Автоматизированная Информационная Система

«Оцифровка»

Техническая инструкция сервиса по информации из оцифрованных ОСК, посредством СМЭВ Түндүк

г. Бишкек 2019

Сервис работает посредством СМЭВ Түндүк, путем реализации веб-сервиса согласно протоколу СМЭВ.

Для получения WSDL-файла необходимо обратиться по ссылке адреса в сервер безопасности СМЭВ Түндүк.

Терминология:

СМЭВ - система межведомственного электронного взаимодействия.

Сервер безопасности - Шлюз подключения к СМЭВ.

Веб-сервис разработан методом протокола SOAP и состоит из 6 методов для двух видов услуг:

Заголовки сообщений

В этом разделе описываются заголовки SOAP, которые используются СМЭВ Түндук.

В таблице ниже представлены заголовки СМЭВ Түндүк.

Когда клиент службы отправляет запрос на сервер безопасности, обязательно должна присутствовать одна из служб полей или centralService. Если используется поле centralService, сервер безопасности разрешает центральную службу и автоматически заполняется в поле службы идентификатором конкретной услуги, которая реализует центральную службу.

Поле	Тип	Обязательное или Опциональное	Описание
client	XRoadClientIdentifierType	Обязательное	Идентифицирует Клиента — сущность, которая инициирует вызов услуги.
service	XRoadServiceIdentifierType	Опциональное	Идентифицирует сервис, вызываемый запросом.
centralService	XRoadCentralServiceIdentifierType	Опциональное	Идентифицирует центральную службу, вызываемую запросом.
id	string	Обязательное	Уникальный идентификатор для сообщения. Рекомендуемый метод генерации сообщения ID - 30 байт

			случайных данных, в кодировке Base64.
userId	string	Обязательное	Пользователь, действие которого инициировало запрос. Идентификатор пользователя должен быть с двухбуквенным ISO код страны (например, KG20114199609123).
requestHash	string	Опциональное	Для ответов. Это поле содержит хэш запроса SOAP сообщения. Это поле автоматически заполняется поставщиком сервисов сервера безопасности.
requestHash/ @algorithmId	string	Обязательное	Идентифицирует хэшалгоритм, который был использован для вычисления значения запроса поля Hash. Алгоритмы обозначаются как URI перечисленные в XML-DSIG спецификации [DSI13].
issue	string	Опциональное	Идентифицирует заявление, основание или документ, который послужил основанием обращения к сервису. Это поле используется для подключения службы запросов (и ответов) к рабочим процедурам

authentication Method	string	Обязательное	Метод аутентификации
profileVersion	string	Обязательное	Версия протокола СМЭВ Түндүк. Для СМЭВ Түндүк 6 версия протокола 4.0

При ответе служба копирует все поля заголовка из запроса в ответ.

Поле requestHash используется для создания прочной связи между запросом и ответом.

Таким образом, можно доказать, что, например, определенная запись реестра возвращается в ответ на определенный запрос. Используя журналы, можно восстановить и проверить пару запрос-ответ.

Поле requestHash автоматически создается сервером безопасности поставщика услуг и проверяется сервером безопасности клиента службы.

Поле authenticationMethod должно быть одним из следующих:

EID - с удостоверением личности (ID-CARD, DIGI-ID);
MID - с мобильным удостоверением личности (MOBILE-ID);
CERT - с другим сертификатом;
EXTERNAL - через стороннюю услугу (BANK-LINK);
PASSWORD - с идентификатором пользователя и паролем;
SYSTEM - когда запрос был отправлен заданием cron или другим автоматическим процессом.

Данные заголовки являются стандартными для использования СМЭВ Түндук и присуствуют во всех методах.

Описание метода сервиса и его индивидуальные параметры:

GetInformationFromScannedOSK(GetInformationFromScannedOSKRequest request)

Метод возвращает по параметрам входа имеются или нет сведения в оцифрованных ОСК.

Где:

request – параметры запроса с заголовком СМЭВ Түндук и дополнительные параметры:

identifier – полученный уникальный персональный идентификатор;

family, name, patronymic, dateOfBirth – ФИО и дата рождения запроса. Обязательными являются фамилия и имя, а также в параметры family, name, patronymic, допускается и поиск по части параметра, поиск осуществляется по принципу «Содержится в».

Запрос:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xro="http://x-road.eu/xsd/xroad.xsd" xmlns:iden="http://x-
road.eu/xsd/identifiers" xmlns:scan="http://scan_service.x-road.ee/scan">
   <soapenv:Header>
      <xro:protocolVersion>4.0</xro:protocolVersion>
      <xro:issue>123</xro:issue>
      <xro:id>123</xro:id>
      <xro:userId>123</xro:userId>
      <xro:service iden:objectType="SERVICE">
         <iden:xRoadInstance>central-server</iden:xRoadInstance>
         <iden:memberClass>GOV</iden:memberClass>
         <iden:memberCode>70000006</iden:memberCode>
         <!--Optional:-->
         <iden:subsystemCode>osk-subsystem</iden:subsystemCode>
         <iden:serviceCode>GetInformationFromScannedOSK</iden:serviceCode>
         <!--Optional:-->
         <iden:serviceVersion>v1</iden:serviceVersion>
      </xro:service>
      <xro:client iden:objectType="SUBSYSTEM">
         <iden:xRoadInstance>central-server</iden:xRoadInstance>
         <iden:memberClass>GOV</iden:memberClass>
         <iden:memberCode>70000100</iden:memberCode>
         <!--Optional:-->
         <iden:subsystemCode>system</iden:subsystemCode>
      </xro:client>
   </soapenv:Header>
   <soapenv:Body>
      <scan:GetInformationFromScannedOSK>
         <scan:identifier>456hgf-30da-40eb-9e51-345gfgdd</scan:identifier>
         <scan:family>АВД</scan:family>
         <scan:name>ТАЛР</scan:name>
         <scan:patronymic></scan:patronymic>
         <scan:dateOfBirth></scan:dateOfBirth>
      </scan:GetInformationFromScannedOSK>
```

```
   </soapenv:Body>
</soapenv:Envelope>
```

Метод сервиса возвращает:

```
        <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Header>
      <h:client a:objectType="SUBSYSTEM" xmlns:h="http://x-
road.eu/xsd/xroad.xsd" xmlns:a="http://x-road.eu/xsd/identifiers"
xmlns="http://x-road.eu/xsd/xroad.xsd"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <a:xRoadInstance>central-server</a:xRoadInstance>
         <a:memberClass>GOV</a:memberClass>
         <a:memberCode>70000100</a:memberCode>
         <a:subsystemCode>system</a:subsystemCode>
      </h:client>
      <h:id xmlns:h="http://x-road.eu/xsd/xroad.xsd" xmlns="http://x-
road.eu/xsd/xroad.xsd">123</h:id>
      <h:requestHash algorithmId="http://www.w3.org/2001/04/xmlenc#sha512"
xmlns:h="http://x-road.eu/xsd/xroad.xsd" xmlns="http://x-
road.eu/xsd/xroad.xsd">1i5EUGxirmn+baOYigr3upGTYQclnfXCJ5507he6a6mT+9HxvIfMlKMpS
TTCVESgoZo7fm0b/aAfRR6sAzdrTg==</h:requestHash>
      <h:issue xmlns:h="http://x-road.eu/xsd/xroad.xsd" xmlns="http://x-
road.eu/xsd/xroad.xsd">123</h:issue>
      <h:protocolVersion xmlns:h="http://x-road.eu/xsd/xroad.xsd"
xmlns="http://x-road.eu/xsd/xroad.xsd">4.0</h:protocolVersion>
      <h:service a:objectType="SERVICE" xmlns:h="http://x-road.eu/xsd/xroad.xsd"
xmlns:a="http://x-road.eu/xsd/identifiers" xmlns="http://x-
road.eu/xsd/xroad.xsd">
         <a:xRoadInstance>central-server</a:xRoadInstance>
         <a:memberClass>GOV</a:memberClass>
         <a:memberCode>70000006</a:memberCode>
         <a:subsystemCode>osk-subsystem</a:subsystemCode>
         <a:serviceCode>GetInformationFromScannedOSK</a:serviceCode>
         <a:serviceVersion>v1</a:serviceVersion>
      </h:service>
      <h:userId xmlns:h="http://x-road.eu/xsd/xroad.xsd" xmlns="http://x-
road.eu/xsd/xroad.xsd">123</h:userId>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <GetInformationFromScannedOSKResponse xmlns="http://scan_service.x-
road.ee/scan">
         <Status>
            <Code>12</Code>
            <Name>ByThisParametersDataNotFound</Name>
         </Status>
      </GetInformationFromScannedOSKResponse>
   </s:Body>
</s:Envelope>
```

Ответ состоит из параметров запроса с заголовком СМЭВ Түндук и из корпуса сервиса, где:

Status – статус обработки данных:

a)

Code = 0;
Name = "Success";

Успешно. В этом случае значит, что по параметрам запроса имеются сведения и возвращается массив из объектов Result, где:

CardPersonFamily – фамилия из картотеки;
CardPersonName – имя из картотеки;
CardPersonPatronymic – отчество из картотеки;
CardPersonDateOfBirth – дата рождения из картотеки;
CardCriminalCaseNumber – номер уголовного дела из картотеки;

b)

Code = 1;
Name = " ServiceError";
Нет подключения или внутренняя ошибка!
Result = null

c)

Code = 2;
Name = "InvalidCompanyIdentifier";

Неверный идентификатор!
Result = null

d)

Code = 7;
Name = " ParameterFamilyIsEmpty";

Параметр «Фамилия» пуст!
Result = null

e)

Code = 8;
Name = " ParameterNameIsEmpty";

Параметр «Имя» пуст!
Result = null

f)

Code = 9;
Name = " ParameterFamilyMustContains3Characters";

Параметр «Фамилия» должен иметь минимум 3 буквы!
Result = null

g)
Code = 10;
Name = " ParameterNameMustContains3Characters ";

Параметр «Имя» должен иметь минимум 3 буквы!
Result = null

h)
Code = 11;
Name = " DateOfBirthMustBeInFormat:dd.mm.yyyy";

Параметр «Дата рождения» должен иметь формат dd.mm.yyyy !
Result = null

i)
Code = 12;
Name = " ByThisParametersDataNotFound";

По указанным параметрам сведения не имеются!
Result = null