLVJBteWQsa525s1CHhkVqofURm57i60tfcKdYOReW0ZOrnyStmcyHzWC_D_q33argKWfwxE33Q"
}
```

Access token (`access_token`) skal angives i http headeren ved alle efterfølgende kald

Authorization: Bearer <token>

Access token udløber efter en kort tidsperiode som er bestemt af udstederen. I ovenstående eksempel udløber (`expires_in`) den efter 1200 sekunder hvilket svarer til 20 minutter.

Eksempel på kald af operationen status:

```
GET /api/v1/claimservice/claimhistory/status
Host: demoedi.forsikringogpension.dk
Authorization: Bearer Vv7gPtgWE4YQ6JqgweOv5k_OFouRsT6Y7J3BKgxClTotPCFSbSUrdT7ieqBN
0vuc1jtgZDk1m1FI7qwn5I4Kh_AcVtOnvJYPZFDViMHjUf07R16cLxPe2VPrc_peGsXQbFncjtd6S1HEGNw
7Zo6eeNkQY1t2ga97PakAWmwAiXOdnjnUsxcbDoG_W8bnQBNEkPEGetXHtvQ9V21hLoxeoL4UQ6pe3Xyrf
uAx-ZmpEvIZ0wKAtdWTSaebWXHnudYJwtXoAA9IHPfSGJWHdD92vIgljxrNEj7Ln6ycW7ICrRIUk2Fxpik6
woPit4szcCdic1_nhVYnySzdjBM7wHeRv-2cV4-IIvz_6u4i7od_zxjh6o1m1pfGoy9-e-5vtXOQUA
```


Eksempel på svar ved korrekt angivet token:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache
{
  "ResultDate": "2018-09-25T13:44:59.5565937+02:00",
  "ResultCode": 0,
  "ResultText": "Forsikringsselskab til test #1"
}
```

Eksempel på svar ved udløbet eller forkert token:

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache
{
  "Message": "Authorization has been denied for this request."
}
```

Refresh token

Refresh token har en levetid på 48 timer og fornyes hver gang der hentes en ny access token. Når access token er udløbet skal selskabet hente en ny access token ved at angive refresh token i requestet.

Flg. oplysninger skal angives som POST parametre:

Grant_type skal være refresh_token

Client_id er den samme som ved anmodning om en access token

Refresh_token skal have værdien for refresh_token som blev returneret ved anmodning om en access token.

Eksempel på anmodning om en token:

```
POST /authserver/oauth/token HTTP/1.1
Host: demoedi.forsikringogpension.dk
grant_type=refresh_token
&client_id=a18344f95c694272b6203b35a1ffe059
&refresh_token=x-GAzugdkRSQUh9akGI1ROYEe9Is0uAE7URgd2P6ezigovhbccpGNnggyHA4r
IO_H9aA3Ue7JqHpakl4GtaQbK2RfWEM8kdeUCIMQ6am4p5ASJl8ktxL81-qq7iIGQ1b4pj9sDu5Ddov7Vv
6sZbVbIkJmyx2JfaLuvVElujIrmLE8AGdtFF1340fr9-87r-R7sZaBrXi1qp0AVxTlRPLGLA9A0On5hw1Q
6-qXangZAP6tq_sIbhEp5VnAIbhfPrHMA6h81NKa-MwAs2fx14oRtjp7W1N6oNPrsh-zlsG82HHq3yfbY5
B07WYLVJBteWQsa525s1CHhkVqofURm57i60tfcKdYOReW0ZOrnyStmcyHzWC_D_q33argKWfwxE33Q
```

Ved korrekt klient id og refresh token returnerer serveren en ny access token og en ny refresh token.

Eksempel på svar:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
```

```
Pragma: no-cache
{
  "access_token": fo1BGSAhfudG46-gFmNHWAzA6ZfqBkMfVzWcoVn4uniGeQLJ50YjIs02ou6sJE
SepOhaTSlAzdRY8w8S9of7s2jzF5CZvjU6vGb1J6yy1m7QLLdg2kirrO70R6oXK_ygCxfue68zhFqEu0E
Kijvgpjz6Sp0NT-go-dPKfk113CcGD3Pgzx77R3x2HaAKc_bupbTwBYjcZtMSvKH1-Ac5bVGPPlik8n5F
YpWLgbWLLMRK7ejifq1BiBsiAq4koQEct1GQAKpGacRNu0twlhkk1NOFE8jf3_cOHfI3zI-t-QZhL-UDGM
Du32TftniZjiCA5ProNubQW87TkjjgRQEA9I525dmeFCfkg_-Mz5tirQfn128QHIXNduJu3LjluY9otUA",
  "token_type": "bearer",
  "expires_in": 1200,
  "refresh_token": "SIHDB5pdaBuofHRLlvyBrNdfnMxDkYq6kBXb_dV8-mm28TTZgYj-x6yLs49
oMjbr3g2h89uSoiGHi7K1ZUjVLHu0UikrXW2o15ejfJ_ad2H_Bdboxyrd1-yzKKxpQKJK3PFb0N583UQD
qyyHpA7CtTJ_sdC-Bn07sYCctez-rJSH1zTBKq5eCIx7yG2ZTK0sE6kBvFaE5mz8QgpbneQrxwIiqvxVR
9KAR2-sjOaD6ZasiPXDXMe5duCo1wsWY0pYqOUIWgYU19v1guYQANrXGqOFbNNO4QclQfdws_joC1RKuR
FsesL4nfJoNvFFGIQdvEgZGG6sj-sr9rE3oER6s1dsLoHDQyY-vKTXRk0h0BN1U-p4k9td3E1QnnIiOzKA"
}
```

Hvis både access token og refresh token er udløbet, skal der anmodes som en ny access token ved hjælp af grant typen `client_credentials`.

4. ClaimService Api

I det følgende afsnit beskrives Forsikring & Pensions Rest Api for online skadehistorik. Via dette Api kan selskaberne afsende en forespørgsel til et andet selskab. Endvidere er det muligt at hente en selskabsliste samt åbne og lukke ens selskab – se guide for online skadehistorik.

Selskabet skal lave et tilsvarende Rest Api som F&P serveren kan kalde, når forespørgslen sendes videre til det svarende selskab – se side 24 for beskrivelse af selskabets Api.

DEMO

API Endpoint: <https://demoedi.forsikringogpension.dk/api/v1/claimservice/claimhistory>

Ip-adresse: 81.19.248.248

Produktion

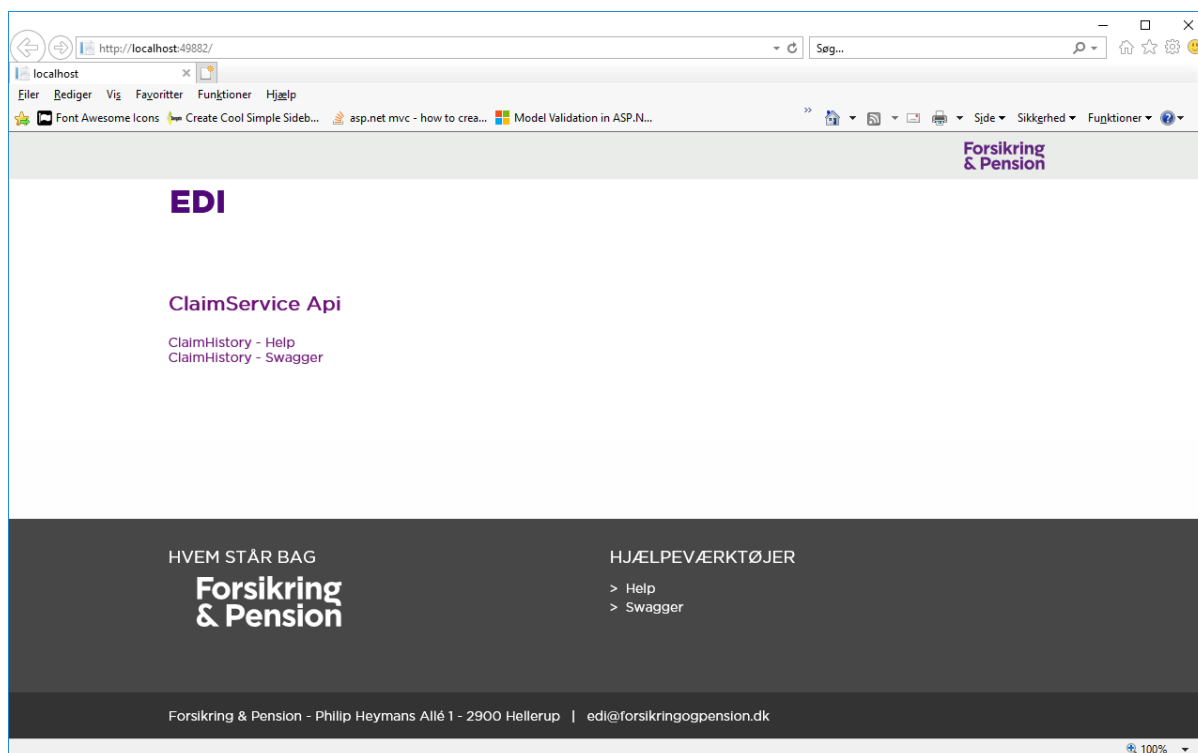
API Endpoint: <https://edi.forsikringogpension.dk/api/v1/claimservice/claimhistory>

Ip-adresse: 81.19.248.247

Online dokumentation

ClaimService Api'et har en online dokumentation, der findes på nedenstående adresse:

<https://demoedi.forsikringogpension.dk/api>



Dokumentationen findes i 2 udgaver:

- **Help**, der beskriver alle kald og datastrukturer
- **Swagger**, der beskriver alle kald og datastrukturer samt integrerer token og test af operationer

Fra Swagger er det også muligt at teste autorisationen, hvor der hentes en access token.

Operationer

I det følgende beskrives operationer, som Rest Api'et udstiller. Som nævnt findes disse også i en online udgave.

ClaimHistory		Show/Hide	List Operations	Expand Operations
GET	/api/v1/claimhistory/status			Returns Company Information
GET	/api/v1/claimhistory/companies			Returns list of Companies
GET	/api/v1/claimhistory/closemessage			Send Company Close Message
GET	/api/v1/claimhistory/openmessage			Send Company Open Message
POST	/api/v1/claimhistory/historyrequest			Returns Claim History Response

[BASE URL: , API VERSION: v1]

Felter der ikke antager nogen værdi kan enten udelades eller udfyldes med null.

Status				
Denne operation benyttes til at teste autorisationen samt til at få status på et selskab. Operationen returnerer en dato/tid, kode, tekst samt åbn/lukke status.				
Input				
Parameter	Type	Krævet	Længde	Beskrivelse
companyId	String	Nej	9	VIR nummer på det selskab, der ønskes status på. Blank=Viser status for eget selskab
Output				
Parameter	Type	Krævet	Længde	Beskrivelse
ResultDate	DateTime	Ja		Dato/tid
ResultCode	Integer	Ja		Retur kode
ResultText	String	Ja	255	Retur tekst. <selskabets navn>
OpenStatus	Boolean	Ja		Returnerer selskabets åben/lukke status. true=åben, false=lukket
GET api/v1/claimservice/claimhistory/status?companyId={companyId}				
Eksempel på response				
<pre>{ "ResultDate": "2018-09-25T13:44:59.5565937+02:00", "ResultCode": 0, "ResultText": "Forsikringsselskab til test #1", "OpenStatus": true }</pre>				

Companies				
Denne operation benyttes til at hente en liste af selskaber, som er tilmeldt skadehistorik. Operationen returnerer en dato/tid, kode, tekst samt en liste af selskaber og de branche/produktgrupper, som pågældende selskab er tilmeldt.				
Input				
Parameter	Type	Krævet	Længde	Beskrivelse
Output				
Parameter	Type	Krævet	Længde	Beskrivelse
ResultDate	DateTime	Ja		Dato/tid
ResultCode	Integer	Ja		Retur kode
ResultText	String	Ja	255	Retur tekst
Companies	List<Company>	Ja		
<Company>				
Name	String	Ja	35	Selskabets navn
Id	String	Ja	9	VIR-nummer
Address1	String	Ja	35	Adresse 1
Address2	String	Nej	70	Adresse 2
PostalCode	String	Ja	4	Postnummer
City	String	Ja	35	By
IndustryProductGroups	List<String>	Ja		Liste af branche/produktgrupper, som selskabet er tilmeldt. Mulige værdier er: 001/001 001/002 001/003 001/004 001/005 001/006 001/007 004/001 004/002 004/003 004/004 004/005 004/006 004/007 004/008 004/009 006/001 006/002 006/003 006/004

				006/005 006/006 006/007 006/008 006/009
GET api/v1/claimservice/claimhistory/companies				
Eksempel på response				
<pre>{ "ResultDate": "2018-09-25T13:46:53.4430232+02:00", "ResultCode": 0, "ResultText": "OK", "Companies": [{ "Name": "Company A ", "Id": "VIR000001", "Address1": "Address 1", "Address2": "", "PostalCode": "4000", "City": "Roskilde", "IndustryProductGroups": ["001/001", "004/001", "006/001"] }, { "Name": "Company B ", "Id": "VIR000002", "Address1": "Address 2", "Address2": "", "PostalCode": "2900", "City": "Hellerup", "IndustryProductGroups": ["001/001", "004/001", "006/001"] }] }</pre>				

CloseMessage

Denne operation benyttes til at informere Forsikring & Pension om, at selskabet lukker for ind- og udgående forespørgsler.

Input	Type	Krævet	Længde	Beskrivelse
-------	------	--------	--------	-------------

Output	Type	Krævet	Længde	Beskrivelse
--------	------	--------	--------	-------------

ResultDate	DateTime	Ja		Dato/tid
------------	----------	----	--	----------

ResultCode	Integer	Ja		Retur kode
------------	---------	----	--	------------

ResultText	String	Ja	255	Retur tekst
------------	--------	----	-----	-------------

GET api/v1/claimservice/claimhistory/closemessage

Eksempel på response

```
{
  "ResultDate": "2018-09-25T13:49:32.220004+02:00",
  "ResultCode": 0,
  "ResultText": "Company Closed succesfully"
}
```

OpenMessage

Denne operation benyttes til at informere Forsikring & Pension om, at selskabet genåbner for ind- og udgående forespørgsler.

Input	Type	Krævet	Længde	Beskrivelse
-------	------	--------	--------	-------------

Output	Type	Krævet	Længde	Beskrivelse
--------	------	--------	--------	-------------

ResultDate	DateTime	Ja		Dato/tid
------------	----------	----	--	----------

ResultCode	Integer	Ja		Retur kode
------------	---------	----	--	------------

ResultText	String	Ja	255	Retur tekst
------------	--------	----	-----	-------------

GET api/v1/claimservice/claimhistory/openmessage

Eksempel på response

```
{
  "ResultDate": "2018-09-25T13:49:32.220004+02:00",
  "ResultCode": 0,
  "ResultText": "Company Opened succesfully"
}
```


HistoryRequest				
<p>Denne operation benyttes til at sende en forespørgsel til et andet selskab, der udveksler skadehistorik. Afsender skal være tilmeldt løsningen og de branche/produktgrupper, der forespørges på. Et online selskab skal svare med det samme.</p> <p>Operationen returnerer en dato/tid, kode, tekst samt svar på forespørgslen.</p>				
Input	Type	Krævet	Længde	Beskrivelse
RequestId	String	Ja	9	VIR-nummer på det selskab, der forespørger
ResponseId	String	Ja	9	VIR-nummer på det selskab, der svarer på forespørgslen
Test	Boolean	Ja		Testmarkering. true = test false eller null = produktion
RequestDate	DateTime	Ja		Dato/tid for forespørgslen
ReferenceNumber	String	Ja	50	Forespørgslens løbe-/referencenr.
CustomerIdQualifier	String	Ja	3	Forsikringstagers identifikationstype. Mulige værdier: CPR eller CVR (CPR er kun mulig på branchegruppe 001 motor)
CustomerId	String	Ja	10	Forsikringstagers CPR- eller CVR-nummer. Format: CVR: nnnnnnnn CPR: ddmmaaånnnn
CustomerName	String	Ja	128	Forsikringstagers navn
ObjectIdQualifier	String	Ja	3	Objekt identifikationstype: POL=policenummer (kun branchegruppe 001 motor) REG=regnr. (kun branchegruppe 001 motor) VIN=stelnummer (kun branchegruppe 001 motor) ALL=samlet
ObjectId	String	Nej	20	Objekt identifikation Udelades eller null ved identifikationstypen ALL
RequestType	Integer	Ja	1	Forespørgselstype: 1=Skadehistorik 2=Ren bonus (kun branche-/produktgruppe 001/001) 3=Bonus og skadehistorik (minimum 001/001)
ConsentForm	Boolean	Ja		Bekræftelse på at der er givet samtykke. True = Ja

				False eller null = Nej
IndustryProductGroups	List<String>	Ja		Liste af branche/produktgrupper der forespørges på
Output	Type	Krævet	Længde	Beskrivelse
ResultDate	DateTime	Ja		Dato/tid for svaret
ResultCode	Integer	Ja		Retur kode – se side 23
ResultText	String	Ja	255	Retur tekst – se side 23
RequestId	String	Ja	9	VIR-nummer på det selskab, der forespørger
ResponseId	String	Ja	9	VIR-nummer på det selskab, der svarer på forespørgslen
ReferenceNumber	String	Ja	50	Forespørgslens referencenummer
CustomerIdQualifier	String	Ja	3	Forsikringstagers identifikationstype. Mulige værdier: CPR = CPR-nummer (kun branchegruppe 001 motor) CVR = CVR-nummer
CustomerId	String	Ja	10	Forsikringstagers CPR/CVR-nummer
CustomerName	String	Ja	128	Forsikringstagers navn
Policies	List <Policy>	Nej		Liste af policer som forsikringstager har eller har haft
<Policy>				
PolicyNumber	String	Ja	50	Policenummer
PolicyStartDate	String	Ja	10	Policens ikrafttrædelsesdato Format: yyyy-mm-dd
PolicyEndDate	String	Nej	10	Policens afgangsdato. Format: yyyy-mm-dd Ved aktiv police skal feltet udelades eller angives til <i>null</i>
Arrears	Bool	Nej		Restance. (kun branchegruppe 001 motor) True = ja, False = nej
IndustryProductGroups	List<String>			Liste af branche/produktgrupper som policen dækker over
Bonuses	List<Vehicle Bonus>	Nej		Liste af bonuser (kun branchegruppe 001 motor)
<VehicleBonus>	Bonus			
Industry ProductGroup	String	Ja	7	Branche/produkt. Format: bbb/ppp = 001/001
RegistrationNumber	String	Nej	7	Registreringsnummer
VINNumber	String	Nej	20	Stelnummer
VehicleType	String	Ja	50	Køretøjstype
ClaimFreeYears	Integer	Ja		Antal skadefri år
LastStepDate	String	Ja	10	Dato for sidste trinskifte Format: yyyy-mm-dd

				Hvis dato ikke kendes skal policens ikrafttrædelsesdato angives.
FixedPremium	Bool	Ja		Fastpræmie. True = ja, False = nej
Claims	List<Claim>	Nej		Liste af skader på pågældende police
<Claim>	Skade			
IndustryProductGroup	String	Ja	7	Branche/produkt hvorpå skaden er sket. Format bbb/ppp.
ProductName	String	Ja	50	Navn på forsikringsproduktet
ObjectIdQualifier	String	Ja	3	Objekt identifikationstype (kun branchegruppe 001 motor): REG=registreringsnummer VIN=stelnummer NA = Ukendt
ObjectId	String	Ja	20	Objekt identifikation Ved NA angives "ukendt"
ClaimLevels	List<string>	Nej		Liste af dækningsniveauer (skal angives for branchegruppe 001 motor). Mulige værdier er: A=Ansvar K=Kasko G=Glas V=Vejhjælp F=Førerplads D=Diverse
ClaimDate	String	Ja	10	Skadedato. Format: yyyy-mm-dd
ClaimType	String	Nej	70	Skadetype
ClaimStatus	Integer	Ja		Status. 0=Åbn, 1=Lukket
ClaimPayed	Integer	Nej		Betalt beløb angivet i øre
ClaimReserve	Integer	Nej		Reserve beløb angivet i øre
ClaimImpact	Bool	Nej		Belastende (kun branchegruppe 001 motor) True = ja False = nej
<pre> POST api/v1/claimservice/claimhistory/historyrequest BODY { "RequestId": "VIR000001", "ResponseId": "VIR000000", "Test": false, "RequestDate": "2018-09-25T12:35:51", "ReferenceNumber": "0123456789A", "CustomerIdQualifier": "CVR", "CustomerId": "11111114", "CustomerName": "Anders And", "ObjectIdQualifier": "ALL", "ObjectId": null, "RequestType": 3, </pre>				

```

"ConsentForm": true,
"IndustryProductGroups": [
  "001/001",
  "004/001",
  "006/001"
]
}

```

Eksempel på response

```

{
  "ResultDate": "2018-09-25T15:35:58.1588783+02:00",
  "ResultCode": 0,
  "ResultText": "OK",
  "RequestId": "VIR000001",
  "ResponseId": "VIR000000",
  "ReferenceNumber": "0123456789A",
  "CustomerIdQualifier": "CVR",
  "CustomerId": "11111114",
  "CustomerName": "Hansen A/S",
  "Policies": [
    {
      "PolicyNumber": "policy 0",
      "PolicyStartDate": "2018-09-25",
      "PolicyEndDate": null,
      "Arrears": false,
      "IndustryProductGroups": [
        "001/001",
        "004/001",
        "006/001"
      ],
      "Bonuses": [
        {
          "IndustryProductGroup": "001/001",
          "RegistrationNumber": "CW12345",
          "VINNumber": "WBS66512436",
          "VehicleType": "Car",
          "ClaimFreeYears": 6,
          "LastStepDate": "2018-09-25",
          "FixedPremium": true,
        },
        {
          "IndustryProductGroup": "001/001",
          "RegistrationNumber": "ZP12345",
          "VINNumber": "AKB66512436",
          "VehicleType": "Car",
          "ClaimFreeYears": 1,
          "LastStepDate": "2018-09-25",
          "FixedPremium": true,
        }
      ],
      "Claims": [
        {
          "IndustryProductGroup": "001/001",
          "ProductName": "Bilforsikring",
          "ObjectIdQualifier": "REG",
          "ObjectId": "CW12345",
          "ClaimLevels": [
            "A",
            "K"
          ],
          "ClaimDate": "2018-09-25",
        }
      ]
    }
  ]
}

```

```
    "ClaimType": "Færdselsuheld",
    "ClaimStatus": 1,
    "ClaimPayed": 245000,
    "ClaimReserved": 0,
    "ClaimImpact": true
  },
  {
    "IndustryProductGroup": "004/001",
    "ProductName": "LandbrugBygningsforsikringen",
    "ClaimDate": "2018-09-25",
    "ClaimType": "Kasko",
    "ClaimStatus": 1,
    "ClaimPayed": 1212000,
    "ClaimReserved": 0,
    "ClaimImpact": null
  }
]
}
```

Serverkontroller

F&P-serveren kontrollerer indholdet i forespørgslen inden den sendes videre til det svarende selskab. Serveren afviser forespørgslen, hvis kontrollerne ikke er overholdt. Der kontrolleres følgende i forespørgslen:

- Klient-id og -nøgle skal være autoriseret
- Forespørgslen kontrolleres for korrekte datatyper
- Test bit må ikke være sat i produktion med mindre der sendes til robot selskab (VIR000000) eller eget selskab
- Spørgende selskab (RequestId) skal være tilknyttet klient-id
- Spørgende selskab (RequestId) skal være kendt på serveren og være tilmeldt skadehistorik
- Svarende selskab (ResponseId) skal være kendt på serveren og være tilmeldt skadehistorik
- RequestId og ResponseId må ikke være identisk med mindre at test bit er true
- Hvis CustomerIdQualifier er CVR skal CustomerId være et validt CVR-nummer (modulus 11 kontrol)
- Hvis CustomerIdQualifier er CPR skal CustomerId være et validt CPR-nummer (modulus 11 kontrol)
- ConsentForm skal være true
- IndustryProductGroups skal antage min. en eller flg. værdier: 001/001-001/007, 004/001-004/009, 006/001-006/009
- Spørgende selskab (RequestId) skal være tilmeldt de branche/produktgrupper, der forespørges på
- Svarende selskab skal være tilmeldt de brancher, der forespørges på
- ReferenceNumber må ikke være brugt tidligere
- Afsender og modtager skal begge være åbne
- Forsikringstagers identifikationstype skal være CPR eller CVR
- Objektidentifikationstype skal være POL, REG, VIN eller ALL
- Hvis Objektidentifikationstype er ALL så må objektidentifikation ikke være udfyldt
- Hvis Objektidentifikationstype er POL, REG eller VIN så skal objektidentifikation være udfyldt
- Forespørgselstype skal være 1, 2 eller 3
- Forespørgselstype 2 (ren bonus) må kun benyttes hvis der alene spørges på branche/produktgruppen af typen motor (001/001)
- Forespørgselstype 3 (bonus og skadehistorik) må kun benyttes hvis der spørges på minimum branche/produktgruppen af typen motor (001/001)

Returkoder og -tekst

I det følgende beskrives værdierne af http StatusCode, ResultCode og ResultText, som returneres ved alle Api kald til FP-serveren.

http statuskoder		Api returkoder	
Code	Message	Code	Text
Følgende returkoder kommer fra FP-serveren			
200	OK	0	OK
400	Bad request	1	Requesting Company is closed
400	Bad request	2	Responding Company is closed
400	Bad request	11	Responding Company is unknown
504	Gateway timeout	12	No answer from responding Company
400	Bad request	22	Requesting and responding company cannot be the same unless test bit have been set
400	Bad request	23	ConsentForm must be true
400	Bad request	24	Customer id is not valid CPR/CVR-number
400	Bad request	25	Invalid industry/product group(s)
400	Bad request	26	Requesting company does not have selected industry/product groups
400	Bad request	27	ReferenceNumber is not unique
400	Bad request	28	Test bit not allowed
400	Bad request	29	Invalid CustomerIdQualifier
400	Bad request	30	Invalid ObjectIdQualifier
400	Bad request	31	ObjectId must be obmitted when IdQua is ALL
400	Bad request	32	ObjectId is mandatory
400	Bad request	33	Invalid RequestType
400	Bad request	34	Invalid RequestType for selected IndustryProductGroups
400	Bad request	35	Responding company does not have selected industry groups
400	Bad request	36	Invalid Request datatype
400	Bad request	252	ReferenceNumber is mandatory
400	Bad request	253	Invalid requestId
401	Unauthorized		
503	Service Unavailable	254	Service currently unavailable
502	BadGateway	251	Error calling responding Company
500	Internal System Error	255	Systemerror from FP-server
Følgende returkoder er sendt videre fra det modtagende selskab			
400	Bad request	501	Unknown customer
400	Bad request	502	Unknown object/policy (only for 001 Motor)
400	Bad request	504	Invalid Authorization to responding Company
400	Bad request	555	Systemerror from Company

5. Selskab Api

I det følgende afsnit beskrives det Rest Api, som selskaberne skal implementere i eget system.

Forudsætninger

Selskabet skal lave et Rest Api som overholder den struktur, som Forsikring & pension har defineret og beskrevet herunder.

Som udgangspunkt skal selskabet implementere OAuth2 service og flow som beskrevet i afsnit 3. Det er op til selskabet at beslutte, om de vil understøtte Refresh token. Det er ikke et krav jf. OAuth2 standarden.

FP-serveren har begrænset understøttelse for kald til selskabernes APIer via JWT tokens (<https://tools.ietf.org/html/rfc7519>), beskrevet i den efterfølgende afsnit (JWT tokens).

Selskaber, der ønsker at implementere OAuth2 skal levere flg. oplysninger til Forsikring & Pension:

- Adressen på selskabets api (EndPoint)
- Token endpoint til at hente en token
- Klient-id som identificerer Forsikring & Pension (Client_Id)
- Klient nøgle (Client_Secret)
- Ip-adresse(r) hvorfra selskabet kalder Forsikring & Pension

Selskaber, der ønsker at implementere JWT tokens skal levere flg. oplysninger til Forsikring & Pension:

- Adressen på selskabets api (EndPoint)
- JWT encryption algorithm (HS256, HS384, eller HS512)
- Brugernavn som identificerer Forsikring & Pension overfor selskabet (Username)
- En klient nøgle (Secret) er genereret på FP-Serveren på basis af den valgt algorithm og sendes til selskabet.

Selskabet skal desuden informere EDI-kontoret om, hvilke branche/produktgrupper selskabet udveksler. Der kan kun forespørges på de branche/produktgrupper, som selskabet er tilmeldt.

JWT tokens

FP-serveren understøtter udstedelse af JWT tokens med følgende specifikationer:

Sikkerhed: JWT med JSON Web Signature (JWS) (<https://tools.ietf.org/html/rfc7515>) og understøttelse af følgende symmetric encryption algorithms: HS256, HS384, eller HS512.

JWS format: [Base64-URL encoded header].[Base64-URL encoded payload].[Signature]

JWS (JOSE) header eksempel: {"alg":"HS256","typ":"JWT"}

JWS payload (Claims Set):

Navn	Type	Definition	Std. værdi
------	------	------------	------------

unique_name	string	Brugernavn angivet af selskabet (custom/unregistered claim type).	
nbf	ndate*	<i>Not before - Identifies the time on which the JWT will start to be accepted for processing. (IETF, rfc7519 afs. 4.1.5)</i>	
exp	ndate*	<i>Expiration time - Identifies the expiration time on or after which the JWT must not be accepted for processing. (IETF, rfc7519 afs. 4.1.4)</i>	
iat	ndate*	<i>Issued at - Identifies the time at which the JWT was issued. (IETF, rfc7519, afs. 4.1.6)</i>	
iss	string	<i>Issuer - Identifies principal that issued the JWT. (IETF, rfc7519 afs. 4.1.1)</i>	fogp
aud	string	<i>Audience - Identifies the recipients that the JWT is intended for. (IETF, rfc7519 afs. 4.1.3)</i>	{Client_Id fra FogP}

* Datoer (ndate) i JWTen claims (nbf, exp, iss) er angivet ifølge definitionen i IETF, rfc7519:

<p>NumericDate</p> <p>A JSON numeric value representing the number of seconds from 1970-01-01T00:00:00Z UTC until the specified UTC date/time, ignoring leap seconds. This is equivalent to the IEEE Std 1003.1, 2013 Edition [POSIX.1] definition "Seconds Since the Epoch", in which each day is accounted for by exactly 86400 seconds, other than that non-integer values can be represented. See RFC 3339 [RFC3339] for details regarding date/times in general and UTC in particular.</p>

JWS Payload eksempel:

```
{
  "unique_name": "testuser1",
  "nbf": 1543239542,
  "exp": 1543240742,
  "iat": 1543239542,
  "iss": "fogp",
  "aud": "b461d8835b804cf1b33f35e49d81e2"
}
```

Token eksempel:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfbmFtZSI6InRlc3Rlc2VyMSIsIm5iZiI6MTU0NTAzNDU1MiwiaWF0IjoxNTQ1MzUyLCJpYXQiOiJlNDUwMzQ1NTIsImZlcyI6ImZvZ3AiLCJhdWQiOiJiNDUyZDg4MzVlODAwY2YxYjMzZjMlZW51NDk0ODFlMiJ9.-qWwqmdDI9Gb6O3BoMTfhTI5UhmYKqS1M2rbDn3Bpmm
```

Selskaber som er tilsluttet denne løsning skal levere flg. oplysninger til Forsikring & Pension:

- Adressen på selskabets api (EndPoint)
- JWT encryption algorithm (HS256, HS384, eller HS512)
- Brugernavn som identificerer Forsikring & Pension overfor selskabet (Username)

En klient nøgle (Secret) er genereret på FP-Serveren på basis af den valgt algorithm og sendes til selskabet.

Kontakt EDI-kontoret for yderligere oplysninger.

Operationer

Selskabets Api skal implementere 2 operationer, som beskrives i det følgende afsnit.

Status				
Denne operation benyttes til at teste autorisationen. Operationen returnerer en dato/tid, kode samt tekst.				
Input				
Parameter	Type	Krævet	Længde	Beskrivelse
Output				
Parameter	Type	Krævet	Længde	Beskrivelse
ResultDate	DateTime	Ja		Dato/tid
ResultCode	Integer	Ja		Retur kode
ResultText	String	Ja	255	Retur tekst
GET <selskabets endpoint>/status				
Eksempel på response <pre>{ "ResultDate": "2018-09-25T13:44:59.5565937+02:00", "ResultCode": 0, "ResultText": "OK" }</pre>				
HistoryRequest				
Denne operation benyttes til at modtage en forespørgsel fra et selskab. Operationen kaldes af FP-serveren på vegne af et andet selskab. Input- og output er identisk med operationen HistoryRequest, som er beskrevet i afsnit 4.				
Kontroller				
<ul style="list-style-type: none"> Client_id og client_secret skal være tildelt Forsikring & Pension Forsikringstager (CustomerId) skal være kendt Der må kun sendes bonus på brc/prd 001/001 Beløb og reserve må ikke oplyses ved ved branche/produkterne: <ul style="list-style-type: none"> 004/006 og 006/006 Arbejdsskade 004/008 og 006/008 Patientforsikring 004/009 og 006/009 Sundhedsforsikring Samt dækningsniveauerne: <ul style="list-style-type: none"> F – Førerplads G – Glas V – Vejhjælp 				

Returkoder og -tekst

I det følgende beskrives værdierne af http statuscode, ResultCode og ResultText, som selskabet skal returneres ved alle Api kald.

http statuskoder		Api returkoder	
code	message	Result Code	Result Text
200	OK	0	OK
400	Bad request	501	Unknown customer
400	Bad request	502	Unknown object/policy (only for 001 motor)
401	Unauthorized		
500	Internal System Error	555	Systemerror from Company

Rest Api med Oauth2 skabelon

Forsikring & Pension laver et Rest Api test projekt med kildekode i C#, som viser hvorledes de forskellige operationer implementeres.

Projektet kan rekvireres ved henvendelse til EDI-kontoret.

6. Test

Test af afsendelse af forespørgsel kan testes ved at sende forespørgslen til et robotselskab på FP-serveren, som genererer et automatisk svar. Selskabet har VIR-nummeret `VIR000000`.

Robotselskabet hedder "Test skadeselskab (autosvar)" – og svarer nogle gange med "Ukendt kunde" og ellers med "Historik".

Ved at anvende nedenstående CVR-nummer i CustomerId, kan robotten svare med forskellige svarmuligheder:

CustomerId	Result
<code>00000nnn</code>	Den pågældende returkode returneres. Fx. <code>00000002</code> = 2 - Responding Company is Closed <code>00000502</code> = 502 - Unknown object/policy
<forkert CPR/CVR-nummer>	24 - CustomerId is not a valid CPR/CVR-number
<korrekt CPR/CVR-nummer>	0 - Tilfældigt svar hver gang Antal policer: 0-4 Antal skader: 0-10

Test af modtagelse af forespørgsel kan testes ved at selskabet sender forespørgslen til sig selv.

Ved alle test skal test markeringen være sat (`true`).

Teknisk testrapport

Testrapporten er et automatiseret værktøj, som giver overblik over de ting og områder, som skal være testet inden selskabet kan gå i produktion.

De forskellige test er opdelt i sektioner, og hver sektion har sin egen status. Sektionerne er:

- Tilmelding
- Udveksling
- Kald til EDI Api
- Kald til Selskab Api
- Antal forespørgsler
- Antal svar

Testrapporten henter en token og kalder selskabets statusoperation. Resten af testen skal udføres af selskabet og rapporten opsummerer status på testen (antal og seneste dato for forekomst). Testrapporten er tilgængelig for API Administratorer.

En vejledning for oprettelse af API administratorer og beskrivelse af testrapporten kan rekvireres hos EDI-kontoret, edi@forsikringogpension.dk

Krav til test

Nedenstående testpunkter skal som minimum være godkendt på DEMO inden et selskab kan gå i produktion:

Api kald til FP-server

- E1. Token – at der kan hentes og refreshes en access token
- E2. Status – er der forbindelse til EDI (bør ikke kaldes oftere end 4 gange/time)
- E3. Companies – hent en selskabsliste (max. én gang i døgnet)
- E4. HistoryRequest – at der kan sendes en forespørgsel til et selskab (robotselskab) og kontrollere at svaret kan modtages

Hvis selskabet vil benytte open/close operationerne skal flg. testes:

- E5. OpenMessage – send åbn status
- E6. CloseMessage – send lukke status

Selskabets eget Api

- S1. Token – at F&P kan anmode om en access token og denne kan refreshes (gælder kun Oauth2 og ikke JWT)
- S2. Status – at der kan sendes en status
- S3. HistoryRequest – at der kan modtages en forespørgsel og gives et svar