

TECHNOLOGY FOR TAX MANAGERS

CA Prasenjit Palchoudhuri



INSIDE THIS ISSUE

PG. 2

Compliance by Design-OECD vision

PG. 3-4

Web-API and Programming API

PG. 4-5

Global examples and objectives attained

WHAT IS AN API

APPLICATION PROGRAMMING INTERFACE

The discussion paper '**Tax Administration 3.0**: The Digital Transformation of Tax Administration', published by the Organization for Economic Co-operation and Development's Forum on Tax Administration in 2020, speaks about the ideal digital tax administration of the future.

At the end of this discussion, as a Tax manager, you would be able to interact with internal technology teams and service providers in relation to the workings of APIs, by way of

- Understanding API and how it works;
- Obtaining the list of all APIs from technology or service providers;
- API customization to meet the compliance needs; *as to how effective in handling such discussions going forward*



OECD'S VISION

TOWARDS TAX ADMINISTRATION 3.0

This envisages a system as “**compliance by design**” whereby organization’s own systems as well as integration with external systems has to happen with the purpose of **Realtime reporting of taxes** in the immediate future. This where we as tax managers, we need to have a fair bit of understanding as what an API means. In a non-technical term APIs are how one application speaks to another. Sounds simple right? This is as simple as it looks.

Always remember, that your organization either have multiple internal applications (usually a standard norm) or depend on a vendor application to communicate with Tax Portals. An API makes an efficient environment where **one application is able to speak to another** by sending the required information and also obtaining the answers in most cases in a secured way, by way of either real time or batch method. This is usually a two-way communication.

Have you ever thought, that while booking for an accommodation for your favorite destination the booking app on your PC or mobile would seek certain details. Initially it would be using User ID and password, then the place where you want to go, the dates, no of persons(child/adults), twin bed, king bed etc. All the required information is passed through an API that is working behind the application.

```
"description": "Booking availability",
"variables": {
  "schema": {
    "enum": [
      "https",
      "http"
    ],
    "default": "https"
  }
}

s": {
  "/booking_availability": {
    "post": {
      "tags": [
        "Booking"
      ],
      "summary": "Booking API",
      "description": "To check if an advertised",
      "operationId": "booking-availability",
      "requestBody": {
        "required": true,
        "content": {
          "application/x-www-form-urlencoded": {
            "schema": {
              "$ref": "#/components/sche"
            }
          }
        }
      },
      "responses": {
        "200": {
          "description": "Booking Availabili",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/"
              }
            }
          }
        }
      }
    }
  }
}
```

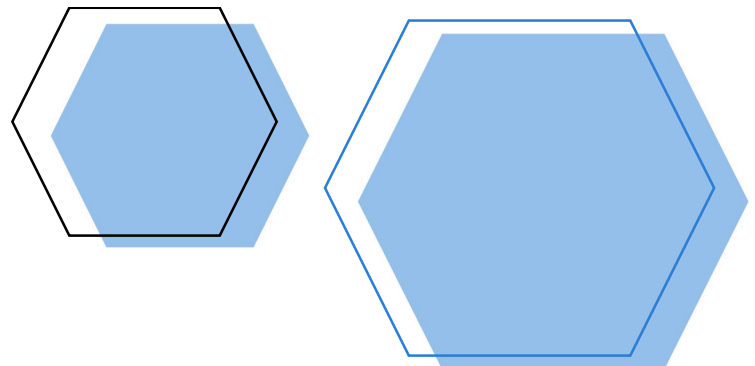
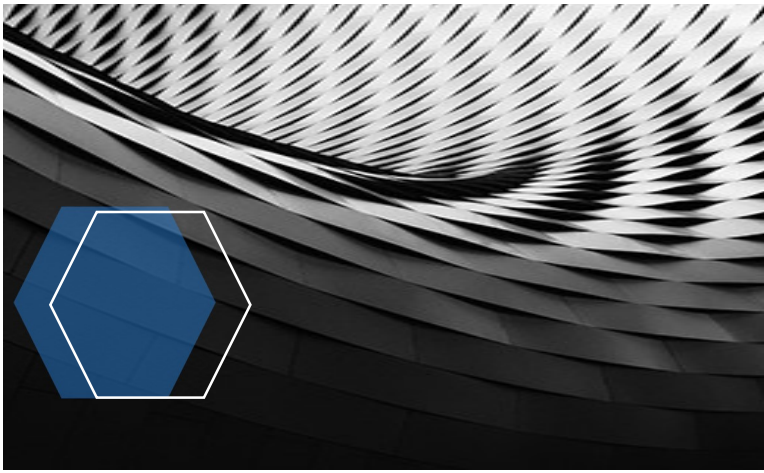


Fig 1



SIMPLE VANILLA API

WEB API

A **Web API** also called Web Services is an extensively used API over the web and can be easily accessed using the HTTP protocols. These are usually Open-Sourced API. Example are self-connecting travel bookings API or Google MAPs or an E-commerce website.

If you look on the Fig 1 in pg 2 above, this is an extract of an API JSON text, but without understanding JSON you would be able to understand, the description parameters read as “Booking Availability” as required from the server and the response is “Booking” and so forth.

The flow is given in Fig 3 in pg 3 on the left as to how the data moves.

Next, we will explain on an authenticate programming API gateway as used by Tax Authorities across the world, when you set up your electronic VAT reporting or e-invoice generation.

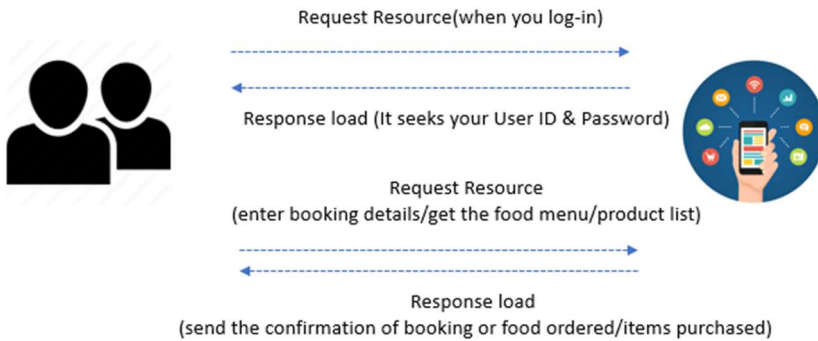


Fig 2

PROGRAMMING API

THE INDUSRTY NORM FOR TAX APPLICATIONS

This is bit complex and moves away from a simple open-source APIs and makes a remote program appear to be local by making use of RPCs (Remote Procedural Calls). Usually unlike Open-Sourced API these are based on validation and authentication. There are various types of API SOAP, Rest, JSON, XML based, but when we talk about Tax based most common are REST. It permits different format like HTML, XML, JSON, etc and making it easier to manage and configure.

Why REST API

- **Standard method HTTPS protocol** so while data moves, they are encrypted at both ends. In simple terms it gives the data best security while communicating with an external application.
- **Client server protocol.** In simple term this is either from your application/SFTP to vendor application or if you are directly integrated with the Government URL, then the secured channel between the two and the authentication methodology.
- **Front end back-end communication.** In simple term, you are able to retrieve data, review and post it to any corresponding server making it a two-way communication channel.

Note: OAuth 2.0, which stands for “Open Authorization”, is perhaps the most secured and de facto standard authorization, across organization, while speaking to an external application.

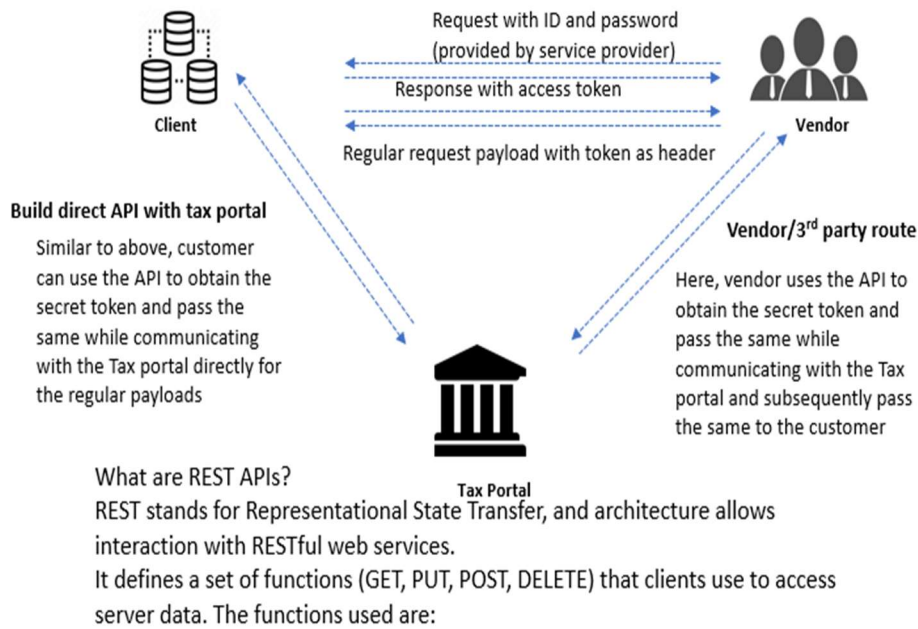


Fig 3

To explain in simple English, as per the flow fig 3 above, you will see that in either case your application(client's) can directly obtain the access token from the government tax portal or go via the vendor application. If you want to directly access the government portal, you would be using the grant token as was provided by government, and would be using the same every time while sending & obtaining information from the tax portal. In the other option, you may decide to use the vendor application to directly interact with government portal. In this case the vendor application obtains the access token based on your user ID and save in its application and then interacts with the government portal based on the access token granted earlier.

But as tax manager, your need to know how the **REST** Method works. Simply saying these are series of verbs **GET, POST, PUT, PATCH** and **DELETE**. From a pure tax managers perspective,

- **GET** request is to fetch a record from the server: If you want to obtain the details of the information of all the e-invoices details generated for a particular period. i.e. invoice No, UID or QR code;
- **POST** request to create a new record, so if you want to update a vat return detail for B2C sales which were not captured by B2B invoices on a tax portal, where the requirement is all your sales to match with tax portal details for finalizing the VAT return;
- **PUT** request to replace a record, a **PATCH** request to modify a record, which were originally details captured while creating the initial e-invoice details has to be modified due to any reason;
- **DELETE** request to delete the e-invoice details generated on tax portal, earlier which may be required to be cancelled.

Taking the example of **HRMC** approach in **MTD**, it has various APIs enable to retrieve information's and can be verified from the API documentation under UK HMRC site

- **Retrieve** VAT obligations
- **Submit & view** VAT return for period
- **Retrieve** VAT liabilities
- **Retrieve** VAT payments
- **Retrieve** financial details relating to a penalty. Any MTD complaint VAT software needs to have the capabilities to handle these APIs.

Also, various Tax authorities are coming up with more and more APIs and this may enable us to understand what this APIs are all about and what information are being fetched or provide by push and pull method.

For EU VAT , VIES **Batch validation** for validating the VAT registration numbers, a new, batch method of validating EU VAT numbers has been available on the European Commission's website (VIES).

On the right is the sample xml response for a valid number based on the match statement for VIES as to how for customer, the response payload passes through the API and let the sender know whether it's a valid VAT number or not.



```
{ { "isValid" : true, "requestDate" :  
"2024-03-09T11:01:06.305Z",  
"userError" :  
  
"VALID", "name" : "XXXXXX",  
"address" : "\XXXXXX  
00664\XXXXXXXXXX\n",  
"requestIdentifier" : "",  
"originalVatNumber" :  
"8XXXXXXXXX",  
"vatNumber" : "XXXXXXXXXXXX",  
"viesApproximate" :  
  
{ "name" : "---", "street" : "---",  
"postalCode" : "---", "city" : "---",  
"companyType" : "---",  
"matchName" : 3, "matchStreet" : 3,  
"matchPostalCode" : 3,  
"matchCity" : 3,  
"matchCompanyType" : 3 } }
```

As a Tax manager, you can simply read the xml, as “isValid” is the response as to say whether the number is valid based on the “input data” which are captured under different attributes like "originalVatNumber" , "vatNumber" , "name" , "street" , "postalCode" & "city" as inout data, as send through the API.

While interreacting with vendor, you can also ask them **to list out all the APIs provided by the application as well as the tax authority**. Ideally vendor's application must handle all the available APIs, as provide by the tax authority.

For example, for **India GST**, the government has provided multiple APIs like **Registration Module API, View ledger details AP, Payment details API**, miscellaneous activities other than the standard return filing APIs (**where you can check the HSN code of product too**). As a tax manager you can always obtain the list of APIs and ask the vendor about its capability to integrate in its application. Always remember this APIs can be modified to your need and its vendor agnostic. That means your own team can manage and build the API interaction and not depend on a 3rd party application. This related to the first figure where you connect the APIs directly with the Tax portal.

In conclusion, this article is meant for tax managers who can ably participate in a vendor discussion or even discussing with its own technology team in understanding how the API works and what could the requirements from compliance and reporting perspective. As Tax manager you are now confident to discuss

- ✓ How the APIs work;
- ✓ To decide whether you need a 3rd party vendor or use your own technology team to build it and reduce less dependency on external vendor;
- ✓ How a vendor solution with communicate with your own application and the tax portal and the authentication;
- ✓ To what extent vendor is able to scale to the need of the tax department as per the requirements from the tax authorities as per development APIs provide by Tax offices.