

Автоматизированная Информационная Система

**«Е-Заявление»**

**Техническая инструкция сервиса  
«Апостиль», посредством СМЭВ Түндүк**

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.

### Условные обозначения:

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

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<br>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.<br>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/<br>@algorithmId | string | Обязательное | Идентифицирует хэш алгоритм, 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<br>Method | string | Обязательное | Метод аутентификации                                                       |
| profileVersion           | string | Обязательное | Версия протокола СМЭВ<br>Түндүк. Для СМЭВ Түндүк<br>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 обозначения являются стандартными для использования СМЭВ Түндүк и присутствуют во всех методах.*

**Сервисы:**

*ApostilVerifyDataResponse* ***ApostilVerifyData***(*ApostilVerifyDataRequest request*)  
– проверка данных для апостилирования справок.

А также:

*ApostilVerifyDataTestResponse* ***ApostilVerifyDataTest***  
(*ApostilVerifyDataTestRequest request*) – тестовая среда – проверка данных для апостилирования справок.

Где:

request – параметры запроса с заголовком СМЭВ Түндүк и дополнительные параметры;

companyIdentifier - полученный уникальный персональный идентификатор;

documentCode - 16 значный код – идентификатор заявления/справки, заявитель набирает код со Справки, если сервис имплементируется как отдельная функция, или берется с сервиса для справок о несудимости из сервиса ***ERecordV2StatementConfirmPayment***, ***PayDocumentCreateStatementMode***, ***PayDocumentExistStatementMode***, поле *DocumentCode*, или из сервиса ***BatkenStatementData***, поле *DocumentCode*, или из сервиса ***PayDocument*** системы **emvd-service** для гос.порталов, поле *DocumentCode* - при необходимости для продолжение как одно целое, можно добавить функцию одновременной подачи заявления на получения апостилированной справки, только с согласованием поставщика услуг (МВД КР).

**Примечание:**

для тестовой среды (***ApostilVerifyDataTest***), *documentCode* пользуетесь значением тестовых сервисов (***ERecordV2StatementConfirmPaymentTest***, ***PayDocumentCreateStatementModeTest***, ***PayDocumentExistStatementModeTest***, ***BatkenStatementDataTest***, ***PayDocumentTest***) или пользуетесь значением **9075092410934101**, это значение в тестовой среде не возвращает Код ошибки **7** (**RequestHasApostil** - Заявление/справка по указанному коду уже имеет заказанную и оплоченную апостиль!) которую можно использовать многократно.

apostilTypeCode - тип услуги, код из реестра -

| Тип                               | Код |
|-----------------------------------|-----|
| Апостиль в течение 2 рабочих дней | 1   |
| Апостиль в течение 1 рабочего дня | 2   |
| Апостиль в течение 3-х часов      | 3   |

Метод сервиса возвращает:

а)

объект Status -

Code = 0;

Name = "Success";

**“Успешно”**: в этом случае возвращается объект ObjectResult, где:

DocumentCode - 16 значный код – идентификатор заявления/справки.

PersonName, PersonSurname, PersonPatronymic – имя, фамилия, отчество, заявителя.

ApostilTypeName, ApostilTypeCode - тип услуги, наименование и код.

ApostilTypeSumOfService - сумма услуги.

PaymentIdentifier - идентификатор подтверждения оплаты, используется во втором методе *ApostilConfirmPayment* (*ApostilConfirmPaymentTest*) для подтверждения оплаты.

b) объект Status -

Code = 1;

Name = "ServiceError";

Нет подключения или внутренняя ошибка!

ObjectResult = null

c) объект Status –

Code = 2;

Name = "InvalidCompanyIdentifier";

Неверный идентификатор!

ObjectResult = null

d) объект Status –

Code = 3;

Name = "DocumentCodeMustContain16Characters";

DocumentCode должен содержать 16 цифр!

ObjectResult = null

e) объект Status –

Code = 4;

Name = "InvalidApostilTypeCode";

Неверный тип услуги!  
ObjectResult = null

- f) объект Status –  
Code = 5;  
Name = "ApostilTypeNotFound";

Тип услуги не найден!  
ObjectResult = null

- g) объект Status –  
Code = 6;  
Name = "InvalidRequestByDocumentCode";  
Заявление/справка по указанному 16 значному коду не существует!  
ObjectResult = null

- h) объект Status –  
Code = 7;  
Name = "RequestHasApostil";

Заявление/справка по указанному 16 значному коду уже имеет заказанную и оплаченную апостиль!  
ObjectResult = null

- i)  
Code = x;  
Name = "Другой статус";  
Другие возникшие статусы по время использования сервиса. В этом случае:  
ObjectResult = null

**Пример запроса:**

```
<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:apos="http://apostil_service.x-road.ee/apostil">
  <soapenv:Header>
    <xro:protocolVersion>4.0</xro:protocolVersion>
    <xro:issue></xro:issue>
    <xro:id></xro:id>
    <xro:userId></xro:userId>
    <xro:service iden:objectType="SERVICE">
      <iden:xRoadInstance>central-server</iden:xRoadInstance>
```

```

    <iden:memberClass>GOV</iden:memberClass>
    <iden:memberCode>70000006</iden:memberCode>
    <iden:subsystemCode>record-service</iden:subsystemCode>
    <iden:serviceCode>ApostilVerifyDataTest</iden:serviceCode>
    <iden:serviceVersion>v1</iden:serviceVersion>
  </xro:service>
  <xro:client iden:objectType="SUBSYSTEM">
    <iden:xRoadInstance>central-server</iden:xRoadInstance>
    <iden:memberClass>YouMemberClass</iden:memberClass>
    <iden:memberCode>YouMemberCode</iden:memberCode>
    <iden:subsystemCode>YouSubsystemCode</iden:subsystemCode>
  </xro:client>
</soapenv:Header>
<soapenv:Body>
  <apos:ApostilVerifyDataTest>
    <apos:companyIdentifier>00000000-0000-0000-0000-
0000000000</apos:companyIdentifier>
    <apos:documentCode>9075092410934101</apos:documentCode>
    <apos:apostilTypeCode>1</apos:apostilTypeCode>
  </apos:ApostilVerifyDataTest>
</soapenv:Body>
</soapenv:Envelope>

```

*ApostilConfirmPaymentResponse* **ApostilConfirmPayment**  
*(ApostilConfirmPaymentRequest request)* – подтверждение оплаты для  
 апостилирования справок.

А также:

*ApostilConfirmPaymentTestResponse* **ApostilConfirmPaymentTest**  
*(ApostilConfirmPaymentTestRequest request)* – тестовая среда – подтверждение  
 оплаты для апостилирования справок.

Где:

request – параметры запроса с заголовком СМЭВ Түндүк и дополнительные  
 параметры;

companyIdentifier - полученный уникальный персональный идентификатор;

documentCode - 16 значный код – идентификатор заявления/справки,

**Важно:**

должно использоваться то же самое идентичное значение параметра с методом *ApostilVerifyData*.

paymentIdentifier - идентификатор подтверждения оплаты (PaymentIdentifier) из объекта результата метода *ApostilVerifyData*.

Метод сервиса возвращает:

а)

объект Status -

Code = 0;

Name = "Success";

**“Успешно”**: в этом случае возвращается объект *ObjectResult*, где:

*ApostilIdentifier* - идентификатор апостиля;

*DocumentCode* - 16 значный код – идентификатор заявления/справки.

*PersonName*, *PersonSurname*, *PersonPatronymic* – имя, фамилия, отчество, заявителя.

*ApostilTypeName*, *ApostilTypeCode* - тип услуги, наименование и код.

*Receiver* – Получатель (относится к банковским реквизитам);

*TransitAccount* - Транзитный счет (относится к банковским реквизитам);

*AccumulationAccount* - Лицевой счет (относится к банковским реквизитам);

*BICBank* - БИК банка (относится к банковским реквизитам);

*ReceiverINN* – ПИН получателя (относится к банковским реквизитам);

*GNI* – ГНИ (относится к банковским реквизитам);

ОКРО – ОКПО (относится к банковским реквизитам);

*Bank* – Банк (относится к банковским реквизитам);

б) объект Status -

Code = 1;

Name = "ServiceError";

Нет подключения или внутренняя ошибка!

*ObjectResult* = null

с) объект Status –

Code = 2;

Name = "InvalidCompanyIdentifier";

Неверный идентификатор!

*ObjectResult* = null

d) объект Status –

Code = 3;

Name = "DocumentCodeMustContain16Characters";

DocumentCode должен содержать 16 цифр!

ObjectResult = null

e) объект Status –

Code = 4;

Name = "PaymentIdentifierIsEmpty";

Неверный идентификатор подтверждения оплаты!

ObjectResult = null

f) объект Status –

Code = 5;

Name = "InvalidPaymentIdentifier";

Неверный идентификатор подтверждения оплаты!

ObjectResult = null

g) объект Status –

Code = 6;

Name = "InvalidRequestByDocumentCode ";

Заявление/справка по указанному 16 значному коду не существует!

ObjectResult = null

h) объект Status –

Code = 7;

Name = "RequestHasApostil ";

Заявление/справка уже имеет заказанную и оплаченную апостиль!

ObjectResult = null

i)

Code = x;

Name = "Другой статус";

Другие возникшие статусы по время использования сервиса. В этом случае:

ObjectResult = null

**Пример запроса:**

```
<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:apos="http://apostil_service.x-road.ee/apostil">
  <soapenv:Header>
    <xro:protocolVersion>4.0</xro:protocolVersion>
    <xro:issue></xro:issue>
    <xro:id></xro:id>
    <xro:userId></xro:userId>
    <xro:service iden:objectType="SERVICE">
      <iden:xRoadInstance>central-server</iden:xRoadInstance>
      <iden:memberClass>GOV</iden:memberClass>
      <iden:memberCode>70000006</iden:memberCode>
      <iden:subsystemCode>record-service</iden:subsystemCode>
      <iden:serviceCode>ApostilConfirmPaymentTest</iden:serviceCode>
      <iden:serviceVersion>v1</iden:serviceVersion>
    </xro:service>
    <xro:client iden:objectType="SUBSYSTEM">
      <iden:xRoadInstance>central-server</iden:xRoadInstance>
      <iden:memberClass>YouMemberClass</iden:memberClass>
      <iden:memberCode>YouMemberCode</iden:memberCode>
      <iden:subsystemCode>YouSubsystemCode</iden:subsystemCode>
    </xro:client>
  </soapenv:Header>
  <soapenv:Body>
    <apos:ApostilConfirmPaymentTest>
      <apos:companyIdentifier>0000000-0000-0000-0000-
000000000</apos:companyIdentifier>
      <apos:documentCode>9075092410934101</apos:documentCode>
      <apos:paymentIdentifier>869faee9-0524-458d-b313-
3d04863b3bf8</apos:paymentIdentifier>
    </apos:ApostilConfirmPaymentTest>
  </soapenv:Body>
</soapenv:Envelope>
```