**API Integrations**

The reason why we separate the users is the disparity in the approaches to implementing both software. Talking about API integration, there is no reason for a regular internet enjoyer to know anything about how exactly it works. The usage experience for this group is post-factum, meaning that all the possible interactions with API integrations need no additional skills or add-ons. The API is a ready-to-use product for the final consumer. All the technical aspects are done by the developers, and the systems work on the background level, so most processes are invisible to the website visitor.

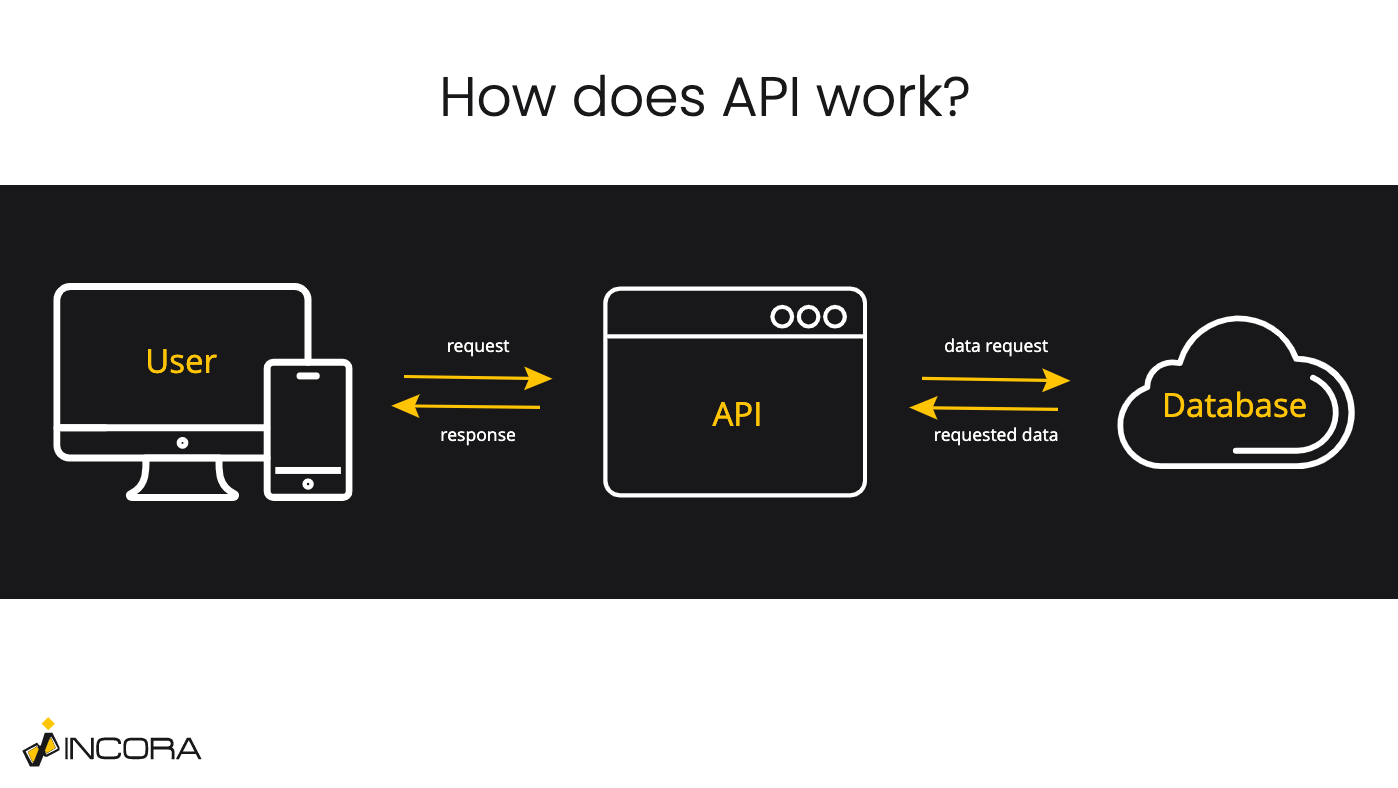
Actually, API stands for applicational processing interface. Mostly, it is not about creating a totally new product, but connecting existing software, which allows skipping a lot of different processes, especially the development. This connection makes it possible to use features and functions of third-party programs or systems by your product.

Usually, the API integrations are used to implement resource-intensive software. For example, developing services like Google Drive or Dropbox requires a lot of time and effort, but using cloud API allows exploiting existing products with all their functions in exchange for money. In other words, API integrations are used to buy the software bypassing the production procedure.

You may wonder [how exactly does API integration works](https://incora.software/insights/what-is-api-integration/94). The issue is that there is no one and only method which works in all cases. Each API has its own requirements for integration. Usually, these differences concern minor aspects but it is better to consider an example of such a procedure.

### How Does Google Drive API Integration Work?

Actually, there is no need to delve into the details, because all the needed information is already provided by Google on their website, or in the step-by-step instructions on [how to integrate different Cloud APIs](https://incora.software/insights/cloud-api-use-cases-for-google-drive-dropbox-onedrive-api/43), featured in our blog. Despite the fact, that there are only 5 points to be done, each of them has its required subprocesses as well, so the actual scheme is more complicated.



[In order to integrate Google Drive API into your product you need:](https://developers.google.com/workspace/guides/get-started)

1. To create a Google Cloud project for your product.
2. To choose the needed API (in our case it is Google Drive).
3. To implement the authentication and authorization.
4. To configure OAuth consent for the users.
5. To create access credentials.

Thanks to this list, we can conclude that to integrate any Google products into your website, you will need not only to pay for the particular API but also to use Google Workspace, configure all the needed alerts and notifications according to the required protocols, as well as to enable obligatory Google authentification and authorization processes at your website. Moreover, all these steps are mandatory. So, the API owner could stop providing the access to it in case of non-compliance with the agreements. So, despite the fact that you don’t have full control over the API integration, you are also limited by the agreements.

What is important to understand is that each API provider has its own terms and agreements to use, so the limitations and obligations may vary or even be absent.

### API Integrations Pros and Cons

#### Advantages

Summing all the above, we may say that the applicational processing interfaces are very different. This is why every API should be considered separately. Nevertheless, there are few statements, which apply to all the APIs.

First of all, API integrations usually offer a combination of various features and functions, that interact with each other. These implementations highly increase the possible functionality of the product, they are integrated into. APIs decrease the number of resources needed. Also using them allow bypassing the development at all. Mostly, applicational processing interfaces provide access to the software, which is hard and very consuming to develop. In addition, almost all the technical components, including z and bug fixing are covered by the provider.

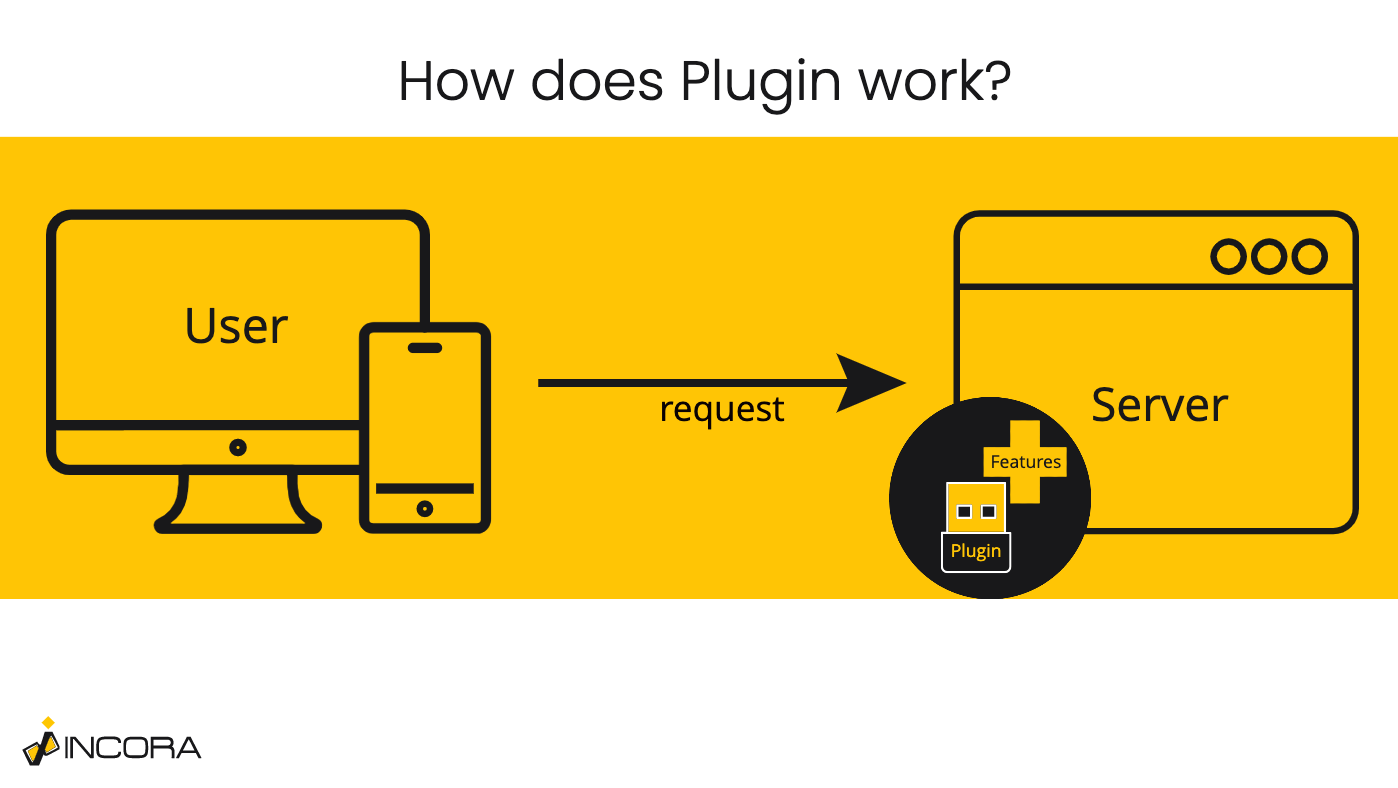
#### Drawbacks

On the other hand, it limits you in terms of usage, obligating you to follow the terms of use and make corrections if they are changed. Also, the variety of interfaces and the differences between them complicates the process of choosing the needed one making your product provider-dependent. It is important not to forget the fact that the API is under the full control of its owner. And of course, the main disadvantage of the API integrations is their cost, because most of them are paid-to-use. It is hard to tell even the approximate price, due to many factors that could influence it.

## What Is Plugin?

A plugin, also known as a plug-in is additional software, that helps to improve some basic aspects of the working processes of your application or website. Despite the fact, that both softwares are used to improve or add new features, there are a lot of differences between API and plugins.

First of all, plugins are easier to develop. Basically, it is not a full-scale software, but [its additional component](https://en.wikipedia.org/wiki/Plug-in_(computing)), which enables or improves specific features. In other words, it is a way of customization by using different add-ons. Also, it is just a component, so it limits the possible plugin functionality.



In fact, plug-ins are working more locally, so each of them usually provides a single feature. Nevertheless, the variety of possible plugin features is incredibly huge. But mostly, they are used as extensions to enable additional software. For example, the Unity game engine or Adobe Flash player plugins for web browsers were used to allow different types of multimedia.

Nowadays, web plugins were mostly replaced by browser extensions and are not supported anymore. It is very important not to confuse them, because, even though extension and plugin functionality are sometimes similar, they are still different. But it is fair to say that web plugins could be considered as ancestors of modern extensions.

Most of the plugin installation processes require additional software or apps. Even after they were deprecated by browsers, plugins are still used as extensions for different applications and still could be met on the Internet but not as standalone software. The best way to explain the key plugin functionality is to take a look at the specific cases.

### Adobe Flash Player

The Adobe Flash plugin installation process is as simple as any other. All you needed to do is to download the setup program from the original website and run it. Most of the processes were automatized, so the user didn’t have to have additional knowledge and programming background.

This was the main feature of such plug-ins because most of them simplified the user experience a lot. Also, the Adobe Flash player was used not only to play the multimedia such as audio or video files in the browser but also to create them.

Nevertheless, there were many issues with such standalone plugins. They were insecure and it was impossible to use multimedia without them. Most of the web plugin functionality was replaced by HTML5.

### Website Construction Plugins

Despite that standalone plugins are becoming history today; the extension ones are becoming more popular every day. Mainly, such plugins are used for website construction and creation. For example, WordPress uses them as an additional instrument.

Once again, they are user-friendly and plugin functionality provides a lot of possible customization elements. As a great bonus, WordPress plugins usually do not require any plugin installation processes, they are literally plugged into the software. WordPress offers a library of custom plugins developed by third-person companies or users and WP itself.

The benefit of such an approach is that most website constructors’ plugins are doing the developers’ job by changing the code, so the end-user has only to choose the needed plug-in and the software will do everything else.

On the other hand, the usage of such plugins is limited by the platform they were created for. Also, these tools are still vulnerable in many aspects, such as security.

## Final Decision: API Vs Plugin

Finally, after this slight acquaintance with both software, we can now compare them and answer the question: “What is the difference between API and plugins?”

But before answering it, let’s summarize the commons. First of all, both are used to improve or add new features by adding supplementary software components and they rather play the role of a retranslator, meaning that they create connections between other systems, etc. Their main audience are developers. On the other hand, most regular internet surfers have no idea what API exactly is, but have some guesses about plugins, even if they may be wrong. Nevertheless, the term “plugin” is familiar to most of them.

The distinction in recognition is not the major difference between API and plugins. Actually, this contrast is more than one particular element, it is rather a combination of various reasons, which impact each other. For example, all the working processes of API integrations are done on the backend, meaning that a regular user has no guess about them. It impacts the lack of end-user awareness about the API.

Moreover, comparing API vs plugins it is impossible to ignore the fact, that the scope of their functionality is very different: API is an interface that enables the usage of massive resource-intensive software with a variety of possible features; at the same time, the plugin is more locally directed, mostly providing or improving a single feature.

This is why plugins could not be used as full-fledged software or for specific functions. For instance, such things as the [payment gateway](https://incora.software/insights/how-to-deploy-a-payment-gateway-payment-api-practical-use/48), integration into management systems like [LPMS](https://incora.software/insights/api-integration-into-lpms/40) or [MPMS](https://incora.software/insights/list-of-features-for-medical-practice-management-software/89), as well as providing [eCommerce](https://incora.software/insights/ecommerce-api-in-action-woocommerce-shopify-setup-in-2021/50) support are possible thanks to API integrations. Plugins can be used as additional tools, but for embodying or optimization of such products.

Meanwhile, plugins have a great advantage for beginners or small businesses - it is much cheaper to use, compared to the API. Even though both of them have issues with security, it is unquestionable that API is safer to use anyway. Instead, the plugin installation process is easier and less demanding than the API integration process.