What is an API? What is an Interface? Aren’t they the same thing?

The short answer…yes, sort of.

Let’s start with what they are and where we may have seen them in action.

Interfaces

For this discussion, an *interface* is how person or thing interacts with another thing. It has an initiator that requests an action and the responder – a device, application or appliance – executes the response.

It could be as simple as the control panel on a piece of equipment – changing settings on the washing machine and starting it.

It could be as complex as a multi-function application like Amazon.com.

Have you ever shopped on Amazon? Then you have used the very sophisticated Amazon Marketplace User Interface. It is structured, branded, personalized, responsive, and organized.

Things are specifically positioned on the screen, you see recently viewed or purchased items advertised or on sale; you have your own account.

The intent is to provide a shopping experience where you are most likely to buy. This interface includes the use of behavior tracking, purchase history, inventory insight, price evaluations, and charity opportunities…just for starters.

Interfaces are everywhere, with more popping into our lives daily. When you open your smart phone, you see a very powerful user interface that lets you access tools and applications. We call them ‘apps’, which is short for ‘application’. From there you can continue to multiple other interfaces – one for each app. Your text message app has one, the email app has one, the internet browser has another, and so on.

Each of these interfaces is specific to the app that supports it and gives the user access to the functionality in that application. You, the user, are the initiator of actions and requests. The apps are the responders.

APIs

API is an acronym for *Application Programming Interface*. Note that ‘Interface’ is in its name.

An API is one type of interface, but an interface may not be an API.

Think of the interface as a pool, the API is one swimmer in the pool. The API is a specialized initiator, just one possible way to get a request fulfilled.

Back to your smart phone, the text message application works with the operating system of the device to get the messages off the phone and out to a destination – the other person's phone. For this the app uses an API, it uses the iOS, Android or Windows device-specific API.

Have you seen the smart home technology that makes lights come on and off, temperature settings adjust, blinds open and close? Do you have an app to let you lock your doors?

These all use APIs to 'talk' to each other, sending messages that cause the other device to act in a specific way.

This gets interesting when we realize that Amazon's Alexa and Apple's Siri are interfaces, but they respond to our voice alone. While we know of their human interface, they work with other devices via APIs.

An API requires no human interaction and messages are sent back and forth all the time.

Information is shared between these apps through a defined API.

APIs are machines talking to each other in code, there are no human decisions here. It is one computer program interacting with a different one by submitting requests that are built in a pre-defined manner.

A client can build their API from anywhere. It could be:

- An application in their customer rewards website.
- A function of their employee rewards and incentives application.
- Part of their HRIS (Human Resources Information System)
- A standalone application that uses extract files from other applications to process requests.

Client Options

We now focus on the Gift Certificates Incentives Web Service API and the GiftCertificates.com website for client options and examples.

Client users can interact with the GiftCertificates.com website through the public user *interface*. On the site, they create an account, log in and order items online.

The purchase is simple and direct, without programming or specialized knowledge of how things work. The user interface is intuitive, people expect to see a shopping cart and a checkout button to make their purchases.

Businesses can expand that capability to purchase rewards with the Gift Certificates Incentives API. The purchasers can be employees, administrators, or a client's external customers. This access is depends on 1) their choice for an interface or an API and 2) how much control they need or want and 3) how much development they are willing to undertake.

Interface Options

1. Use the public GiftCertificates.com website interface to order awards for their employees or customers. This is the public interface and free to use by anyone and may be a great solution for small or unique orders by any type or size of business.
2. Use a proprietary interface with their own URL, branding, colors and award selections to allow their employees or customers to redeem or purchase awards. This site is built for them by GCI and fees are associated with this option.

   For companies that want something reflecting their company brand and values, this is an excellent option. It has been used in partnerships with other businesses who want to provide incentives.

   Example: InterActive Health – this group provides a wellness management program to businesses. They wanted to incorporation rewards delivery and redemption to their offerings. Using a website branded for them but built and maintained by GCI allowed a shorter time to market, known reliability, and affordable start-up. This solution also uses an abbreviated version of the GCI Web Service API to complete the transaction. See the API section for details on that segment of their implementation.

API Options

1. Integrate the purchase of awards into an internal system using the Gift Certificates Incentives API. There are two distinct levels currently seen in the GCI client roster:
   a. Via a custom user interface or integration with another system, clients build their own API that can place orders, track and manage activity by human intervention or triggered events such as an employee work anniversary. They can use as much or as little of the available functionality as their business needs dictate.

      This solution is chosen by enterprise clients and those businesses which desire integration with their internal HR systems such as PeopleSoft, EncoreHR, IBM Kanexa, Kronos, etc.

   b. Via an abbreviated version of the API functionality, clients can hybridize the Interface-API to fulfill their requirements.

      Example: InterActive Health uses a very short list – about 2-3 - of the available functions in the API. Their initial purpose was only to transfer the reward value from their client-specific wellness management system to GCI, allowing users to receive their notification and begin redemption almost instantly.
2. Build their own request and redemption site: pulling catalogs and merchant information to populate the pages and submitting, tracking, and managing orders.

   This would be a custom client interface also using the API to fill the site with information, place, track, and verify orders. It would require significant development efforts from the client.

This is generally a client that wants to manage a full rewards system for their customers or a third-party provider.

This scenario is rare, with specific requirements. Please gather their primary requirements and schedule time with GCI Technology Services group before presenting this option to a client.

When to Propose an Integration?

1. When a client wants their own branding - GCI Web Interface; or their own site - GCI Integration Services API.
2. When they want to automate reward creation or purchase, particularly if it is through their HR system or from an extract file from a similar system - GCI Integration Services API.

API Technical Basics – A Peek Under the Hood

In the API options, the client would build an application to bridge their system to GCI. They can build their application in any programming language they choose, but will need to understand web services.

Web services use defined methods, standards and technology – HTTP, SOAP, and XML – to pass messages back and forth. These are termed a **request-response message pair**.

The Gift Certificates Incentives Web Service API accepts and responds to messages. It communicates across the internet, one web application to another.

Each request-response pair between the two applications completes a message. Every message request must include the **client credentials** or the web service will not process the request. This credentials check is their user id and password login.

Included in the message is the **method** request. A method is a named transaction that fulfills a specific purpose and has a defined group of properties. An example is the Create method. Create does exactly what it says: it creates an order. Inside the Create method is a list of the fields containing the data needed to build the order in the GCI system.

Example of a Web Service API transaction:

Client A has an API and sends a Create request to the GCI Web Service.

Request: The client application builds the request message for Create. This includes:

a) The ExternalCredentials method at the top of the request. This provides the client authentication information to the web service.

b) The Create method will follow. This will contain all the details about the order being created, submitted, processed and shipped.

c) The client application bundles everything together and sends the entire message to the GCI Web Service via a standard internet protocol (HTTP) request.

Response: The GCI API receives the message and opens it.

a) First it verifies the ExternalCredentials – confirming that the client has access to the web service.

b) It recognizes Create, and takes the details, tests the information and processes the request.

c) It packs up the response message and sends it back to the client application. This would include the order ID and status fields to let their application know if the request processing is a success or a failure. If there is a failure, additional details are provided to identify the issue.

That completes the request-response pair of messages between the client application and the GCI web service API.

While this may sound simple, the requests themselves can become complex. This is especially true for multiple ShipTos or Recipients in a single message. These messages are layered into themselves and can create long and detailed requests. It is recommended that developers with some experience in web services be engaged to build these applications.

Further Information

For a deep dive into the methods used for requests, review the API documentation. A client will do this to build their own application. Understanding the certificate types, SKUs, message and occasion sheets, files, and members is necessary to implement a functional application. This is where GCI staff knowledge can become very important and helpful to a client building their own application.

Consultation with the internal GCI Technology Services group is required for any client wishing to build an API or to have a custom interface created. The helps everyone engaged understand the requirements for the available options and align to the client’s true business needs, timeline, and budget.

Further questions should be directed to your manager or the GCI IT team.