# Открытые API

Открытые API построены на основе контрактов данных и предназначены для предоставления возможности публикации открытых сервисов ФНС России для предоставления доступа зарегистрированным в ФНС внешним пользователям.

Открытые API представлены двумя сервисами:

1. Асинхронным сервисом сообщений (см. OpenApiAsyncMessageConsumerService-v0.1.wsdl)
2. Синхронным сервисом сообщений (см. OpenApiMessageConsumerService-v0.1.wsdl)

## Асинхронный сервис сообщений

Асинхронный сервис сообщений предоставляет 2 метода:

1. SendMessage – метод отправки сообщения;
2. GetMessage – метод получения сообщения.

### Метод отправки сообщения (SendMessage)

Метод отправки сообщения предназначен для отправки XML-сообщения в соответствии с определенной для конкретного сервиса схемой.

Сигнатура метода выглядит следующим образом:

SendMessageResponse SendMessage(SendMessageRequest request) throws AuthenticationException, где

SendMessageRequest – запрос, содержащий единственный элемент Message – сообщение, позволяющее содержать любой XML с обязательным указанием схемы. Например,

<CheckTicketRequest xmlns="urn://x-artefacts-gnivc-ru/inplat/servin/OpenApiAsyncMessageConsumerService/types/1.0">…</CheckTicketRequest>

SendMessageResponse – ответ, содержащий единственный элемент MessageId – идентификатор сообщения, присвоенный запросу, который требуется сохранить для дальнейшего обращения за ответом.

В результате вызова метод может вернуть исключение AuthenticationException, указывающее на то, что были переданы неверные аутентификационные реквизиты, либо что закончилось время их действия.

### Метод получения сообщения (GetMessage)

Метод получения сообщения предназначен для получения XML-сообщения в соответствии с определенной для конкретного сервиса схемой.

Сигнатура метода выглядит следующим образом:

GetMessageResponse GetMessage(GetMessageRequest request) throws AuthenticationException, MessageNotFoundException, где

GetMessageRequest – запрос, содержащий единственный элемент MessageId – идентификатор сообщения, присвоенный запросу, который требуется передать для получения ответа.

GetMessageResponse – ответ, содержащий 2 элемента: ProcessingStatus – статус выполнения запроса, и Message – сообщение, который позволяет содержать любой XML с обязательным указанием схемы.

ProcessingStatus – перечисление, с двумя значениями: PROCESSING – запрос обрабатывается, COMPLETED – обработка запроса завершена.

Элемент Message возвращается только в случае, если ProcessingStatus равен COMPLETED.

Например,

<ns:GetMessageResponse>

<ns:ProcessingStatus>COMPLETED</ns:ProcessingStatus>

<ns:Message>

<CheckTicketResponse xmlns="urn://x-artefacts-gnivc-ru/inplat/servin/OpenApiAsyncMessageConsumerService/types/1.0">

…

</CheckTicketResponse>

</ns:Message>

</ns:GetMessageResponse>

## Синхронный сервис сообщений

Синхронный сервис сообщений предоставляет 1 метод:

1. GetMessage - метод получения сообщения

### Метод получения сообщения (GetMessage)

Метод получения сообщения предназначен для синхронной отправки и получения XML-сообщения в соответствии с определенной для конкретного сервиса схемой.

Сигнатура метода выглядит следующим образом:

GetMessageResponse GetMessage(GetMessageRequest request), где

GetMessageRequest – запрос, содержащий единственный элемент Message – сообщение, позволяющее содержать любой xml с обязательным указанием схемы. Например,

<tns:AuthRequest xmlns:tns="urn://x-artefacts-gnivc-ru/ais3/kkt/AuthService/types/1.0">

<tns:AuthAppInfo>

<tns:MasterToken>MASTER\_TOKEN\_ISSUED\_BY\_FNS</tns:MasterToken>

</tns:AuthAppInfo>

</tns:AuthRequest>

GetMessageResponse – ответ, содержащий единственный элемент Message – сообщение, позволяющее содержать любой xml с обязательным указанием схемы. Например,

<tns:AuthResponse xmlns:tns="urn://x-artefacts-gnivc-ru/ais3/kkt/AuthService/types/1.0" >

<tns:Result>

<tns:Token>TEMPORARY\_TOKEN\_ISSUED\_BY\_FNS</tns:Token>

<tns:ExpireTime>2001-12-17T09:30:47Z</tns:ExpireTime>

</tns:Result>

</tns:AuthResponse>

## Правила именования и организации элементов в схемах сервисов

Правила именования элементов в схемах сервисов, если не оговорено дополнительно, соответствуют следующим правилам:

1. Элементы схемы вида MethodRequest предназначены для передачи в качестве запросов к соответствующим методам.
2. Элементы схемы вида MethodResponse предназначены для получения в качестве ответа от соответствующего метода. MethodResponse содержит 2 взаимоисключающих элемента Result и Fault. Result – возвращается в случае успешного вызова метода, Fault – в случае наличия ошибок.

# Порядок взаимодействия с открытыми API

1. ФНС предоставляет доступ к открытым API, путем выдачи мастер-токена, и адресов сервисов к которым предоставляется доступ.
2. Внешний пользователь, пользуясь синхронным сервисом сообщений, на предоставленный ФНС адрес сервиса аутентификации направляет SOAP-запрос, в соответствии со схемой сервиса аутентификации (AuthService-types-v0.1.xsd), передавая полученные идентификатор внешнего приложения и мастер-токен. В ответ возвращается временный токен и время истечения, выданного токена, до истечения которого временный токен необходимо обновить, направив повторный запрос.
3. Внешний пользователь, после получения временного токена, на предоставленный ФНС адрес открытого сервиса направляет SOAP-запрос, в соответствии со схемой открытого сервиса (например, для открытых сервисов ККТ KKTService-types-v0.1.xsd), также добавив в заголовки http:

FNS-OpenApi-Token: TEMPORARY\_TOKEN\_ISSUED\_BY\_FNS, где

FNS-OpenApi-Token – временный токен, см. п.2.

FNS-OpenApi-UserToken: QVBQX1VTRVJfSURfMQ==, где

FNS-OpenApi-UserToken – токен, уникально идентифицирующий пользователя в рамках внешнего пользователя (приложения), представленный в base64. Длина токена в base64 должна быть <= 160.

# Пример реализации клиента открытых API на C#

1. open-api-client\open-api-client.sln – солюшен проекта MS Visual Studio 2013
2. open-api-client\Program.cs – пример реализации клиента с комментариями
3. open-api-client\OpenApiMessageConsumerService-v0-1-soapui-project.xml – файл проекта SoapUI, для запуска мок-сервиса синхронного сервиса сообщений
4. open-api-client\OpenApiAsyncMessageConsumerService-v0-1-soapui-project.xml – файл проекта SoapUI, для запуска мок-сервиса асинхронного сервиса сообщений

Запуск проекта:

1. Импортировать в SoapUI проекты:
   1. open-api-client\OpenApiMessageConsumerService-v0-1-soapui-project.xml
   2. open-api-client\OpenApiAsyncMessageConsumerService-v0-1-soapui-project.xml
2. Запустить “OpenApiMessageConsumerServiceSoapBinding MockService”, остановить, запустить. При необходимости раскомментировать интересующий вариант ответа в “OpenApiMessageConsumerServiceSoapBinding MockService”.GetMessage.Response 1, при этом закомментировав предыдущий.
3. Запустить “OpenApiAsyncMessageConsumerServiceSoapBinding MockService”, остановить, запустить. При необходимости раскомментировать интересующий вариант ответа в “OpenApiAsyncMessageConsumerServiceSoapBinding MockService”.GetMessage.Response 1, при этом закомментировав предыдущий.
4. Открыть open-api-client.sln MS Visual Studio 2013, запустить проект в режиме отладки (F5)

# Коды возвращаемых сообщений

200 – фискальный признак корректен, фискальный документ найден, запрос выполнен успешно;

400 – не пройден форматно-логический контроль реквизитов фискальных документов;

404 – фискальный документ не найден в оперативном хранилище (сформирован более 2,5 месяцев назад);

406 – направленный фискальный признак не прошел проверку системы криптозащиты, поиск фискального документа прерван, дальнейшие попытки проверки запрещены;

503 – недокументированная ошибка в работе сервиса, для выяснения причин следует обратиться в техподдержку, указав: URL запроса к сервису, текст запроса к сервису, текст ответа от сервиса.