WIPO
ВСЕМИРНАЯ ОРГАНИЗАЦИЯ
ИНТЕЛЛЕКТУАЛЬНОЙ
СОБСТВЕННОСТИ

1 октября 2021 г.

ВЕБ-СЕРВИСЫ ГААГСКОЙ СИСТЕМЫ

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

ОБЛАСТЬ ПРИМЕНЕНИЯ

Настоящий документ содержит вводные сведения о веб-сервисах Гаагской системы (Hague Web Services, HWS), которые представляют собой межмашинный (M2M) интерфейс Гаагской системы.

ОБЩИЕ ПОЛОЖЕНИЯ

HWS - это безопасный и надежный API-протокол с высоким уровнем доступности, основанный на архитектуре HTTPS/REST и обеспечивающий обмен данными с Гаагской системой. HWS может использоваться для отправки или получения данных.

HWS может использоваться для:

- отправки решений или заявок, поданных не напрямую
- проверки статуса загрузки
- подачи запросов о статусе обработки
- получения Бюллетеней Гаагской системы
- получения конфиденциальных копий документов (только для ведомств ИС, осуществляющих экспертизу).

HWS – это рекомендуемый канал обмена данными с Гаагской системой. Соответственно, ведомствам ИС настоятельно рекомендуется использовать HWS с самого начала. Ведомства, уже осуществляющие обмен данными с Гаагской системой средствами ЭОД, в бумажной или иной форме, рекомендуется перейти на использование HWS.

АУТЕНТИФИКАЦИЯ И БЕЗОПАСНОСТЬ

БЕЗОПАСНОСТЬ API

Протокол API, применяемый в HWS, предназначен для межмашинного обмена сообщениями с *конфиденциальной полезной нагрузкой.*

Аутентификация пользователей производится с помощью ключа-подписи с асимметричным шифрованием, который представляет собой элемент Профиля безопасности API Security Profile 1.0 для финансовых систем. Профиль безопасности API для финансовых систем может применяться с API, используемыми в любой отрасли, где требуется повышенный уровень безопасности по сравнению с уровнем, обеспечиваемым стандартными протоколами OAuth или OpenID Connect. Таким образом, это профиль Oauth с повышенным уровнем безопасности, подходящий для защиты API с высокими имманентными рисками.

СОЗДАНИЕ ПАРЫ КЛЮЧЕЙ И ФОРМИРОВАНИЕ ИДЕНТИФИКАТОРА КЛИЕНТА (ID КЛИЕНТА)

На иллюстрации ниже показаны все этапы процесса регистрации идентификатора клиента API (ID клиента) и открытого ключа в ВОИС, а также открытого интернет-адреса клиентского приложения.

Действия ведомства:

1. Создать пару открытых и закрытых ключей (см. Приложение A: Создание пары ключей клиента в OpenSSL).

2. Сформировать сертификат x509 при помощи открытого ключа.

3. Запросить доступ к HWS, отправив сообщение электронной почты по адресу hague.it@wipo.int с приложением:

   (a) заполненной формы ВОИС (см. Приложение B: Форма запроса доступа к API, разработанная Департаментом ИКТ ВОИС);

   (b) сертификата x509.

Действия ВОИС:

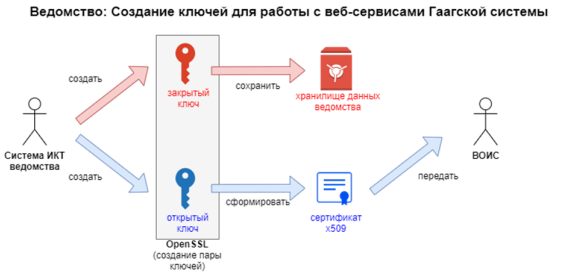
1. Получив указанную выше информацию, сформировать ID клиента.

2. Привязать открытый ключ к ID клиента.

3. Внести соответствующий интернет-адрес в «белый список».

4. Настроить HWS для авторизации запросов с ID клиента.

5. Подтвердить ID клиента ведомству ИС.



АУТЕНТИФИКАЦИЯ В HWS

Как только ID клиента, открытый ключ и открытый интернет-адрес зарегистрированы в ВОИС и сервисы HWS настроены, ведомство ИС может использовать API.

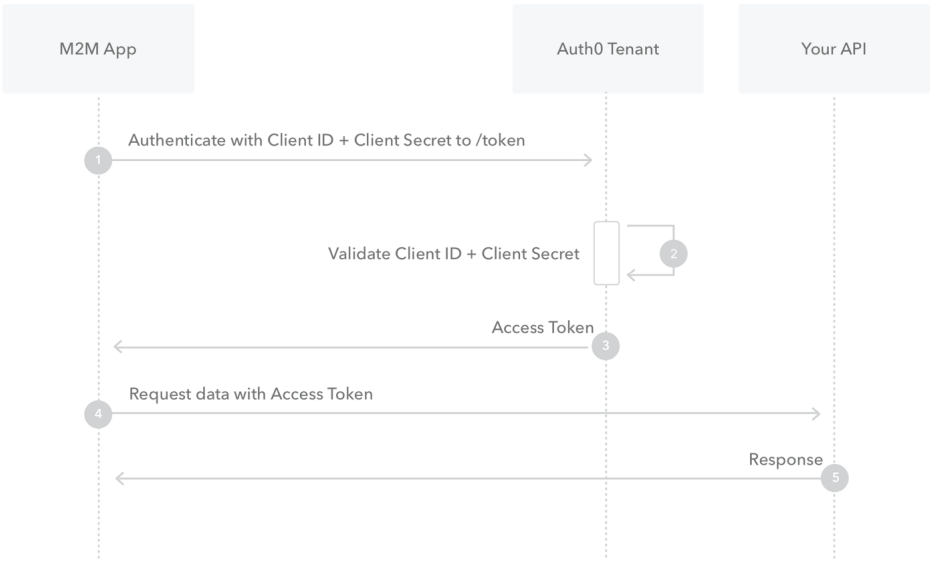
Схема взаимодействия показана на приведенной ниже диаграмме:

1. HTTPS-запрос участнику Auth0, содержащий ID клиента и токен JWT, подписанный закрытым ключом. Примечание: запрос **должен** поступить с интернет-адреса из «белого списка».

2. Подтверждение HTTPS-запроса, формирование токена доступа JWT.

3. В случае успешного завершения процедуры система выдает токен доступа JWT со сроком действия один час.

4. В пределах этого срока последующие вызовы конечных точек HTTPS могут выполняться с использованием того же токена доступа JWT.



Перевод легенды приведенной выше иллюстрации:

M2M App	Приложение M2M
Auth0 Tenant	Участник Auth0
Your API	Ваш API
Authenticate with Client ID + Client Secret to /token	Аутентификация при помощи ID клиента + секретного токена клиента
Validate Client ID + Client Secret	Подтверждение идентификатора клиента + секретного токена доступа клиента
Access Token	Токен доступа
Request data with Access Token	Запрос данных при помощи токена доступа
Response	Ответ

ОПИСАНИЕ API

В API HWS предусмотрены следующие конечные точки архитектуры REST:

1. Отправка заявок, поданных не напрямую, и решений в Гаагскую систему (POST/request).

2. Проверка статуса загрузки отправленной заявки, поданной не напрямую, или решения (GET /request/import).

3. Выяснение статуса конкретного запроса на обслуживание (Service Request, SR) (GET /request/{serviceRequestId}).

4. Получение бюллетеня (GET /publication/bulletin/{weekId}).

5. Получение конфиденциальной копии (GET /publication/copy/confidential/{weekId}).

Полная информация об API HWS (параметры, ответы и т.д.) имеется в Приложении D: API открытой части платформы Гаагской системы.

Вся полезная нагрузка формируется в соответствии со стандартом XML, используемом ВОИС, а именно: ST.96. Полная информацию о стандарте ST.96 v4.0 и определениях XML-схем размещена по адресу https://www.wipo.int/standards/en/st96/v4-0/.

Незначительное количество расширений, необходимых непосредственно для веб-сервисов, в настоящее время находится в процессе стандартизации.

Примечание: Для проверки связи как со стороны клиента, так и со стороны HWS может быть использована конечная точка под названием pingMe. Она не имеет иных функциональных задач и предоставлена исключительно для технического тестирования и проверки.

Отправка заявок, поданных не напрямую, и решений в Гаагскую систему (POST /request)

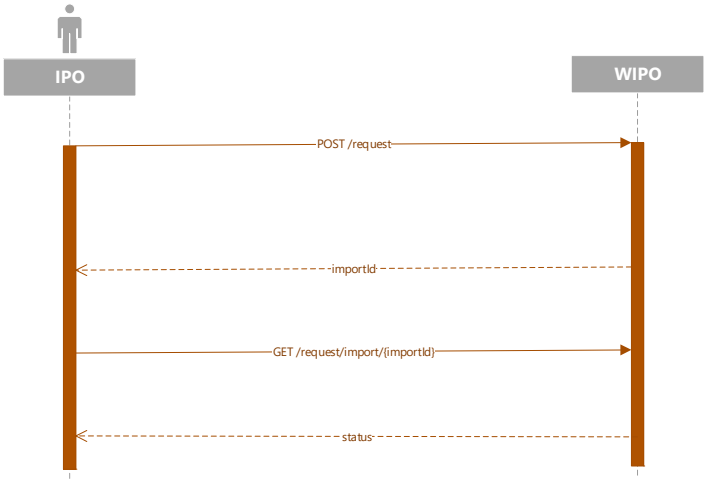
Отправка заявок и решений в Гаагскую систему осуществляется при помощи запроса POST, для которого информационным наполнением является загружаемый пакет (см. ниже).

В случае успешного выполнения система возвращает идентификатор загрузки для каждого пакета: это означает, что загружаемый пакет получен и будет обработан Международным бюро.

В дальнейшем данный идентификатор загрузки может использоваться для получения номера запроса на обслуживание (GET request/import), а номер запроса на обслуживание (SRN) может, в свою очередь, использоваться для выяснения статуса обработки запроса (GET request).

Информационное наполнение запроса, относящегося к заявке и решению представляет собой один zip-файл, содержащий файл XML стандарта ST.96, документы и изображения.

- Как указывается в XML-файле, к этим файлам должен быть указан относительный путь.
- zip-файл (то есть информационное наполнение запроса) должен содержать только одну заявку или одно решение.
- Информационное наполнение не может содержать более одного XML-файла.
- Образцы имеются по адресу ftp://ftpird.wipo.int/ST96_V_4_0_test/import-packages-4.0.zip.



Выяснение статуса обработки конкретного запроса на обслуживание (GET request/{serviceRequestId})

После загрузки пакета в Гаагскую систему операция именуется «Запросом на обслуживание» (SR) и ей присваивается номер запроса на обслуживание (NSR). Такой SRN можно получить при помощи конечной точки GET request/import (см. выше).

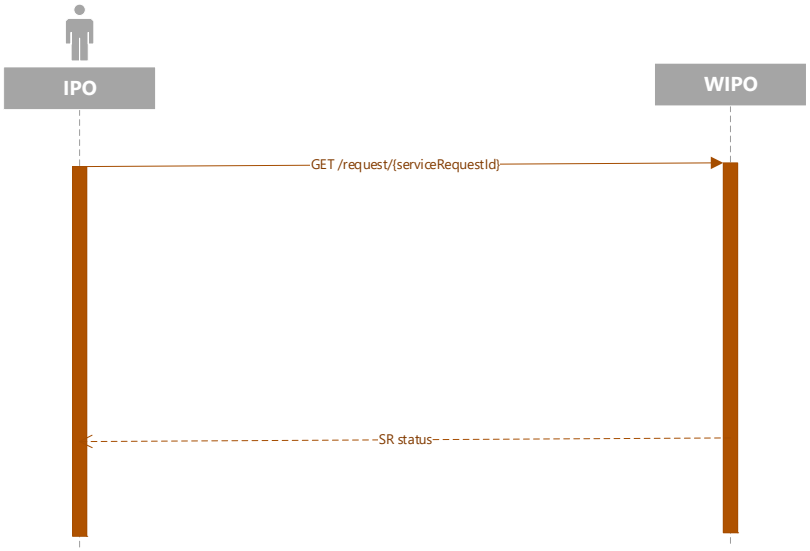
После присвоения SRN появляется возможность проверки статуса запроса при помощи конечной точки «запрос».

Возможные виды статусов:

- Не определен
- В обработке
- В ожидании ответа на запрос
- Зарегистрирован
- Отозван
- Аннулирован

В ответ на запрос о статусе запроса на обслуживание веб-сервисы Гаагской системы отсылают файл формата ST.96, содержащий:

- Идентификатор запроса
- Статус обработки
- SRN (Номер запроса на обслуживание)
- IRN (Международный регистрационный номер)
- в соответствующих случаях - ожидаемую дату публикации.



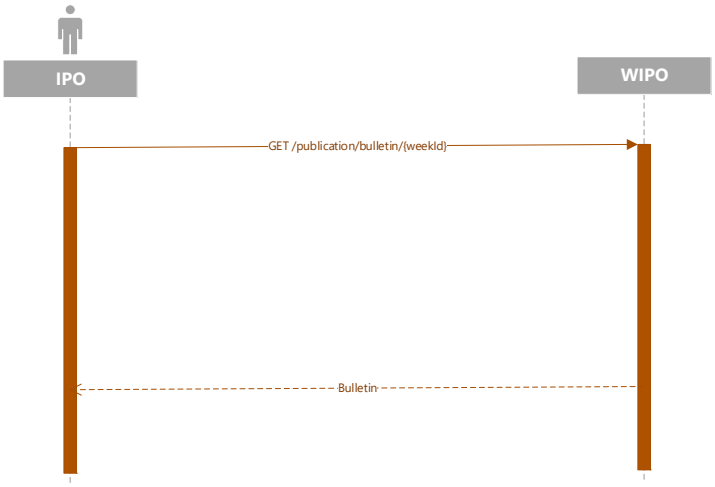
Получение бюллетеня (GET /publication/bulletin/{weekId})

Бюллетени выпускаются Международным бюро еженедельно, как правило, в пятницу вечером (по центральноевропейскому времени). Их можно запрашивать в любое время после их публикации. При этом параметр weekId должен иметь формат yyyyww.

Информационное наполнение ответного файла представляет собой zip-файл следующего содержания:

- библиографические данные бюллетеня в виде файла стандарта ST.96;
- папки с изображениями, соответствующие включенным в файл регистрациям или исправлениям изображений.

Примеры информационного наполнения для бюллетеней (конфиденциальные копии имеют такую же архитектуру) приводятся по адресу ftp://ftpird.wipo.int /ST96_V_4_0.



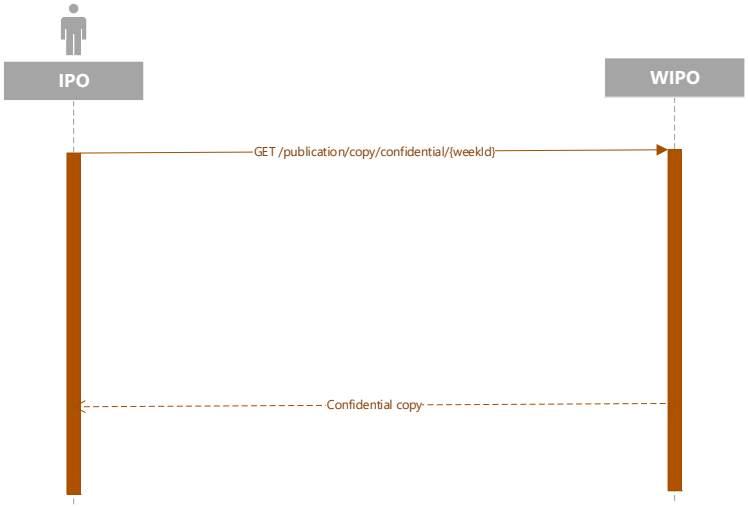
Получение конфиденциальной копии (GET /publication/copy/confidential/{weekId})

Конфиденциальные копии имеют ту же архитектуру, что и бюллетени.

Конфиденциальные копии выпускаются МБ еженедельно, как правило, в пятницу вечером (по центральноевропейскому времени). Их можно запрашивать в любое время после их формирования. При этом параметр weekId должен иметь формат yyyyww.

Информационное наполнение ответного файла – это zip-файл следующего содержания:

- библиографические данные конфиденциальной копии в виде файла формата ST.96;
- папки с изображениями, соответствующие включенным в файл регистрациям или исправлениям, внесенным в отношении изображений.


IPO
WIPO
GET /publication/copy/confidential/{weekId}
Confidential copy

ПРИЛОЖЕНИЕ A: ГЕНЕРАЦИЯ ПАРЫ КЛЮЧЕЙ OPENSSL – КЛИЕНТ

Создание пары асимметричных ключей (закрытый/открытый ключ) и сертификата x509 для межмашинной регистрации в ВОИС.

Генерация артефактов для регистрации по протоколу OIDC ВОИС

```
#!/bin/bash

# Set the environment
PRIVATE_KEY_ES256=hague4offices_private.pem
PUBLIC_KEY_ES256=hague4offices_public.pem
CLIENT_NAME=DAS

# Generates the ES256 keys
openssl ecparam -genkey -name prime256v1 -noout -out "${PRIVATE_KEY_ES256}"

# Extracts the public key
openssl ec -in "${PRIVATE_KEY_ES256}" -pubout -out "${PUBLIC_KEY_ES256}"

# Generates an x509 certificate
CERT_KEY_ES256=es256_cert.pem
OPENSSL_CONF=./openssl.cnf
CERT_CN="${CLIENT_NAME} private_key_jwt authentication"

# Build the certificate config file
printf '[ req ]\n' > "${OPENSSL_CONF}"
printf 'prompt = no\n' >> "${OPENSSL_CONF}"
printf 'distinguished_name = req_distinguished_name\n' >> "${OPENSSL_CONF}"
printf '[ req_distinguished_name ]\n' >> "${OPENSSL_CONF}"
printf 'CN = %s\n' "${CERT_CN}" >> "${OPENSSL_CONF}"

# Creates the x509 certificate
openssl req -x509 -new -config "${OPENSSL_CONF}" -key "${PRIVATE_KEY_ES256}" -out
"${CERT_KEY_ES256}"
```

1. Отправить в ВОИС файл **es256_cert.pem** для настройки доступа к веб-сервисам Гаагской системы (файл **hague4offices_private.pem** всегда следует сохранять в тайне и никогда не передавать другим лицам).

2. Дождаться, чтобы ВОИС направил **идентификатор клиента** и **информацию об области применения** после настройки.

3. Протестировать связь при помощи тестового клиентского приложения, предоставленного Гаагской системой (ссылка должна быть подтверждена).

ПРИЛОЖЕНИЕ B: ФОРМА ЗАПРОСА ДОСТУПА К API, РАЗРАБОТАННАЯ ДЕПАРТАМЕНТОМ ИКТ ВОИС

Приводимые ниже формы представляют собой предварительные версии ВОИС. Следует иметь в виду, что окончательная версия данной формы еще не принята и будет подтверждена дополнительно (01.09.2021).

Заполните данную форму, чтобы предоставить общую информацию о контексте запроса.

Общие сведения	
Тип запроса?	Запрос на создание ☐ Запрос на обновление информации ☐
Опишите обновленную информацию[1]	Контактная информация ☐ Интернет-адрес/интервал интернет-адресов клиента ☐ Сертификат ☐ Области применения ☐
Среда[2]	Эксплуатационный режим
Наименование приложения API	
Описание приложения	
Сетевой адрес (адреса) API	
Коммерческий собственник приложения	
Почтовый адрес[3] коммерческого собственника приложения	
Имя контактного лица по техническим вопросам, касающимся приложения	
Адрес электронной почты[4] контактного лица по техническим вопросам, касающимся приложения	
Кто будет пользоваться приложением?	Сотрудники ☐ Внешние пользователи ☐
Как обеспечивается защита API?	Использование токена доступа, полученного при помощи процедуры Oauth 2 Client Credentials

1 Данные сведения следует предоставлять только при направлении запроса на обновление.
2 Выберите среду.
3 Будет использоваться для уведомления о том, когда планируется реализовать компонент Oauth2 Provider в эксплуатационном режиме, что может повлиять на работу приложения.
4 Будет использоваться для уведомления о том, когда планируется реализовать компонент Oauth2 Provider в эксплуатационном режиме, что может повлиять на работу приложения.

Введите в данную форму информацию о Клиенте:

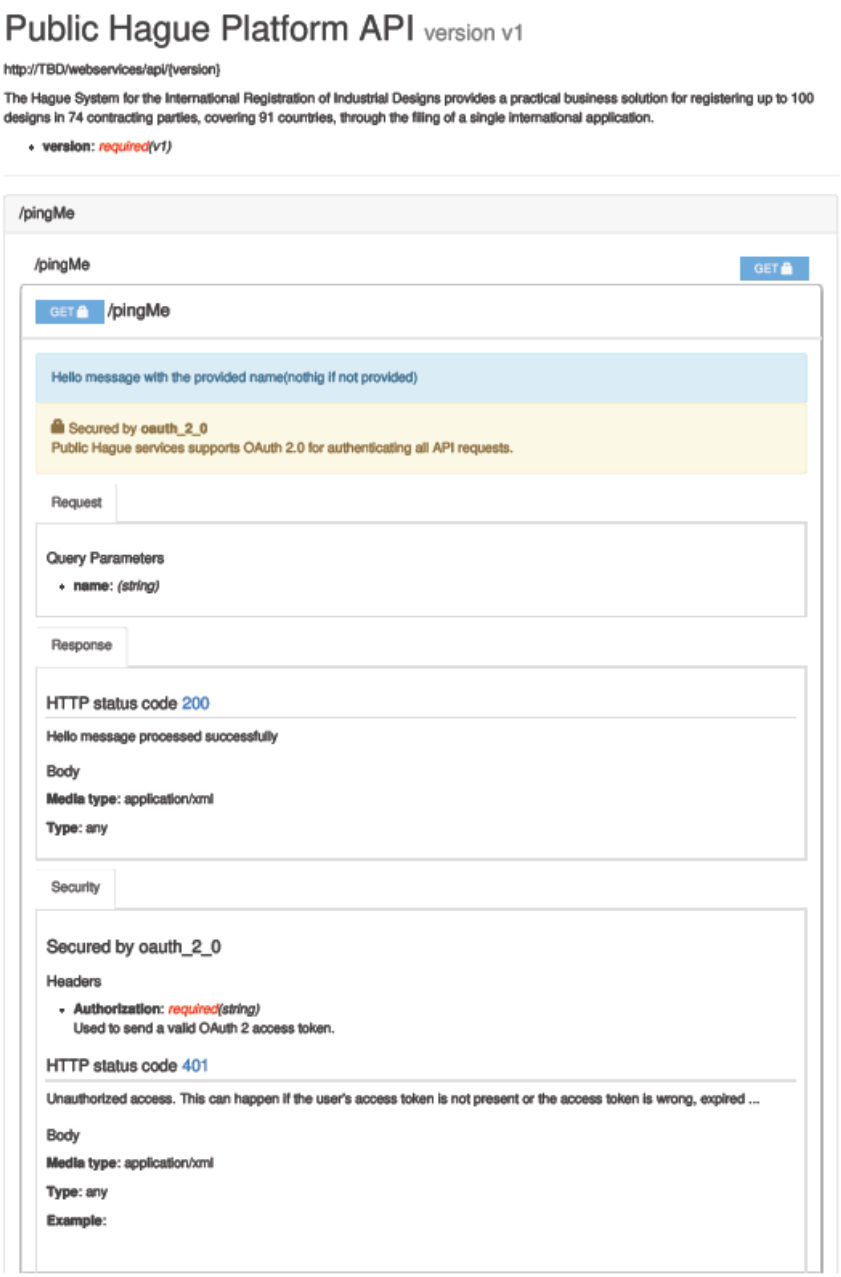
Защита API при помощи OAuth2	
ID клиента	Предоставляется ВОИС
Тип клиента	Конфиденциальный
Поддерживаемые области применения (необязательная информация)	Профиль по умолчанию
СЕРТИФИКАТ (X509V3 – ES256)	
ИНТЕРНЕТ-АДРЕС/ИНТЕРВАЛ ИНТЕРНЕТ-АДРЕСОВ КЛИЕНТА	
Метод аутентификации клиента	private_key_jwt (клиент отправляет свои данные как JWT)

ПРИЛОЖЕНИЕ C: ФРАГМЕНТ КОДА ДЛЯ ПОЛУЧЕНИЯ ТОКЕНА ДОСТУПА В DEV OPENAM ВОИС

Приводимый ниже bash-скрипт – это пример направляемого в ВОИС запроса на аутентификацию с закрытым ключом ведомства ИС:

```
#!/bin/bash
PRIVATE_KEY_ES256=es256_private.pem
CLIENT_ID=das-api-auth
SCOPE="das-api/das-access"
ISSUER="https://logindev.wipo.int/am/oauth2"

# https://logindev.wipo.int/am/oauth2/.well-known/openid-configuration
OIDC_CONFIG_JSON=$(curl -k "${ISSUER}/.well-known/openid-configuration")

# Generic way to obtain the token endpoint
TOKEN_ENDPOINT=$(printf '%s' ${OIDC_CONFIG_JSON} | jq -r ".token_endpoint")
UTC_TIME=$(date -u +%s)
EXP_TIME=$(expr "$UTC_TIME" + 10)
JWT_ID=Un1qu3i0

JSON='{'
JSON=${JSON}$(printf '"iss":"%s"' ${CLIENT_ID})
JSON=${JSON}$(printf ,"sub":"%s"' ${CLIENT_ID})
JSON=${JSON}$(printf ,"aud":"%s"' ${TOKEN_ENDPOINT})
JSON=${JSON}$(printf ,"exp":%s' ${EXP_TIME})
JSON=${JSON}'}'

JSON_HEADER_B64=$(printf '{"alg":"ES256","typ":"JWT"}' | jq -cj | base64 -w0 | tr -d
'\n=' | tr '+/' '-_')

JSON_PAYLOAD_B64=$(printf $JSON | jq -cj | base64 -w0 | tr -d '\n=' | tr '+/' '-_')

JSON_SIGNATURE_ASN1_B64=$(printf '%s.%s' $JSON_HEADER_B64 $JSON_PAYLOAD_B64 | openssl
dgst -sha256 -sign"${PRIVATE_KEY_ES256}" | openssl asn1parse -inform DER | base64 -w0)
JSON_SIGNATURE_HEX=$(printf $JSON_SIGNATURE_ASN1_B64 | base64 -d | sed -n '/INTEGER/p'
| sed 's/.*INTEGER\s*://g' | sed -z 's/[^0-9A-F]//g')
JSON_SIGNATURE_B64=$(printf $JSON_SIGNATURE_HEX | xxd -p -r | base64 -w0 | tr -d '\n='
| tr '+/' '-_')

JWT_ASSERTION=$(printf '%s.%s.%s' $JSON_HEADER_B64 $JSON_PAYLOAD_B64
$JSON_SIGNATURE_B64)

# echo $JWT_ASSERTION
# Access token private_key_jwt
# --insecure is only needed when testing within WIPO premises (because of the
proxy...)
curl \
  --header "Content-Type: application/x-www-form-urlencoded" \
  --data-urlencode "grant_type=client_credentials" \
  --data-urlencode scope="${SCOPE}" \
  --data-urlencode "client_assertion_type=urn:ietf:params:oauth:client-assertion-
type:jwt-bearer" \
  --data-urlencode "client_assertion=${JWT_ASSERTION}" \
  --url "${TOKEN_ENDPOINT}"
```

ПРИЛОЖЕНИЕ D: API ОТКРЫТОЙ ЧАСТИ ПЛАТФОРМЫ ГААГСКОЙ СИСТЕМЫ



Примечание: Более поздние версии этого документа будут содержать дополнительную информацию.