

# Register Over Frivilligt Udelukkede Spillere Services

Report date: 01. Sep. 2011

## **Table of Contents**

<b>1</b>	<b>PROXY SERVICE DESCRIPTIONS</b>	<b>3</b>
1.1	GamblerCreate	6
1.2	GamblerCheck	9
1.3	GamblerCSRPValidation	11

# 1 Proxy service descriptions

The XSD schemas for the request and the response, respectively, of each proxy service are shown in the following sections. The schemas are also made available as separate files. The XSD schemas for the requests and the responses are based on the following common types:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.skat.dk/lur/GamblerCommonTypes"
targetNamespace="http://www.skat.dk/lur/GamblerCommonTypes">
  <complexType name="HeadInformationRequestType">
    <sequence>
      <element name="Kontekst" type="tns:KontekstType"/>
    </sequence>
  </complexType>
  <complexType name="HeadInformationResponseType">
    <sequence>
      <element name="Kontekst" type="tns:KontekstType"/>
    </sequence>
  </complexType>
  <complexType name="KontekstType">
    <sequence>
      <any namespace="##other" processContents="skip"/>
    </sequence>
  </complexType>
  <complexType name="PersonInformationType">
    <sequence>
      <element name="PersonCPRNumber"
type="tns:PersonCPRNumberType"/>
    </sequence>
  </complexType>
  <simpleType name="TilladelsesindehaverType">
    <restriction base="string"/>
  </simpleType>
  <simpleType name="PersonCPRNumberType">
    <restriction base="string">
      <pattern value="\d{10}|(\d{6})-\d{4}"/>
    </restriction>
  </simpleType>
</schema>
```

This file is also available as GamblerCommonTypes.xsd.

A WSDL 1.1 (Web Services Definition Language) file that provides a description of all the proxy services is also made available as a GamblerService.wsdl file:

```
<?xml version="1.0" encoding="UTF-8"?>
<s0:definitions xmlns:s0="http://schemas.xmlsoap.org/wsdl/"
xmlns:s1="http://services.lur.skat.dk"
```

```
xmlns:s2="http://schemas.xmlsoap.org/wsdl/soap/"
name="GamblerService"
targetNamespace="http://services.lur.skat.dk">

<s0:types>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://services.lur.skat.dk"
    xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
    xmlns:rreq="http://www.skat.dk/lur/GamblerCreateRequest"
    xmlns:rres="http://www.skat.dk/lur/GamblerCreateResponse"
    xmlns:creq="http://www.skat.dk/lur/GamblerCheckRequest"
    xmlns:cres="http://www.skat.dk/lur/GamblerCheckResponse"
    xmlns:reqCpr="http://www.skat.dk/lur/GamblerCSRValidationRequest"
    xmlns:resCpr="http://www.skat.dk/lur/GamblerCSRValidationResponse"
    targetNamespace="http://services.lur.skat.dk">

    <xsd:import namespace="http://www.skat.dk/lur/GamblerCreateRequest"
      schemaLocation="GamblerCreateRequest.xsd" />
    <xsd:import namespace="http://www.skat.dk/lur/GamblerCreateResponse"
      schemaLocation="GamblerCreateResponse.xsd" />
    <xsd:import namespace="http://www.skat.dk/lur/GamblerCheckRequest"
      schemaLocation="GamblerCheckRequest.xsd" />
    <xsd:import namespace="http://www.skat.dk/lur/GamblerCheckResponse"
      schemaLocation="GamblerCheckResponse.xsd" />
    <xsd:import namespace="http://www.skat.dk/lur/GamblerCSRValidationRequest"
      schemaLocation="GamblerCSRValidationRequest.xsd" />
    <xsd:import namespace="http://www.skat.dk/lur/GamblerCSRValidationResponse"
      schemaLocation="GamblerCSRValidationResponse.xsd" />

    <xsd:element name="GamblerCreateRequest" type="rreq:GamblerCreateRequest" />
    <xsd:element name="GamblerCreateResponse" type="rres:GamblerCreateResponse" />
    <xsd:element name="GamblerCheckRequest" type="creq:GamblerCheckRequest" />
    <xsd:element name="GamblerCheckResponse" type="cres:GamblerCheckResponse" />
    <xsd:element name="GamblerCSRValidationRequest"
      type="reqCpr:GamblerCSRValidationRequest" />
    <xsd:element name="GamblerCSRValidationResponse"
      type="resCpr:GamblerCSRValidationResponse" />
  </xsd:schema>
</s0:types>

<s0:message name="GamblerCreateRequest">
  <s0:part element="s1:GamblerCreateRequest" name="GamblerCreateRequest" />
</s0:message>
<s0:message name="GamblerCreateResponse">
  <s0:part element="s1:GamblerCreateResponse" name="GamblerCreateResponse" />
</s0:message>
<s0:message name="GamblerCheckRequest">
  <s0:part element="s1:GamblerCheckRequest" name="GamblerCheckRequest" />
</s0:message>
```

```
</s0:message>
<s0:message name="GamblerCheckResponse">
  <s0:part element="s1:GamblerCheckResponse" name="GamblerCheckResponse" />
</s0:message>
<s0:message name="GamblerCSRPValidationRequest">
  <s0:part element="s1:GamblerCSRPValidationRequest"
name="GamblerCSRPValidationRequest" />
</s0:message>
<s0:message name="GamblerCSRPValidationResponse">
  <s0:part element="s1:GamblerCSRPValidationResponse"
name="GamblerCSRPValidationResponse" />
</s0:message>

<s0:portType name="GamblerService">
  <s0:operation name="GamblerCreate">
    <s0:input message="s1:GamblerCreateRequest" />
    <s0:output message="s1:GamblerCreateResponse" />
  </s0:operation>

    <s0:operation name="GamblerCheck">
      <s0:input message="s1:GamblerCheckRequest" />
      <s0:output message="s1:GamblerCheckResponse" />
    </s0:operation>
    <s0:operation name="GamblerCSRPValidation">
      <s0:input message="s1:GamblerCSRPValidationRequest" />
      <s0:output message="s1:GamblerCSRPValidationResponse" />
    </s0:operation>
  </s0:portType>

<s0:binding name="GamblerServiceSOAP" type="s1:GamblerService">
  <s2:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  <s0:operation name="GamblerCreate">
    <s0:input>
      <s2:body use="literal" parts="GamblerCreateRequest" />
    </s0:input>
    <s0:output>
      <s2:body use="literal" parts="GamblerCreateResponse" />
    </s0:output>
  </s0:operation>

  <s0:operation name="GamblerCheck">
    <s0:input>
      <s2:body use="literal" parts="GamblerCheckRequest" />
    </s0:input>
    <s0:output>
      <s2:body use="literal" parts="GamblerCheckResponse" />
    </s0:output>
  </s0:operation>
```

```
<s0:operation name="GamblerCSRPValidation">
  <s0:input>
    <s2:body use="literal" parts="GamblerCSRPValidationRequest" />
  </s0:input>
  <s0:output>
    <s2:body use="literal" parts="GamblerCSRPValidationResponse" />
  </s0:output>
</s0:operation>
</s0:binding>

<s0:service name="GamblerService">
  <s0:port binding="s1:GamblerServiceSOAP" name="GamblerServiceSOAP">
    <s2:address location="http://localhost:7001/LUR/GamblerService" />
  </s0:port>
</s0:service>

</s0:definitions>
```

## 1.1 GamblerCreate

1. Basic information					
<b>GamblerCreate</b>					
<b>System/package:</b>	<b>Version:</b>	<b>Created by:</b>	<b>Date:</b>	<b>Updated by:</b>	<b>Date:</b>
LUR	0.8	jn	20100922	jn	20100110
2. Purpose					
To ensure that a person is registered as a gambler either for a specified limited amount of time or indefinitely from a specified date and time.					
3. Functionality (overview)					
This proxy service allows a Tilladelsesindehaver actor to register a specified person as a gambler either for a specified limited amount of time or indefinitely from a specified time and date. This service does not support the 24 hour cooling-off period.					
If the service is applied for a person that is already registered as a gambler at the time of the request, the information in the Ludomaniregister is left unaltered.					
4. Functionality (Detailed)					
The content of the request is validated against a schema. If this validation succeeds, the following validations are performed.					

The service initially checks whether the actor has the security privileges to report a gambler. If not, the service responds with a technical error. If so, the functionality corresponding to the GamblerCheck service is used to examine if the person is already registered as a gambler. If the person is already registered as a gambler, the service responds with that information. If not, the remaining business validations described in Section 6 below are performed.

Unless any of the validations fail, the person is reported as a gambler

## 5. Data structure

### 5.1 Request

The XSD schema below is also made available as a file: GamblerCreateRequest.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.skat.dk/lur/GamblerCreateRequest"
  xmlns:com="http://www.skat.dk/lur/GamblerCommonTypes"
  targetNamespace="http://www.skat.dk/lur/GamblerCreateRequest">
  <import namespace="http://www.skat.dk/lur/GamblerCommonTypes"
    schemaLocation="GamblerCommonTypes.xsd"/>
  <complexType name="GamblerCreateRequest">
    <complexContent>
      <extension base="com:HeadInformationRequestType">
        <sequence>
          <element name="Actor"
            type="com:TilladelsesindehaverType"/>
          <element name="PersonInformation"
            type="com:PersonInformationType"/>
          <element name="ExclusionStart"
            type="tns:ExclusionStartType"/>
          <element name="ExclusionEnd"
            type="tns:ExclusionEndType" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="ExclusionStartType">
    <sequence>
      <element name="GameExclusionStartDate" type="tns:LurDateType"/>
      <element name="GameExclusionStartTime" type="tns:LurTimeType"/>
    </sequence>
  </complexType>
  <complexType name="ExclusionEndType">
    <sequence>
      <element name="GameExclusionEndDate" type="tns:LurDateType"/>
      <element name="GameExclusionEndTime" type="tns:LurTimeType"/>
    </sequence>
  </complexType>
  <simpleType name="LurDateType">
    <restriction base="string">
```

```
        <pattern value="\d{4}\-\d{2}\-\d{2}"/>
    </restriction>
</simpleType>
<simpleType name="LurTimeType">
    <restriction base="string">
        <pattern value="\d{2}:\d{2}:\d{2}"/>
    </restriction>
</simpleType>
</schema>
```

## 5.2 Response

The XSD schema below is also made available as a file: GamblerCreateResponse.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.skat.dk/lur/GamblerCreateResponse"
  xmlns:com="http://www.skat.dk/lur/GamblerCommonTypes"
  targetNamespace="http://www.skat.dk/lur/GamblerCreateResponse">
    <import namespace="http://www.skat.dk/lur/GamblerCommonTypes"
      schemaLocation="GamblerCommonTypes.xsd"/>
    <complexType name="GamblerCreateResponse">
      <complexContent>
        <extension base="com:HeadInformationResponseType">
          <sequence/>
        </extension>
      </complexContent>
    </complexType>
</schema>
```

## 5.3 Error structure

See Section 5.2.

## 6. Business validations

**6.1 Security check:** If the actor does not have the security privileges to report a gambler, a technical error message is reported (GUN = LUR: SpillerOpret.errInvalidSecurityPrivileges), and the validation process ends.

**6.3 Validate start date and time of exclusion period:** It is validated that specified date and time is no sooner than the current date and time. If this validation fails, an error message is added to the response (GUN = LUR: SpillerOpret.errInvalidStartDateTimeHasPassed), and the validation process continues with the first step in 6.4. If the validation is successful, it is validated that the start date and time is no later than the end of the current month. Should that validation fail, an error message is added to the response (GUN = LUR: SpillerOpret.errInvalidStartDateTimeIsTooLate).

**6.4 Validate end date and time of the exclusion:** If no information has been passed on in the UdelukkelsesperiodeSlutDatoTidspunkt field, it is established that the exclusion should be indefinite, and this step of the validation process ends. It is validated that the end date and time is at least a month later than the start date and time. Should that validation fail, an error message is added to the response (GUN = LUR: SpillerOpret.errInvalidExclusionPeriodShouldAtLeastBeAMonth).



The business validation is complete. If the response does not contain any validation messages, the process continues in step 6.5. Otherwise, a response is generated with the appropriate validation messages.

**6.5 Persist reported gambler:** Should persisting the validated information in the LUR database fail, a response is created (any validation error messages are discarded from the response) with a technical error message (GUN = LUR: SpillerOpret.errInvalidPersistingReportedGamblerFailed). If persistence is successful, a receipt is generated (GUN = LUR: SpillerOpret.GamblerIsReported).

## 7. Notes

For this service the GUNs mentioned in Section 6 should be mapped to text using the Danish language identifier:

- LUR:SpillerOpret.errInvalidSecurityPrivileges = "The user does not have sufficient security privileges to report a gambler."
- LUR: SpillerOpret.errInvalidCPRNumber = "The CPR number is not valid."
- LUR: SpillerOpret.errInvalidStartDateTimeHasPassed = "The specified start date and time should be no sooner than the current date and time."
- LUR: SpillerOpret.errInvalidStartDateTimeIsTooLate = "The specified end date and time should be no later than the end of the current month."
- LUR: SpillerOpret.errInvalidExclusionPeriodShouldAtLeastBeAMonth = "A temporary exclusion of a gambler should last at least one month."
- LUR: SpillerOpret.errInvalidPersistingReportedGamblerFailed = "A technical error occurred in the LUR database. The reported gambler is not registered in the Ludomaniregister."
- LUR: SpillerOpret.GamblerIsReported = "The person is registered as a gambler in the Ludomaniregister."

## 1.2 GamblerCheck

### 1. Basic information

#### GamblerCheck

System/package:	Version:	Created by:	Date:	Updated by:	Date:
LUR	0.8	jn	20100922	jn	20100110

### 2. Purpose

To check whether a person is registered as a gambler in the Ludomaniregister either temporarily or indefinitely, or not at all.

### 3. Functionality (overview)

This service allows a Tilladelsesindehaver actor to check whether a person, identified by the person's CPR number, is registered as a gambler in the Ludomaniregister either temporarily or indefinitely, or not at all. A Tilladelsesindehaver actor gains access to this business service through the GamblerCheck proxy service. Neither of these services support the 24 hour cooling-off period.

#### 4. Functionality (Detailed)

The content of the request is validated against a schema, See Section 5 below.

#### 5. Data structure

##### 5.1 Request

The XSD schema below is also made available as a file: GamblerCheckRequest.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.skat.dk/lur/GamblerCheckRequest"
xmlns:com="http://www.skat.dk/lur/GamblerCommonTypes"
targetNamespace="http://www.skat.dk/lur/GamblerCheckRequest">
  <import namespace="http://www.skat.dk/lur/GamblerCommonTypes"
schemaLocation="GamblerCommonTypes.xsd"/>
  <complexType name="GamblerCheckRequest">
    <complexContent>
      <extension base="com:HeadInformationRequestType">
        <sequence>
          <element name="PersonInformation"
type="com:PersonInformationType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

##### 5.2 Response

The XSD schema below is also made available as a file: GamblerCheckResponse.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.skat.dk/lur/GamblerCheckResponse"
xmlns:com="http://www.skat.dk/lur/GamblerCommonTypes"
targetNamespace="http://www.skat.dk/lur/GamblerCheckResponse">
  <import namespace="http://www.skat.dk/lur/GamblerCommonTypes"
schemaLocation="GamblerCommonTypes.xsd"/>
  <complexType name="GamblerCheckResponse">
    <complexContent>
      <extension base="com:HeadInformationResponseType">
        <sequence>
          <element name="ExclusionStatus"
type="tns:ExclusionStatusType" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="ExclusionStatusType">
    <sequence>
```

```
<element name="GamblerExclusionStatus"  
type="tns:GamblerExclusionStatusType"/>  
  </sequence>  
</complexType>  
<simpleType name="GamblerExclusionStatusType">  
  <restriction base="string">  
    <enumeration value="PersonIsNotRegistered"/>  
    <enumeration value="PersonIsRegisteredIndefinitely"/>  
    <enumeration value="PersonIsRegisteredTemporarily"/>  
  </restriction>  
</simpleType>  
</schema>
```

### 5.3 Error structure

See Section 5.2.

## 6. Business validations

**6.1 Security check:** If the actor does not have the security privileges to check the status of a gambler in the Ludomaniregister, a technical error message is reported (GUN = LUR: SpillerSøg.errInvalidSecurityPrivileges), and the validation process ends.

Assuming that these validations are successful, a request is made against the LUR database to examine whether the person is registered as a gambler at the date and time of the request. If the person is not registered, a message is added to the response (GUN = LUR: SpillerSøg.PersonIsNotRegistered). If the person is registered as a gambler for a limited period of time and the time of the request lies within that period of time, a message is added to the response (GUN = LUR: SpillerSøg.PersonRegisteredTemporarily). If the person is registered as a gambler for a limited period of time, a message is added to the response (LUR: SpillerSøg.PersonRegisteredIndefinitely).

## 7. Notes

For this service the GUNs mentioned in Section 6 should be mapped to text using the English language identifier:

- LUR: SpillerSøg.errInvalidSecurityPrivileges = "The user does not have sufficient security privileges to report a gambler."
- LUR: SpillerSøg.errInvalidCPRNumber = "The CPR number is not valid."
- LUR: SpillerSøg.PersonIsNotRegistered = "The person is not registered in the Ludomaniregister."
- LUR: SpillerSøg.PersonRegisteredTemporarily = "The gambler should be excluded from gambling temporarily."
- LUR: SpillerSøg.PersonRegisteredIndefinitely = "The gambler should be excluded from gambling indefinitely."

## 1.3 GamblerCSRValidation

### 1. Basic information

## GamblerCSRValidation

System/package:	Version:	Created by:	Date:	Updated by:	Date:
LUR	0.8	jn	20101018		

### 2. Purpose

To check whether a person's CPR is registered in the CSR registry. Furthermore the service validates that the CPR belongs to a person 18 years of age or older.

### 3. Functionality (overview)

This service allows a Tilladelsesindehaver actor to check whether a person, identified by the person's CPR number, is registered in the CSR registry (Central registry of all Danish CPRs).

### 4. Functionality (Detailed)

The content of the request is validated against a schema, See Section 5 below.

The DAPMedStiHent service is used to retrieve error messages in either Danish (DK) or English (US).

The service calls the PersonStamoplysningerMultiHent service to retrieve content based on the CPR.

If the service returns a valid person

- CPRStatus is set to CPRIsRegistered
- AgeStatus is set to either Agels18OrAbove OR Agels17OrBelow

Else

- CPRStatus is set to CPRIsNotRegistered

In case of technical errors or validation errors, the user should receive a message.

### 5. Data structure

#### 5.1 Request

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.skat.dk/lur/GamblerCSRValidationRequest"
xmlns:com="http://www.skat.dk/lur/GamblerCommonTypes"
targetNamespace="http://www.skat.dk/lur/GamblerCSRValidationRequest">
  <import namespace="http://www.skat.dk/lur/GamblerCommonTypes"
schemaLocation="GamblerCommonTypes.xsd"/>
  <complexType name="GamblerCSRValidationRequest">
    <complexContent>
      <extension base="com:HeadInformationRequestType">
        <sequence>
          <element name="PersonInformation"
type="com:PersonInformationType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

```
</complexContent>  
</complexType>  
</schema>
```

## 5.2 Response

```
<?xml version="1.0" encoding="UTF-8"?>  
<schema xmlns="http://www.w3.org/2001/XMLSchema"  
  xmlns:tns="http://www.skat.dk/lur/GamblerCSRValidationResponse"  
  xmlns:com="http://www.skat.dk/lur/GamblerCommonTypes"  
  targetNamespace="http://www.skat.dk/lur/GamblerCSRValidationResponse">  
  <import namespace="http://www.skat.dk/lur/GamblerCommonTypes"  
    schemaLocation="GamblerCommonTypes.xsd"/>  
  <complexType name="GamblerCSRValidationResponse">  
    <complexContent>  
      <extension base="com:HeadInformationResponseType">  
        <sequence>  
          <element name="GamblerStatus"  
type="tns:GamblerStatusType" minOccurs="0"/>  
        </sequence>  
      </extension>  
    </complexContent>  
  </complexType>  
  <complexType name="GamblerStatusType">  
    <sequence>  
      <element name="GamblerCPRStatus" type="tns:CPRStatusType"/>  
      <element name="GamblerAgeStatus" type="tns:AgeStatusType"  
minOccurs="0"/>  
    </sequence>  
  </complexType>  
  <simpleType name="CPRStatusType">  
    <restriction base="string">  
      <enumeration value="CPRIsNotRegistered"/>  
      <enumeration value="CPRIsRegistered"/>  
    </restriction>  
  </simpleType>  
  <simpleType name="AgeStatusType">  
    <restriction base="string">  
      <enumeration value="AgeIs18OrAbove"/>  
      <enumeration value="AgeIs17OrBelow"/>  
    </restriction>  
  </simpleType>  
</schema>
```

## 5.3 Error structure

See Section 5.2.

## 6. Business validations

6.1 Security check: If the actor does not have the security privileges to perform the CPR validation in the

Ludomaniregister, a technical error message is reported (GUN = LUR:SpillerCSRPSValidering.errInvalidSecurityPrivileges), and the validation process ends.

Assuming that these validations are successful, a request is made to the CSRP to validate that the CPR exist. If the CPR is not registered, a message is added to the response (GUN = LUR:SpillerCSRPSValidering.CPRIsNotRegistered). If the CPR is registered a message is added to the response (GUN = LUR: SpillerCSRPSValidering. CPRIsRegistered) and the service proceeds to validate the age. If the person registered with the CPR is 18 years of age or older, a message is added to the response (LUR: SpillerCSRPSValidering.PersonIs18yearsOrOlder). Else the message (LUR: SpillerCSRPSValidering.PersonIsNot18yearsOrOlder) is added to the response.

## 7. Notes

For this service the GUNs mentioned in Section 6 should be mapped to text using the English language identifier:

- LUR: GamblerCSRPSValidation.errInvalidSecurityPrivileges = "The user does not have sufficient security privileges to validate a CPR."
- LUR: GamblerCSRPSValidation. CPRIsNotRegistered = "The CPR does not exist in the CSRP registry."
- LUR: GamblerCSRPSValidation. CPRIsRegistered = "The CPR exist in the CSRP registry."
- LUR: GamblerCSRPSValidation. PersonIs18yearsOrOlder = "The CPR belongs to a person age 18 or older."
- LUR: GamblerCSRPSValidation. PersonIsNot18yearsOrOlder = "The CPR belongs to a person age 17 or younger."