WIPO
世界
知识产权
组织

2021 年 10 月 1 日

海牙网络服务

目录

简介

范围

本文件是对海牙网络服务（HWS）的介绍，HWS 是海牙体系的机器对机器界面（M2M）。

概述

HWS 是基于 HTTPS/REST API 的协议，安全可靠，可用性高，用于与海牙体系进行数据交换。可用其发送或接收数据。

HWS 可用于：

- 发送决定或间接申请
- 检查导入状态
- 查询处理状态
- 检索海牙公报
- 检索保密副本（仅限于进行审查的知识产权局）。

HWS 是海牙数据首选交换渠道。因此，强烈鼓励知识产权局从一开始就使用 HWS。我们鼓励已通过 EDI/纸张/其他渠道与海牙交换数据的主管局迁移至 HWS。

验证和安全

API 的安全性

HWS API 是为带有保密有效负载的机器对机器通信而设计的。

该验证基于非对称密钥签名，这是金融级 API 安全配置文件 1.0 的一部分。金融级 API 安全配置文件可应用于所需安全级别高于 OAuth 或 OpenID Connect 标准的任何市场领域的 API。这意味着，它具有 OAuth 的高级安全配置文件，适合保护具有高固有风险的 API。

生成密钥对和配置客户端 ID

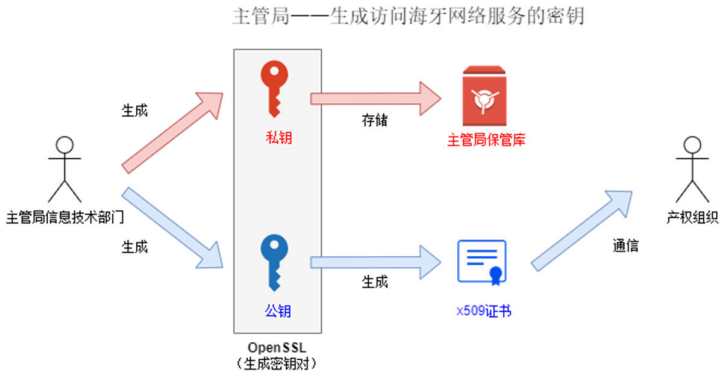
下图显示了注册产权组织 API 客户端 ID 和公钥的端到端过程，以及客户端应用程序的公共 IP 地址。

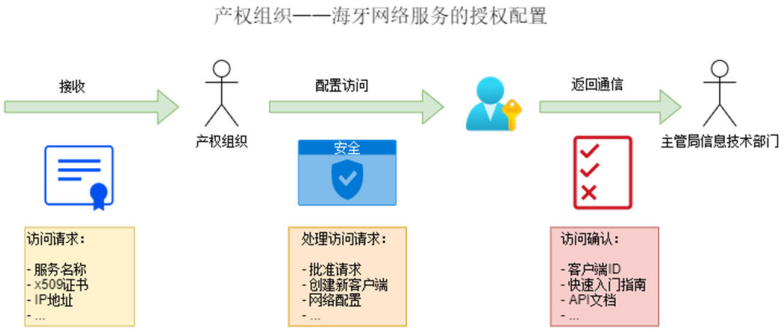
主管局的行动：

1. 生成一对公钥和私钥（见附录 A：生成 OpenSSL 密钥对的客户端）。
2. 使用公钥生成 x509 证书。
3. 请求访问 HWS，请发送包含以下内容的电子邮件至 hague.it@wipo.int：
   (a) 填写完毕的产权组织表格（见附录 B：产权组织信通技术部 API 访问请求表）。
   (b) x509 证书。

产权组织的行动：

1. 收到上述信息后，生成客户端 ID。
2. 将公钥分配/链接到客户端 ID。
3. 将 IP 地址加入白名单。
4. 配置 HWS 以授权对客户端 ID 的请求。
5. 向知识产权局确认客户端 ID。



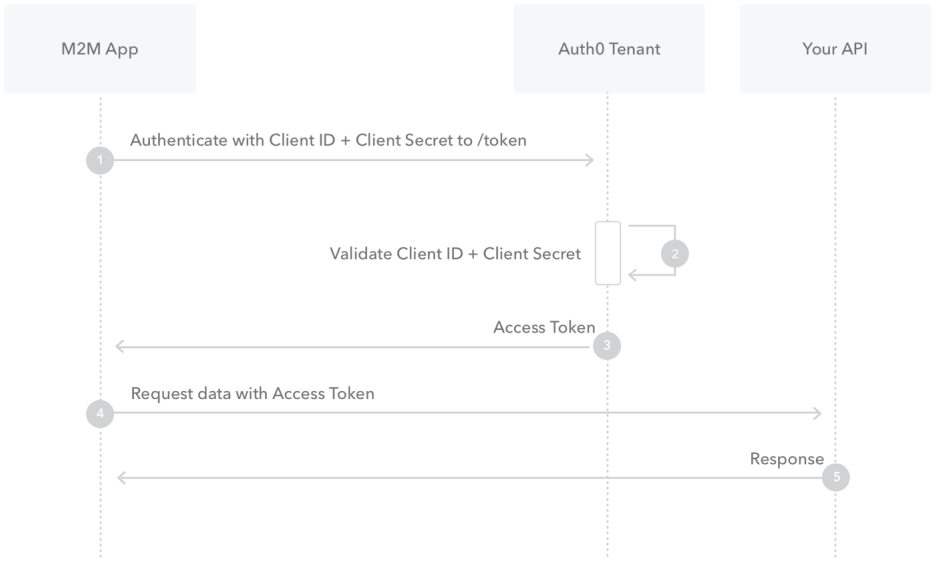


HWS 验证

当客户端 ID、公钥和公共 IP 地址在产权组织完成注册，并已配置 HWS 服务，知识产权局就可以使用 API 了。

下图显示了该互动：

1. 向 Auth0 租户发出 HTTPS 请求，并附上客户端 ID 和由私钥签名的 JWT 令牌。注意：该请求**必须**来自白名单上的 IP。
2. 验证 HTTPS 请求，并生成 JWT 访问令牌。
3. 一旦成功，就会返回有效期为一小时的 JWT 访问令牌。
4. 在过期之前对 HTTPS 端点的后续调用，可使用相同的 JWT 访问令牌进行。



上图文字翻译：

M2M App	M2M 应用程序
Auth0 Tenant	Auth0 租户
Your API	您的 API
Authenticate with Client ID + Client Secret to /token	用客户端 ID+客户端密码/令牌进行认证
Validate Client ID + Client Secret	验证客户端 ID+客户端密码
Access Token	访问令牌
Request data with Access Token	用访问令牌请求数据
Response	响应

API 说明

HWS API 实施了以下 REST 端点：

1. 向海牙发送间接申请和决定（POST /request）。
2. 检查已发送间接申请或决定的导入状态（GET /request/import）。
3. 查询某一特定服务请求（SR）的状态（GET /request/{serviceRequestId}）。
4. 检索公告（GET /publication/bulletin/{weekId}）。
5. 检索保密副本（GET /publication/copy/confidential/{weekId}）。

关于 HWS API 的全部细节（参数、响应等），可见附录 D 附录 D：海牙公共平台 API。

所有的有效负载都是基于产权组织所使用的 XML 标准，即 ST.96。关于 ST.96 第 4.0 版和 XSD 的全部细节，可见 https://www.wipo.int/standards/en/st96/v4-0/。正在对网络服务特别需要的小型扩展进行标准化。

注：名为 pingMe 的端点可用于检查客户端和 HWS 之间的连接。它没有任何功能，但可用于技术测试和验证目的。

向海牙发送间接申请和决定（POST /request）

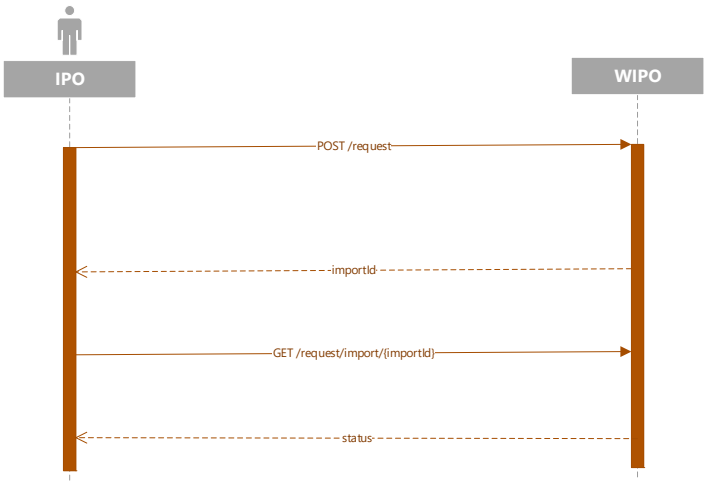
向海牙发送申请和决定是通过 POST 请求完成的，其中的有效负载是导入包（见下文）。

如果成功，将为每个包返回导入 ID，这意味着国际局收到了导入包，并将对其进行处理。

包的导入 ID 以后可用来检索服务请求编号 SRN（GET request/import），反过来也可以使用服务请求编号 SRN 检索请求状态（GET request）。

一个申请和决定请求的有效负载是包含 ST.96 XML 和文件及图像的单一 ZIP 文件。

- 这些文件必须位于 XML 中所表明的相对路径中。
- 一份 ZIP 文件（即一个请求的有效负载）必须只包含一项申请或一项决定。
- 一份有效负载不能包含两个或以上的 XML 文件。
- 样例可见：ftp://ftpird.wipo.int/ST96_V_4_0_test/import-packages-4.0.zip。



查询某一特定服务请求的状态（GET request/{serviceRequestId}）

包被导入海牙体系后，交易被称为服务请求（SR），并被赋予 SRN（SR 编号）。该 SRN 可以通过 GET request/import 端点进行检索（见上文）。

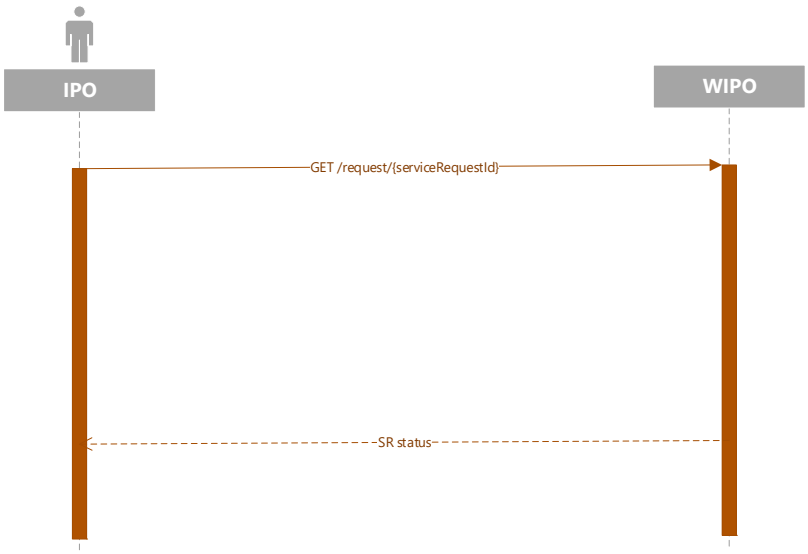
一旦 SRN 可用，便可使用 request 端点来检索请求状态。

状态类型包括：

- 未定义
- 正在处理
- 待规范化
- 已注册
- 已放弃
- 已撤销

作为对 SR 状态查询的回应，海牙网络服务会发回 ST.96 有效负载，其中包括：

- 请求 ID
- 处理状态
- SRN
- IRN（国际注册号）
- 预计公布日期（相关情况下）



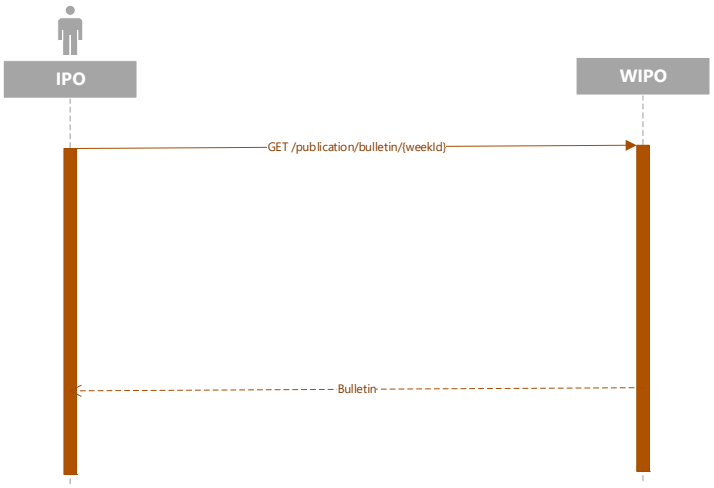
检索公告（GET /publication/bulletin/{weekId}）

国际局每周发布公告，时间通常是周五晚上（中欧时间）。从生成之时起，便可在任何时候对其作出请求。其中 weekId 的参数格式为 yyyyww。

响应的有效负载包含具有以下内容的 ZIP 文件：

- 作为 ST.96 文件的公报著录项目数据；
- 与所包含的注册或图像修改相对应的图像文件夹。

例如，公告的有效负载（保密副本具有相同架构）可见 ftp://ftpird.wipo.int /ST96_V_4_0。



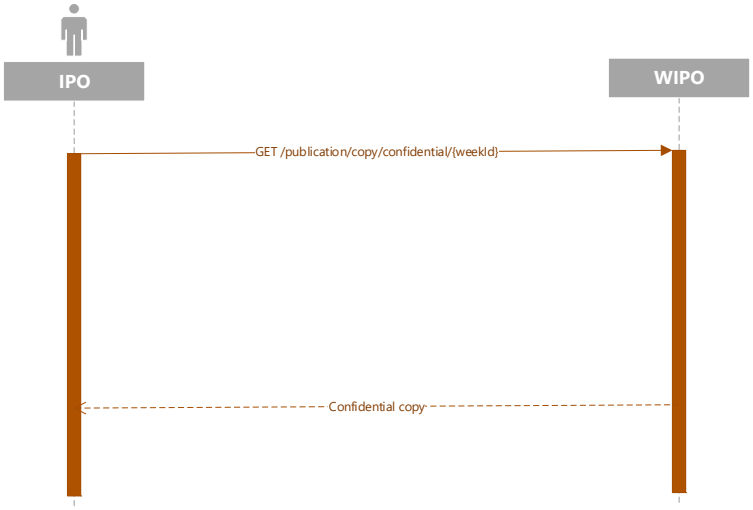
检索保密副本（GET/publication/copy/confidential/{weekId}）

保密副本的架构与公告相同。

国际局每周发布保密副本，时间通常是周五晚上（中欧时间）。从生成之时起，便可在任何时候对其作出请求。其中 weekId 的参数格式为 yyyyww。

响应的有效负载包含具有以下内容的 ZIP 文件：

- 作为 ST.96 文件的保密副本著录项目数据；
- 与所包含的注册或图像修正相对应的图像文件夹。



附录 A：生成 OPENSSL 密钥对的客户端

为产权组织机器对机器注册生成私钥/公钥非对称密钥对和 x509 证书。

为 WIPO OIDC 注册生成工件

```
#!/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. 向产权组织发送 **es256_cert.pem**，用于海牙网络服务访问配置（**hague4offices_private.pem** 应始终保密，绝不共享）。
2. 配置完成后，等待产权组织传回**客户端 ID** 和**作用域。**
3. 使用海牙提供的客户端测试应用程序测试通信（链接有待确认）。

附录 B：产权组织信通技术部 API 访问请求表

下表是产权组织的草拟表格。请注意，该表格的最终版本有待确认（01/09/2021）。

请填写此表，以提供有关背景的一般信息。

一般信息	
请求类型？	创建请求 ☐ 更新请求 ☐
描述最新信息[1]	联系信息 ☐ 客户端 IP/IP 范围 ☐ 证书 ☐ 作用域 ☐
环境[2]	选择环境
API 应用程序名称	
应用程序说明	
API URL(s)	
应用程序企业主	
应用程序企业主邮件[3]	
应用程序技术联系人名称	
应用程序技术联系人电子邮件[4]	
谁将访问该应用程序？	内部人员 ☐ 外部人员 ☐
如何保护 API？	使用通过 Oauth 2 客户端凭据流获得的访问令牌

1 请仅在有更新请求的情况下提供该信息。
2 请选择环境（下拉列表：Development / Acceptance / Production）。
3 将用于计划在生产环境（production）中部署 Oauth2 提供程序组件并可能对应用程序产生影响时发出通知。
4 将用于计划在生产环境（production）中部署 Oauth2 提供程序组件并可能对应用程序产生影响时发出通知。

请填写此表，以提供有关客户端的信息：

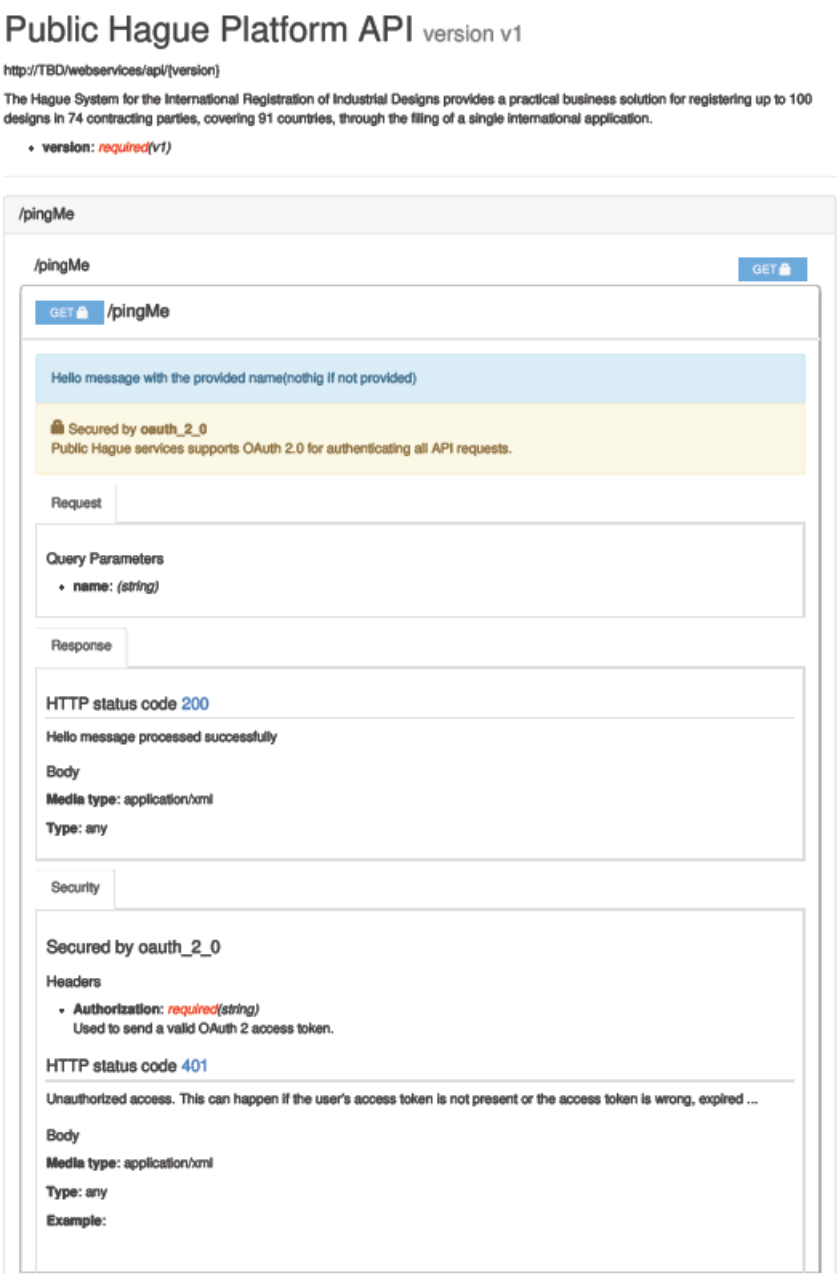
使用 OAuth2 进行 API 保护	
客户端 ID	将由产权组织提供
客户端类型	保密
支持的作用域（可选）	根据默认配置文件
证书（X509V3 - ES256）	
客户端 IP/IP 范围	
客户端身份验证方法	private_key_jwt （客户端将其凭据作为 JWT 发送）

附录 C：从产权组织 DEV OPENAM 获取访问令牌的代码片段

以下 bash 脚本是使用知识产权局的私钥作出产权组织验证请求的示例：

```
#!/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}"
```

附录 D：海牙公共平台 API



Public Hague Platform API version v1

http://TBD/webservices/api/{version}

The Hague System for the International Registration of Industrial Designs provides a practical business solution for registering up to 100 designs in 74 contracting parties, covering 91 countries, through the filing of a single international application.

- **version:** *required(v1)*

/pingMe

/pingMe GET

GET /pingMe

Hello message with the provided name(nothig if not provided)

Secured by **oauth_2_0**
Public Hague services supports OAuth 2.0 for authenticating all API requests.

Request

Query Parameters

- **name:** *(string)*

Response

HTTP status code 200

Hello message processed successfully

Body

Media type: application/xml

Type: any

Security

Secured by oauth_2_0

Headers

- **Authorization:** *required(string)*
  Used to send a valid OAuth 2 access token.

HTTP status code 401

Unauthorized access. This can happen if the user's access token is not present or the access token is wrong, expired ...

Body

Media type: application/xml

Type: any

Example:

注：更多信息将被添加到本文件的后期版本中。