

Dokumentation af eLog API

19. august 2011 / Karsten Bracht Nielsen, FD

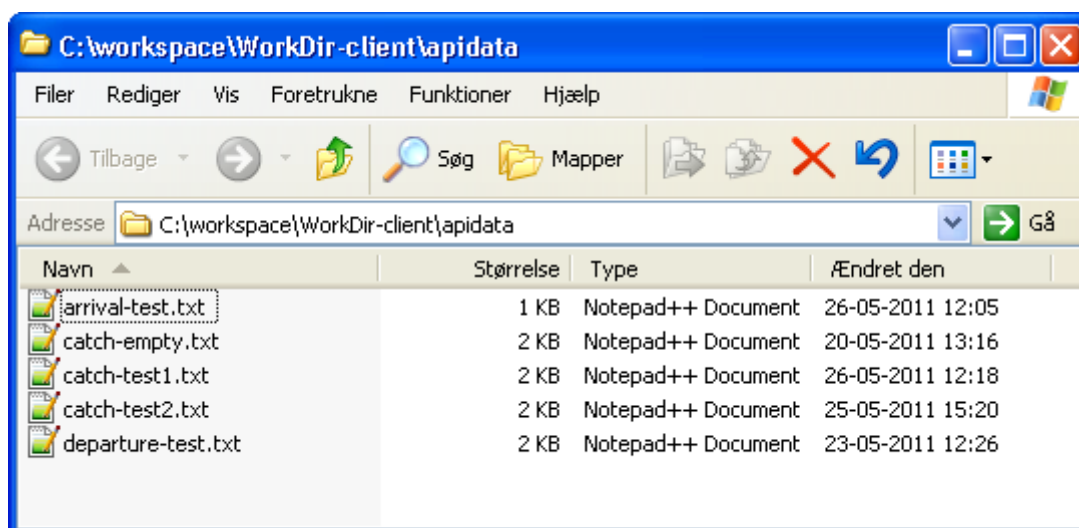
Det dokument er dokumentation af, hvordan web service API'et er implementeret i eLog klient version 5.4.

Netværksopsætning

FD vil ikke være involveret i eller yde support for netværksopsætningen (tildeling af IP-adresser og lignende) på fartøjerne. eLog-klientens API vil lytte på en fast port (for eksempel 8095) på alle netværksinterfaces (binding 0.0.0.0) på PC'en. Der vil ikke blive brugt SSL eller komprimering i HTTP-kommunikationen. Der vil heller ikke være nogen form for authentication eller authorization.

Opsamling af API-beskedes i filer

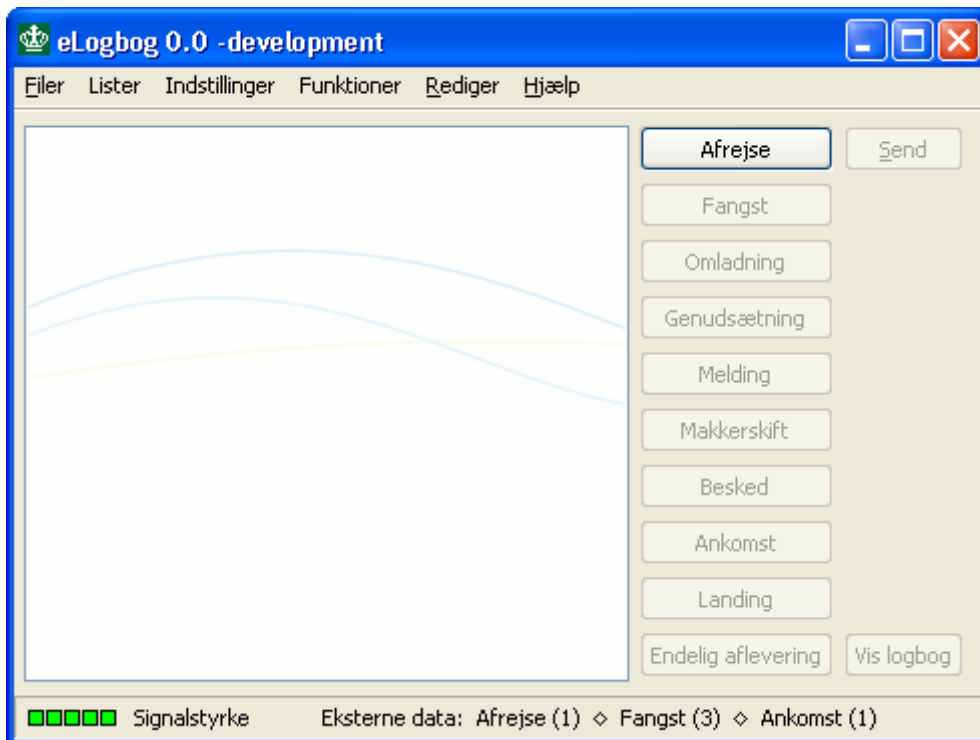
Modtagne API-beskedes bliver gemt midlertidigt i filsystemet.



Filerne bliver gemt med extension ".txt" og ikke ".xml", så det vil være nemt for fiskeren at dobbeltklikke på filen og få indholdet vist i et tekstredigeringsværktøj.

Flow i eLog-klienten

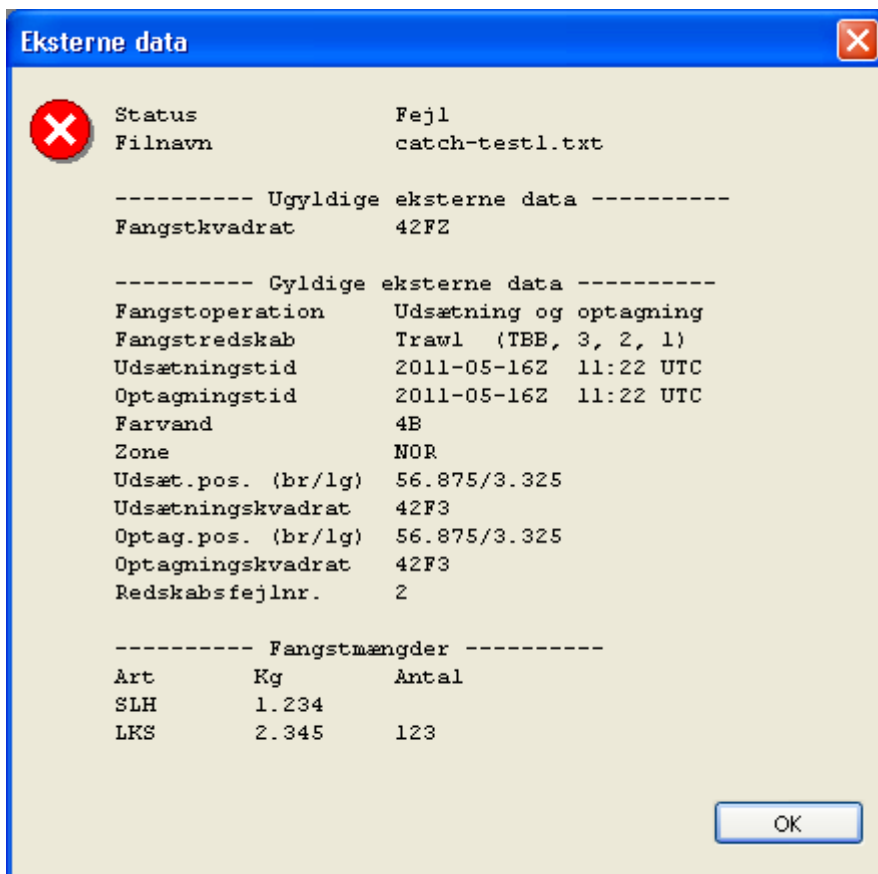
I statusbjælken i bunden af hovedvinduet vil der blive vist hvor mange udestående API-beskedes, som ligger i kø til brug i logbogen.



Statuslinien bliver opdateret hvert 30. sekund, så der kan være en forsinkelse fra man har brugt en API-besked til det rigtige antal bliver vist.

Når der trykkes på Afrejse-, Fangst- eller Ankomst-knappen, og der ligger en API-besked klar, vil dialogen være forudfyldt med API-dataene. Før API-dataene bliver påtrykt, er de normale default-værdi blevet anvendt til at initialisere dialogfelterne. Det vil sige, at hvis der ikke findes API-data for felter, som har default-værdier, så vil default-værdierne stadigvæk blive brugt.

Lige før den udfyldte dialog vises, vises der en besked ny beskedsdialog som opsummerer, hvordan API-dataene er blevet fortolket.



Statusfeltet kan være OK eller Fejl. Filnavnfeltet angiver hvilken API-besked, der er benyttet. Hvis der har været fejl i dataene i API-beskeden, vil fejlene blive listet under overskriften 'Ugyldige eksterne data'.

Alle de API-data, som har kunnet anvendes til at sætte dialogfelterne uden umiddelbare problemer, er listet under overskriften 'Gyldige eksterne data'. At disse API-data har kunnet anvendes til at forudfylde dialogfelterne er ikke ensbetydende med, at de ikke vil kunne give anledning til valideringsfejl, når dialogen gemmes.

Nu vises den forudfyldte dialog.

Fangst [X]

Fangstoplysninger Fangstoplysninger - udvidet Udsætning af passive redskaber Fangst med passive redskaber

Redskabsoplysninger

Fangstredskab: Trawl Redskabsfejl: Ødelagt/revnet net [Rediger liste over fangstredskaber >>](#)

Udsætningsdato og -tid samt position

Dato: 16 / 05 - 2011 Starttid: kl. 11 : 22

Bredde: 56 ° 52 , 5 ' N Længde: 3 ° 19 , 5 ' Ø [Nuværende position](#)

Statistisk rektangel: 42 F3

Fangst dato og -tid samt position

Dato: 16 / 05 - 2011 Sluttid: kl. 11 : 22

Bredde: 56 ° 52 , 5 ' N Længde: 3 ° 19 , 5 ' Ø [Nuværende position](#)

Statistisk rektangel: 42 F3

Fangstoplysninger

Statistisk rektangel: 42

Farvand: 4B [Rediger liste over farvande >>](#)

Øko. zone: NOR [Rediger liste over øko. zoner >>](#)

Ingen fangst

Art	Mængde i kg	Antal
SLH	1.234	
LKS	2.345	123

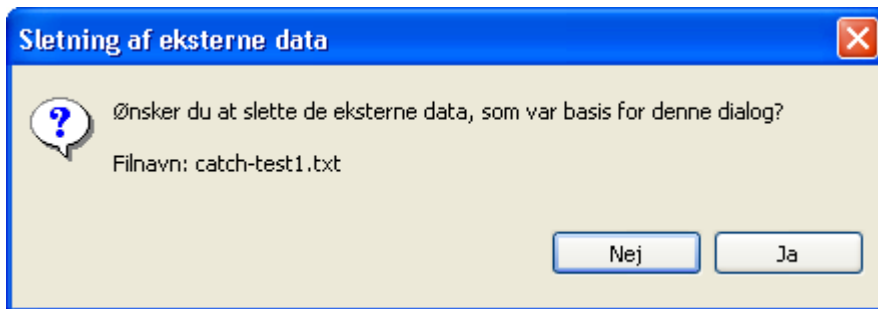
Art	Dagstotal	Total
LKS	0	123 (12)

[Rediger liste over arter >>](#)

Ny Fangst Gem Annuller

Når der trykkes Gem, og valideringen ikke fejler, vil informationerne blive gemt i logbogen som normalt, og den API-fil, som var basis, vil blive slettet automatisk uden notifikation til brugeren.

Hvis der trykkes Annuller i dialogen, vil brugeren blive spurgt om han ønsker at slette API-dataene uden at have anvendt dem i logbogen.



På denne måde kan fiskeren få slettet API-data, som han ikke ønsker at anvende.

XML-format og tegn-enkodning

Alle dokumenter skal være kodet i ISO-8859-1. Det er det format, som logbogs-XML'en (currentlogbook.xml) bliver gemt i.

XML-deklarationen, for eksempel `<?xml version="1.0" encoding="ISO-8859-1"?>`, bliver ignoreret. Eventuelle http-headers bliver også ignoreret.

API-metoder

Bemærk: Det vil ikke være muligt at rette eller slette logbogsinformation via API'et.

API-metoderne bliver i det følgende specificeret via den kommandolinjesyntaks, de kan aktiveres med, hvis man har CURL-værktøjet installeret.

API'et er versioneret med et URL-element: i den første version er det 'v1'.

Da der i første omgang ikke vil blive lavet validering på indkommende beskeder, vil der kun være brug for to typer HTTP-returbeskeder: "200 OK" og "500 Internal server error".

Hent klientversion

```
curl -X GET http://127.0.0.1:8095/elog/v1/version/client
```

Eksempel på resultat: 5.3. Bemærk: ingen xml-indkapsling.

Hent den aktuelle logbog

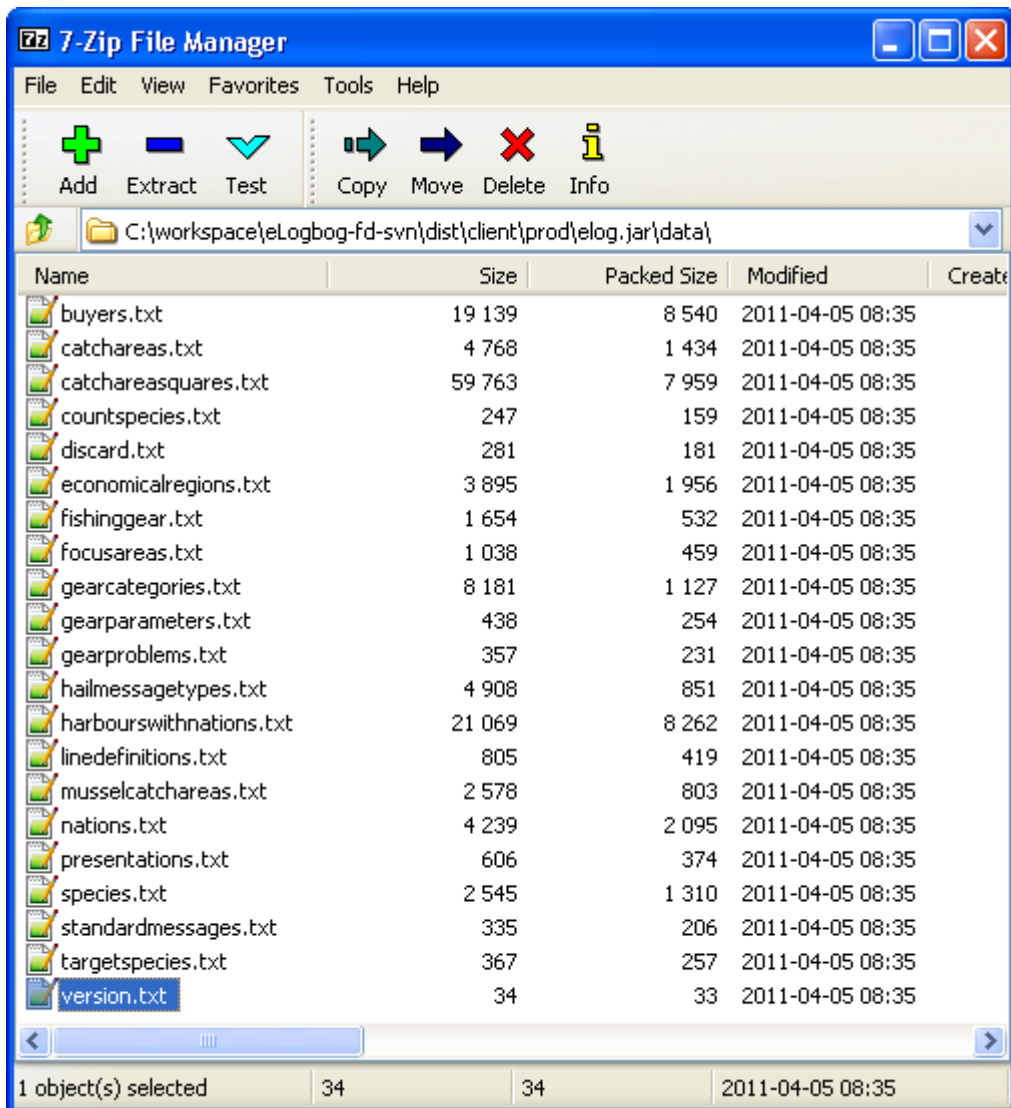
```
curl -X GET http://127.0.0.1:8095/elog/v1/logbook
```

Resultatet er indholdet af filen currentlogbook.xml uden nogen modifikationer.

Hent master data

```
curl -X GET http://127.0.0.1:8095/elog/v1/masterdata/<filename>
```

Som <filename> kan man angive hvilken som helst fra data-mappen i elog.jar-filen:

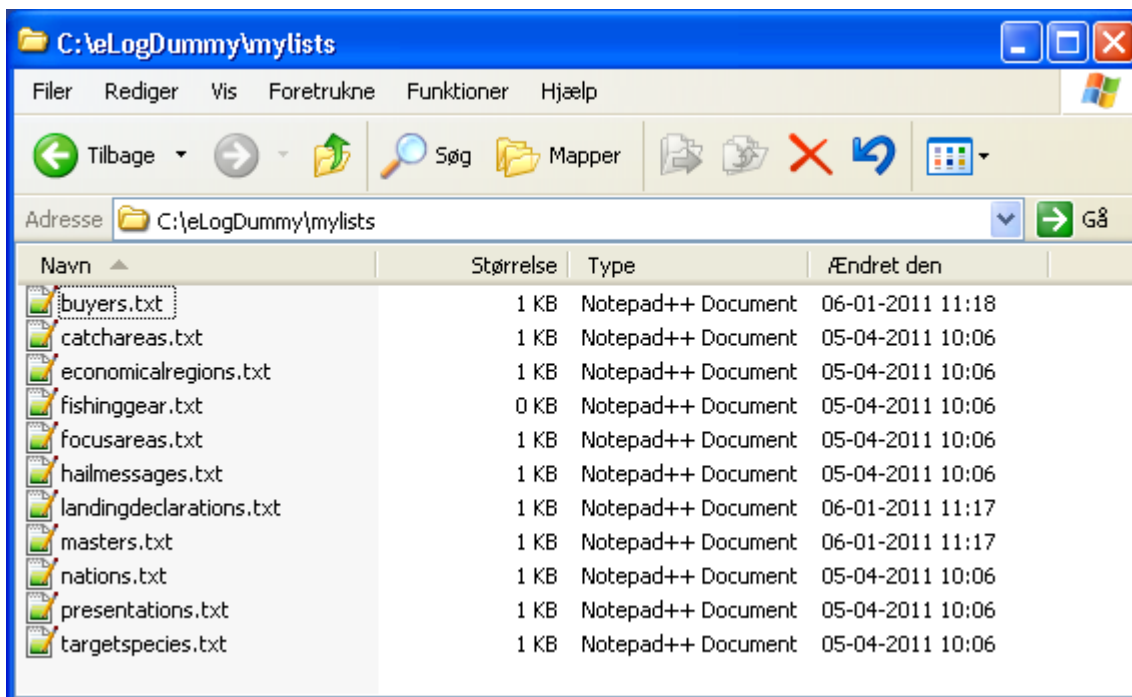


Resultatet vil være filens indhold uden modifikationer. Bemærk, at filerne ikke er i XML-format.

Hent personaliseret liste

```
curl -X GET http://127.0.0.1:8095/eelog/v1/personal_list/<filename>
```

Som <filename> kan man angive hvilken som helst fra mylists-mappen i eLog-installationsmappen. Bemærk, at filerne ikke er i XML-format.



Opret fangst

```
curl -X PUT http://127.0.0.1:8095/elog/v1/catch/<filename> --data-binary
@<data-filename>
```

<filename> skal være et gyldigt filnavn i Windows operativsystemet. Indholdet i <data-filename> vil blive gemt i en fil med navnet "apidata/catch-<filename>.txt". Det vil være ønskeligt, at <filename> indeholder nyttigt metadata, som kan vises til fiskeren, for eksempel: "vdec-2011.05.09-16.26.11". Navnet vil også blive brugt til at sortere efter, hvis der er mere end én besked af en bestemt type i køen.

Eksempel på indhold i <data-filename>:

```
<Dummy-root-element>
  <FishingGear>
    <GearCode>TBB</GearCode>
    <MeshSize>22</MeshSize>
    <Length>33</Length>
    <Count>44</Count>
  </FishingGear>
  <Catch>
    ...
  </Catch>
</Dummy-root-element>
```

FishingGear-elementet skal matche en af de redskaber, som er defineret af fiskeren i filen mylists/geartypes.txt, ellers vil det blive afvist. Bemærk at fiskeren kan ændre i redskabsdefinitionerne på et vilkårligt tidspunkt. Derfor skal klienten af API'et helst verificere gyldigheden af FishingGear-elementet lige inden afsendelse. Alligevel er der ingen garanti for, at fiskeren ikke når at ændre redskabsdefinitionen, inden han får hentet API-besked ind i logbogen.

Catch-elementet kan have to forskellige formater: CatchOperation = BOTH eller GET.

Formatet for BOTH - "Fangstoplysninger - udvidet" ser således ud:

```
<Catch>
  <CatchOperation>BOTH</CatchOperation>
  <GearProblem>3</GearProblem>
  <StartPosition>
    <IcesSquare>42F3</IcesSquare>
    <Longitude>3.325</Longitude>
    <Latitude>56.875</Latitude>
  </StartPosition>
  <EndPosition>
    <IcesSquare>42F3</IcesSquare>
    <Longitude>3.325</Longitude>
    <Latitude>56.875</Latitude>
  </EndPosition>
  <StartDateTime>
    <Date>2011-05-09</Date>
    <Hour>13</Hour>
    <Minute>30</Minute>
  </StartDateTime>
  <EndDateTime>
    <Date>2011-05-09</Date>
    <Hour>16</Hour>
    <Minute>30</Minute>
  </EndDateTime>
  <DetailCollection>
    <CatchArea>
      <IcesSquare>42F3</IcesSquare>
      <CatchAreaName>4B</CatchAreaName>
      <EconomicalRegionCode>EEC</EconomicalRegionCode>
    </CatchArea>
    <Detail>
      <SpeciesCode>SLH</SpeciesCode>
      <Weight>1234</Weight>
    </Detail>
    <Detail>
      <SpeciesCode>TOR</SpeciesCode>
      <Weight>2345</Weight>
    </Detail>
  </DetailCollection>
</Catch>
```

Hvis der ikke har været nogen fangst for slæbet, skal DetailCollection-elementet specificeres således:

```
<DetailCollection>
  <CatchArea>
    <IcesSquare>42F3</IcesSquare>
    <CatchAreaName>4B</CatchAreaName>
    <EconomicalRegionCode>EEC</EconomicalRegionCode>
  </CatchArea>
</DetailCollection>
```

Formatet for GET – "Fangst med passive redskaber" ser således ud:


```

<Catch>
  <CatchOperation>GET</CatchOperation>
  <NumberOfHauls>1</NumberOfHauls>
  <NumberOfPassiveFishingGear>3</NumberOfPassiveFishingGear>
  <Duration>3</Duration>
  <EndPosition>
    <Longitude>3.325</Longitude>
    <Latitude>56.875</Latitude>
  </EndPosition>
  <EndDateTime>
    <Date>2011-05-09</Date>
    <Hour>16</Hour>
    <Minute>0</Minute>
  </EndDateTime>
  <DetailCollection>
    <CatchArea>
      <IcesSquare>42F3</IcesSquare>
      <CatchAreaName>4B</CatchAreaName>
      <EconomicalRegionCode>EEC</EconomicalRegionCode>
    </CatchArea>
    <Detail>
      <SpeciesCode>SLH</SpeciesCode>
      <Weight>2345</Weight>
    </Detail>
    <Detail>
      <SpeciesCode>TOR</SpeciesCode>
      <Weight>1234</Weight>
    </Detail>
  </DetailCollection>
</Catch>

```

De to andre fangstformater CATCHINFORMATION og SET vil sikkert også virke, men de er ikke testet.

Opret afrejse

```
curl -X PUT http://127.0.0.1:8095/elog/v1/departure/<filename> --data-binary
@<data-filename>
```

<filename> skal være et gyldigt filnavn i Windows operativsystemet. Indholdet i <data-filename> vil blive gemt i en fil med navnet "apidata/departure-<filename>.txt". Det vil være ønskeligt, at <filename> indeholder nyttigt metadata, som kan vises til fiskeren, for eksempel: "vdec-2011.05.09-16.26.11". Navnet vil også blive brugt til at sortere efter, hvis der er mere end én besked af en bestemt type i køen.

Eksempel på indhold i <data-filename>:

```

<Departure>
  <Vessel>
    <VisibleIdentification>E532</VisibleIdentification>
    <Name>ROCKALL</Name>
    <FlagState>DNK</FlagState>
    <RadioCallSign>OXNN</RadioCallSign>
  </Vessel>
  <Master>sikn</Master>
  <AnticipatedActivity>REL</AnticipatedActivity>
  <FishingGearsOnBoard>
    <FishingGear>

```

```

        <GearCode>SDN</GearCode>
        <MeshSize>4</MeshSize>
        <Count>3</Count>
    </FishingGear>
    <FishingGear>
        <GearCode>TBB</GearCode>
        <MeshSize>1</MeshSize>
        <Length>2</Length>
        <Count>3</Count>
    </FishingGear>
</FishingGearsOnBoard>
<Partner>
    <VisibleIdentification>MAK1</VisibleIdentification>
    <Name>MAKKER1</Name>
    <FlagState>DNK</FlagState>
    <RadioCallSign>RCS1</RadioCallSign>
</Partner>
<DateTime>
    <Date>2011-05-10</Date>
    <Hour>3</Hour>
    <Minute>0</Minute>
</DateTime>
<DepartureHarbour>
    <HarbourCode>4003</HarbourCode>
    <NationCode>DNK</NationCode>
    <HarbourName>Agersø</HarbourName>
</DepartureHarbour>
<DetailCollection>
    <CatchArea>
        <CatchAreaName>4B</CatchAreaName>
        <EconomicalRegionCode>NOR</EconomicalRegionCode>
    </CatchArea>
    <Detail>
        <SpeciesCode>SLH</SpeciesCode>
        <Weight>1234</Weight>
    </Detail>
</DetailCollection>
<DetailCollection>
    <CatchArea>
        <CatchAreaName>3AN</CatchAreaName>
        <EconomicalRegionCode>EEC</EconomicalRegionCode>
    </CatchArea>
    <Detail>
        <SpeciesCode>TOR</SpeciesCode>
        <Weight>2345</Weight>
    </Detail>
</DetailCollection>
</Departure>

```

Opret ankomst

```
curl -X PUT http://127.0.0.1:8095/elog/v1/arrival/<filename> --data-binary
@<data-filename>
```

<filename> skal være et gyldigt filnavn i Windows operativsystemet. Indholdet i <data-filename> vil blive gemt i en fil med navnet "apidata/arrival-<filename>.txt". Det vil være ønskeligt, at <filename> indeholder

nyttigt metadata, som kan vises til fiskeren, for eksempel: "vdec-2011.05.09-16.26.11". Navnet vil også blive brugt til at sortere efter, hvis der er mere end én besked af en bestemt type i køen.

Eksempel på indhold i <data-filename>:

```
<Arrival>
  <DateTime>
    <Date>2011-05-10</Date>
    <Hour>15</Hour>
    <Minute>0</Minute>
  </DateTime>
  <ArrivalHarbour>
    <HarbourCode>4003</HarbourCode>
    <NationCode>DNK</NationCode>
    <HarbourName>Agersø</HarbourName>
  </ArrivalHarbour>
</Arrival>
```

Hent position for tidspunkt

Bemærk venligst, at positionslogfunktionen ikke er aktiveret som standard i eLog klienten. Dette skyldes, at funktionen betragtes som eksperimentel, indtil den er testet i praksis på et antal udvalgte fiskefartøjer. Positionslogfunktionen aktiveres og konfigureres ved at tilføje linier i filen logbook.properties, som er placeret i roden af eLog installationsmappen, typisk C:\eLog. De linier, som skal tilføjes er:

```
position.log.enabled=<true|false>
position.log.size=<integer>
position.log.interval=<integer>
```

position.log.enabled kan gives værdierne true og false og starter og stopper logningen af GPS positioner. API-funktionen bliver ikke disabled ved at afbryde logningen, men værdien af API'et er selvfølgelig begrænset, hvis der ikke bliver logget løbende.

position.log.size angiver det maksimale antal tidspunkt/position-registreringspar, som på et givet tidspunkt skal holdes i den rullende log. Alle positionslogdataene bliver opsamlet og holdt i PC'ens hukommelse. Det vil sige, man skal være opmærksom på ikke at gøre position.log.size for stor af hensyn til hukommelsesforbruget. Defaultværdien, hvis konfigurationen udelades, er 1440 (= 24*60), hvilket svarer til ét døgn's logning, hvis der samples hvert minut.

position.log.interval angiver sampleintervallet i sekunder og defaultværdien er 60. Minimumintervallet er 5 sekunder.

De nye opsamlede positioner bliver ikke skrevet til harddisken løbende, så hvis eLog klienten bliver lukket unormalt, så bliver dataene ikke gemt. Ved normal nedlukning bliver alle positioner gemt i filen PositionLog.ser i eLog klientens installationsmappe.

```
curl -X GET http://127.0.0.1:8095/elog/v1/position/<date>/<time>
```

<date> er i formatet YYYYMMDD, for eksempel 20110817. <time> er i formatet HHMMSS, for eksempel 090807.

Her et eksempel på et API-kald:

```
curl -vX GET http://127.0.0.1:8095/elog/v1/position/20110819/061200
```

og her er den tilhørende svarbesked:

```
2011-08-19 06:11:57 55 38 77 N 12 8 48 E
```

```
2011-08-19 06:12:02 55 38 77 N 12 8 48 E
```

Der er to linier i svaret: en linie for det nærmeste logtidspunkt som ligger før det forespurgte tidspunkt og en linie for det nærmeste tidspunkt efter. Hvis det forespurgte tidspunkt ligger før eller efter det første eller sidste logtidspunkt, så vil der kun være en linie i svaret. Tiderne er i UTC. Hvis positionsloggen er tom, vil man få et HTTP 500 Server Error svar. Der er ikke noget check på, at tidspunkterne for de returnerede positioner er i rimelig nærhed af det forespurgte tidspunkt.

Positionsformatet er:

```
<lat_deg> <lat_min> <lat_hun> <N|S> <lon_deg> <lon_min> <lon_hun> <E|W>
```

lat_deg: Latitude degrees (0-90)

lat_min: Latitude minutes (0-59)

lat_hun: Latitude hundredths of minutes (0-99)

N or S: Latitude hemisphere

lon_deg: Longitude degrees (0-180)

lon_min: Longitude minutes (0-59)

lon_hun: Longitude hundredths of minutes (0-99)

E or W: Longitude hemisphere