Frontier Software

Integration

SINCE 1983

Frontier software

www.frontiersoftware.com

Integration – At a Glance

Technology and software form the backbone of business operations. Few modern businesses function without finance and payroll systems, or internet connectivity. As businesses grow in size, we can expect to see HRIS, Learning Management, Time & Attendance and ERP solutions being implemented. But the benefits delivered by this dynamic technology landscape come with a new challenge; integrating systems so that data is shared when and how it is required across the entire ecosystem.

For HR and Finance professionals, the need to integrate is often made more complex as they begin to grapple with the terminology and technicalities involved in getting one system to talk to another. When engaging their IT team, they must first develop an understanding of the technical jargon associated with the task before they can even begin to imagine a solution.

This Whitepaper is designed to assist non-technical staff to understand some of the common terminology that they will encounter in their drive for system integration. It then goes on to explain the options available to businesses wishing to integrate existing systems with a Frontier Software solution.

Each option will be provided with an assessment of features and restrictions to further inform decision making. This Whitepaper is not intended to replace the expertise of in-house IT personnel, or provide an exhaustive overview of current thinking in IT. It is designed to provide a basis on which further research and investigation can be made and to aid in conversations with your IT team.

What is an API?

API stands for “Application Programming Interface” and is a term that is commonly used when discussing integration. But it’s not often easy to grasp the concept. Here’s a way to think about API’s using a restaurant analogy[1]. Diners at a restaurant are not permitted to enter the kitchen. In order to know what food is available they are given a menu. After reading the menu, they make an order to a waiter whose job is to pass it to the kitchen and to return to them with what they have requested. The waiter can only deliver what the kitchen can provide.

In technical terms, the waiter is the API. Diners are requesting a service and become the customer (or consumer) of the API. The menu lists what they are permitted to request from the kitchen. The kitchen is the server that satisfies the request. It can only deliver what it has on hand and can create. An API, then, is a bridge that connects the customer (user) with the consumable item (data).

The description above explains what is referred to as a REST API. The API is a digital middleman. It makes requests on behalf of others but does not permit direct access to the database where the information is stored. An API can only deliver what is available and it must use the right words when making the request, just as you would use your manners when asking for help in a retail scenario.

The other type of API is called a real-time API. It does not employ a request-response process. Instead, it simply sends data to registered recipients on a regular basis. An example may be a ride-sharing app that shows you, in real time, the location of the driver in relation to you and provides an estimate of their arrival time.

[1] https://medium.com/apinf/apis-for-non-techies-like-myself-259f60042ba

What is a Server?

A server is a computer that is connected to the internet and runs all the time. Its main tasks are to store data and facilitate communication. A server is a computer that only ‘talks’ to other computers, so it doesn’t have an engaging interface like that of a telephone or tablet. It doesn’t need to because it only deals with its own kind.

Servers can be accessed by apps, which are designed to be used by humans. If a human-facing app, such as messaging, is the front end of your user experience, then the server is the back end. It permits the app to ‘talk’ to it via an API, which understands how to take requests from the app and pass them to the server. Reinforcing the kitchen scenario above, the server is the kitchen area where the request is processed. The server is always listening for requests and can store the responses, whereas an app can be switched off and on to suit the user. That explains why your phone lights up with messages when you disembark an aeroplane. The server was busy receiving the message data while the phone was in aeroplane mode and held onto it until landing when the app asked, “do I have any messages to display”?

When processing requests from an API, the server may access a database. A database holds all information that is captured by a software program. The database will provide the server the raw information that it uses to satisfy a request from an API. Extending the kitchen scenario, the database would be the cold stores where the raw ingredients are stored. The kitchen takes the ingredients and combines them in order to satisfy customer orders from the menu.

What are Web Services?

Web Services is a term used to describe the dynamic interaction between two web-based applications that use a shared language by which to request and respond. The type of platform or programming language each system employs is not relevant because the interaction occurs in language that is universally accepted and understood by all. This is referred to as open standards. To illustrate, a German and an Italian who must collaborate on a project may not be able to do so unless they both use a more widely spoken language, such as English, as the basis for communication. As the name implies, the communication occurs over the World Wide Web.

What is SFTP?

FTP, or ‘File Transfer Protocol’ is a technology that enables parties to send and receive data files to and from a remote file server. This is the process that underpins file transfers. Data that is to be shared between systems is sent to a server that can be accessed by the party needing to collect it. Data sent this way may be stored in a secure server, but is not encrypted during transmission. Potentially, that means that any data in transit runs the risk of being intercepted and read. SFTP refers to ‘Secure File Transfer Protocol’. SFTP provides in-transit encryption so that even if some of the data was intercepted in transmission, it would remain unreadable.

Understanding your Integration options with Frontier Software

When companies begin to contemplate the integration of different, disparate software applications, such as HRIS, Time & Attendance, ERP and payroll systems, consideration will be given to the most appropriate way to achieve the goal. Typically, Frontier Software clients seek data-to-data synchronisation whereby third party systems are integrated to ensure all data relevant to respective systems is the same, or ‘in sync’.

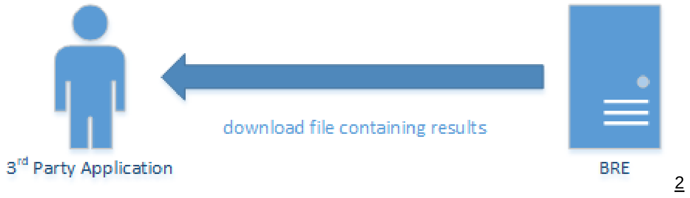
Frontier Software has two distinct methods by which data integration may be achieved, each offering their own pros and cons.

File Transfers?

Put simply, a file transfer describes a process by which one server (eg. ichris) provides information to another server, (eg. an LMS), in the form of a file it can upload, (eg. completed online training). The exchange of files is via SFTP. There are 2 scenarios that may be accommodated.

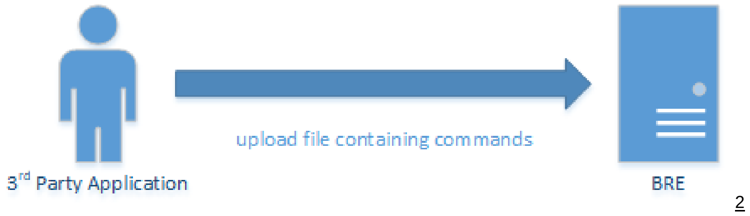
a) Single-direction Frontier to 3rd party.

In this scenario, ichris will generate an output file of required information at a scheduled date and time. It then stores the file on a file server to which the third party has access. The 3rd party software can then access the file and upload it in order to refresh its records, eg. where salary data for new employees is sent to an HRIS to maintain global records.



b) Single-direction 3rd party to Frontier.

In this scenario, the 3rd party system will generate an output file of required information at a scheduled date and time. It then stores the file on a file server to which Frontier Software has access. ichris will then access the file and upload it to refresh its records, eg. course completion data from a LMS is used to update skills matrices in ichris.



[2] Where BRE = ichris

Features & Restrictions of File Transfers

Features	Restrictions
No programming interface required.	ichris server must be made accessible to the 3rd party application.
Relatively cheap and quick to implement	Direct access to the ichris File Server is required by the 3rd party.
Suited to scenarios where data exchange is not time critical.	Cannot implement as an on-demand service. File collection can only be done at scheduled intervals.
	Data files cannot be encrypted.

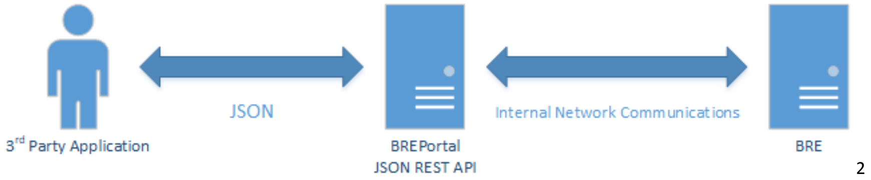
Web Services

Web Services removes the process of file server retrieval and upload from the integration equation. Instead, Web Services allows two systems to 'talk' directly to one another to exchange data. Unlike, file transfers, Web Services permits bi-directional exchange of data. As mentioned above, Web Services don't concern themselves with the platform or programming language employed by the respective systems. Instead, they use universally recognised languages in order to communicate with an API, which then makes the request and takes the response from the server. Frontier Software offers 2 web services options.

c) JSON REST API

Earlier in the whitepaper, we have discussed API's and compared REST API's to Real-Time APIs'. Put simply, JSON, (or JavaScript Object Notation), is an open-standard file format that is used to send the request/response commands to the API. Even more simply, it deals with 'how' the commands are expressed.

This Web Service Configuration is asynchronous. The best way to understand this benefit is in terms of a request/response scenario. A Synchronous exchange relies on a request being sent and then waiting for a response. Nothing can happen until the response is received. Given the volume of requests an API may receive and the potential time lag between receipt and response from a server, this can be a limiting option. Asynchronous design does not restrict the activity of the server making the request of the API until a response is received. It simply sends the request and continues its normal processing until a response is received.

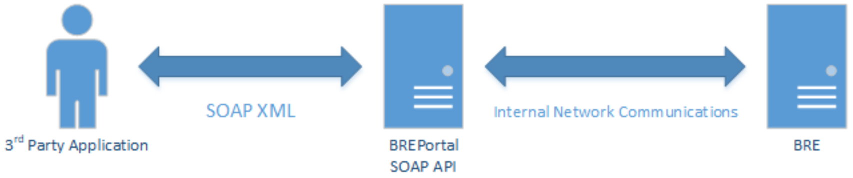


Features & Restrictions of JSON REST API

Features	Restrictions
Asynchronous access.	Requires users to have IIS (Internet Information Server) installed on their Web Server.
Considered to be easy for programmers to read and write commands.	Must have IT staff capable of managing the Web Services coding at the client end.
Suited to scenarios where timely, immediate data exchange is required.	
Fastest bi-directional access to ichris.	
Permits fully encrypted communication.	
Scalable – Can accommodate future functionality and system growth over time.	
ichris server is isolated from the 3rd party application by use of an API.	

d) SOAP Web Service API

As mentioned above, SOAP is another open standard protocol that uses XML in order to write and execute request/responses. SOAP is an older style of Web Service API and may be found in older or existing 3rd party systems that already use Web Services with other providers. Although it provides largely the same outcome as the JSON REST API, it is not asynchronous in nature.



2

Features & Restrictions of SOAP Web Service API

Features	Restrictions
A simpler API.	Offers synchronous access only. Can lead to delays or timeouts on some requests.
Suited to scenarios where information exchange needs to be relatively quick.	Must have IT staff capable of managing the Web Services coding at the client end.
Supports bi-directional access to ichris.	
Permits fully encrypted communication.	
ichris server is isolated from the 3rd party application by use of an API.	

The approach you elect to take to integrating your other systems with Frontier Software requires you to analyse your requirements. Areas to consider should include:

Timeliness of data

Does data updating need to happen immediately, or would another cycle, eg. daily, weekly suffice?

The nature of data exchanged

Is the data sensitive? Is there a chance unprotected data could raise privacy concerns?

Consideration of future-state requirements (Scalability)

What plans do you have for 3rd party system extension? Will new functionality create new data that will need integrating?

Budgets

Development (programming) work is required to create Web Services connections. Implementing Web Services is a more expensive option than implementing file transfers.

Appetite for programming

To implement Web Services, you will need IT staff with the capacity to program and support web services at the client end.

Frontier Software has considerable experience in integration and the implementation process involved. No matter what choice you make regarding the integration of data between Frontier Software and your third party systems, we have the expertise to assist you.

“Innovation, functionality, flexibility and quality are the foundation of our software solution”

Michael Howard - Founder, Frontier Software

Frontier
software