ESAT ESCUELA SUPERIOR DE ARTE Y TECNOLOGÍA

> **Programacion Grafica**
> Ivan Sancho
> Curso 2022/2023

Graphic Programming

AUTOR/ES: Jose Maria Maestre Quiles, Hector Ochando

1 What is an API?

An API (Application Programming Interface) is a set of communication protocols, tools and definitions that allow different applications or systems to interact with each other programmatically.

In other words, an API is a programming interface that allows developers to access certain functions or data of an application or system, without needing to know all the internal details of how it works. APIs are often used to integrate third-party services or applications into an existing platform, to automate tasks or to create new tools.

APIs can be of different types, such as REST, SOAP, GraphQL, among others, and can use different communication protocols, such as HTTP, TCP/IP, UDP, among others.

1.1 How to use an API?

To use an API, the following steps are generally required:

1. Obtain an API key: Many APIs require developers to obtain an access key in order to use their services. This key is unique to each developer and is used to authenticate requests to the API.

2. Read the API documentation: it is important to read the API documentation to learn the details of how the API works, including which endpoints are available, what parameters each request accepts, response formats and any other relevant information.

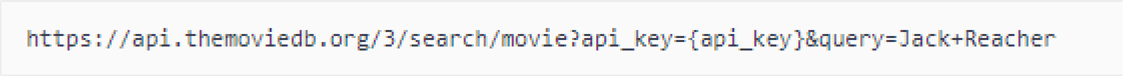
3. Make an API request: once you have an access key and have read the documentation, you can start using the API by making requests to your endpoints. This is usually done through HTTP calls using a programming language, such as Python, JavaScript, Ruby, etc.

4. Process the response: once the API request has been made, you will receive a response in the format specified in the documentation, usually in JSON format. This response will generally include the data requested by the developer, which can be processed and used in the application or system being worked on.

1.2 API example (Jose Maria Maestre)

For this example, I will use the REST movieDB API.

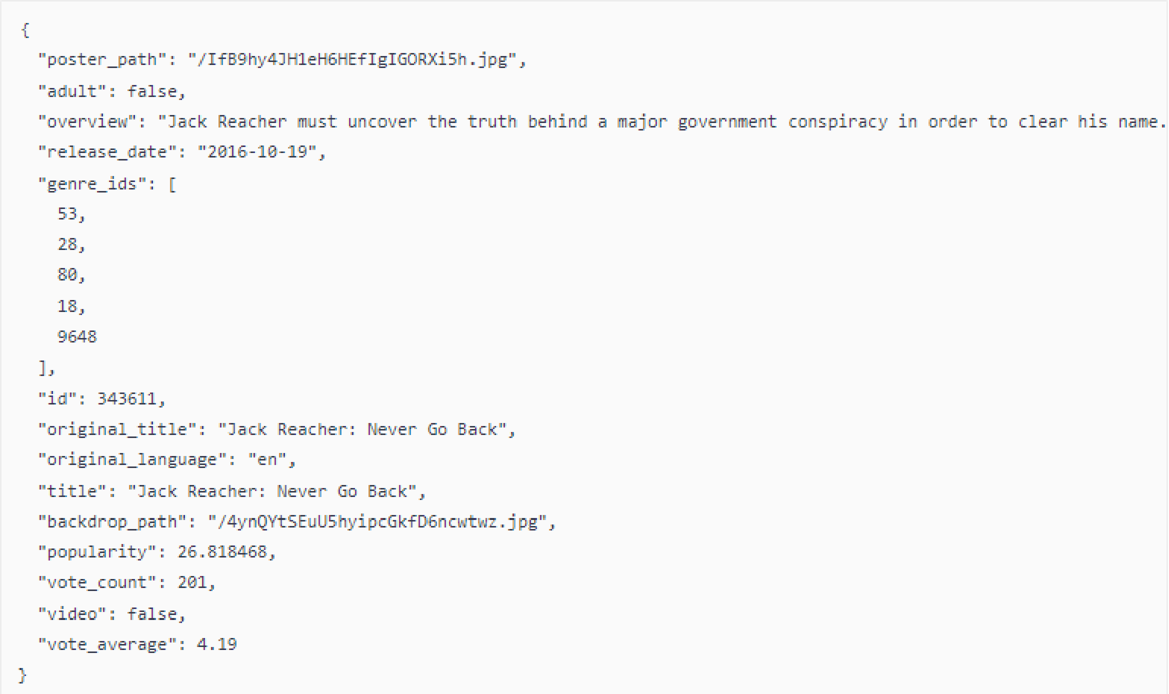
We will search for a movie with the title "Jack Reacher" with this https request



```
https://api.themoviedb.org/3/search/movie?api_key={api_key}&query=Jack+Reacher
```

In the {api_key} part we must replace it with the "key" provided by the organisation when registering.

This request will return all the relevant information about the movie "Jack Reacher" in JSON format.



```
{
  "poster_path": "/IfB9hy4JH1eH6HEfIgIGORXi5h.jpg",
  "adult": false,
  "overview": "Jack Reacher must uncover the truth behind a major government conspiracy in order to clear his name.
  "release_date": "2016-10-19",
  "genre_ids": [
    53,
    28,
    80,
    18,
    9648
  ],
  "id": 343611,
  "original_title": "Jack Reacher: Never Go Back",
  "original_language": "en",
  "title": "Jack Reacher: Never Go Back",
  "backdrop_path": "/4ynQYtSEuU5hyipcGkfD6ncwtwz.jpg",
  "popularity": 26.818468,
  "vote_count": 201,
  "video": false,
  "vote_average": 4.19
}
```

We can also search for films in a certain genre, featuring certain actors, directed by a certain director, etc.

1.3 API example (Hector Ochando)

The Unity API (Application Programming Interface) is a set of functions and procedures that developers can use to interact with the Unity game engine to create their own applications and games.

The Unity API is designed to be used with the C# programming language, but can also be used with other programming languages, such as JavaScript or Boo.

The Unity API provides a wide variety of functions and procedures that allow developers to perform common tasks in games and applications. These functions and procedures cover areas such as in-scene object creation, time management, user input, physics, animation, sound, networking and artificial intelligence.

In addition, Unity also provides a number of pre-built components that can be used in applications. These pre-built components are pre-built objects that include scripts and configurable properties that allow developers to quickly create common elements in games, such as characters, cameras, visual effects and other elements.

The Unity API also includes a number of additional tools and resources that allow developers to create high-quality games and applications. For example, Unity includes an advanced physics system that allows realistic simulation of objects and collisions, and an animation system that allows the creation of complex and realistic animations.

In short, the Unity API is a set of functions and procedures that allow developers to interact with the Unity game engine and create their own applications and games. The API covers a wide variety of areas and offers a number of additional tools and resources that allow developers to create high quality games and applications.

2 What is an SDK?

An SDK (Software Development Kit) is a set of software tools and resources that developers use to create applications for a specific platform. SDKs are a valuable resource for developers, enabling them to create applications more quickly and efficiently by providing tools and resources to simplify and accelerate the development process.

A typical SDK can include a variety of resources and tools, such as programming libraries, development tools, documentation, code samples, emulators, simulators and other resources.

The programming library provided by the SDK contains a number of functions and procedures that developers can use to interact with the system or platform for which the application is being developed.

The development tools included in an SDK can help developers create, debug and test applications more efficiently. These tools include code editors, debuggers, compilers, emulators, simulators and other resources that make developers' work easier.

The documentation provided by the SDK is very important as it helps developers understand how to use the tools, libraries and resources provided by the SDK. This documentation includes user manuals, programming guides, tutorials and other resources that explain how to use the SDK tools and resources.

The code samples provided by the SDK are very valuable to developers, as they allow them to see how the functions and procedures provided by the programming library can be used. These code samples are usually simple programs that demonstrate how to use different functions and procedures of the SDK.

Overall, an SDK is an essential tool for developers, as it allows them to create applications faster and more efficiently by providing useful tools and resources. By using an SDK, developers can save time and effort by having access to programming libraries, development tools, documentation, code samples and other resources that allow them to focus on creating applications rather than having to write all the code from scratch.

2.2 How to use an SDK?

The process for using an SDK can vary depending on the platform or technology for which you are developing an application, but here are the general steps you can follow:

Download and install the SDK: The first thing you need to do is download and install the SDK for the platform or technology you want to develop on. Detailed instructions on how to install and configure the SDK can usually be found on the SDK vendor's website.

Configure the development environment: Once you have installed the SDK, you need to configure the development environment that you will use to create your applications. The SDK usually includes development tools and compilers that can be used in different environments, such as Visual Studio or Eclipse.

Study the documentation: It is important to study the SDK documentation to understand how to use the tools and resources it provides. Documentation often includes user manuals, programming guides, tutorials and other resources that explain how to use the SDK.

Use the programming library: The SDK's programming library is a collection of functions and procedures that you can use to interact with the platform or technology for which you are developing the application. It is important to study the SDK documentation to understand how to use these functions and procedures.

Create sample code: To learn how to use the SDK and the programming library, it is advisable to create small sample programs that use the functions and procedures in the SDK. The code samples provided by the SDK can also help.

Compile and test your application: Once you have created your application using the SDK, you should compile and test it on the appropriate platform or technology. If the SDK includes debugging tools, use them to identify and fix any bugs that may arise.

In summary, using an SDK involves downloading and installing the SDK, setting up the development environment, studying the documentation, using the programming library, creating sample code, compiling and testing the application. By following these steps, developers can use the SDK to create applications faster and more efficiently.

2.3 SDK example (Hector Ochando)

Unreal is a Software Development Kit (SDK) that includes a game engine, the Unreal API, development tools, programming libraries and other components for creating applications and games.

The Unreal Engine is the core part of the Unreal SDK. It provides a set of tools and features for game and application development, such as high-quality graphics, advanced physics, animations, artificial intelligence, sound, networking and more. Developers can use the Unreal Engine to create game projects and applications for multiple platforms, including PCs, consoles and mobile devices.

The Unreal API is a set of functions and procedures that allow developers to interact with the Unreal Engine game engine. The API covers a wide variety of areas and provides a number of additional tools and resources that enable developers to create high quality games and applications.

In addition, the Unreal SDK also includes development tools such as the Unreal Editor, which is a visual tool for creating and editing game content and applications, and compilation and debugging tools for creating and testing projects.

In short, Unreal is a complete SDK that includes the Unreal Engine game engine, the Unreal API, development tools, programming libraries and other components for creating high-quality games and applications. Developers can use the Unreal SDK to create game projects and applications for multiple platforms and take advantage of the tools and resources provided by the SDK.

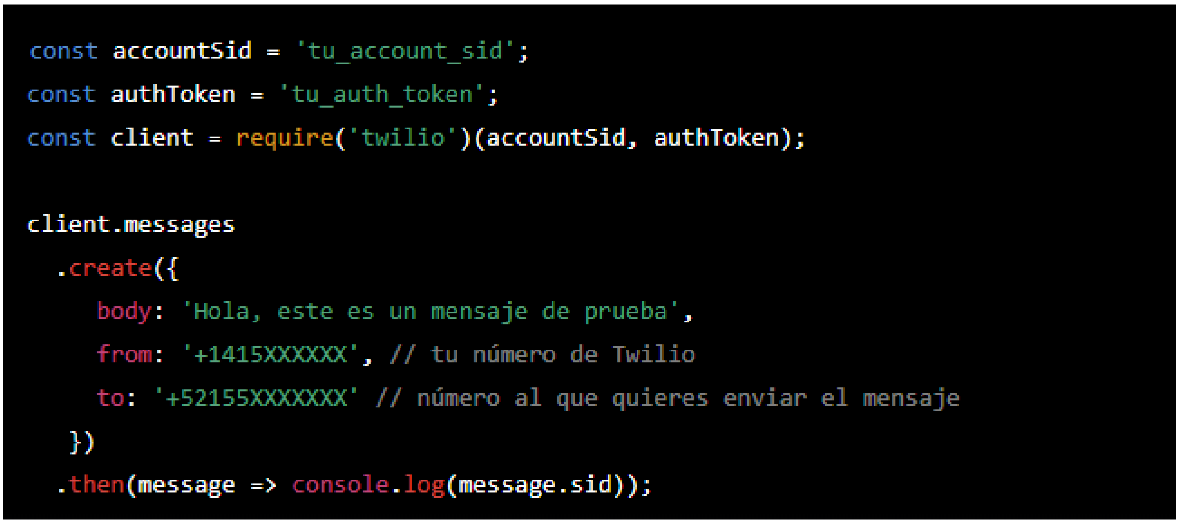
2.4 SDK example (Jose Maria Maestre)

Twilio is an SDK that allows you to create a messaging and calling application using its internal API.

Twilio offers a wide variety of programming languages to work with, such as JavaScript, Python, Ruby, PHP, among others, making it easy to use regardless of the developer's preferred programming language.

In addition, the Twilio SDK offers comprehensive documentation, including code samples for each of the supported programming languages. This allows developers to start working with the Twilio API quickly and without having to spend a lot of time researching.

For example, in JavaScript, the following code could be used to send a text message with Twilio:



```
const accountSid = 'tu_account_sid';
const authToken = 'tu_auth_token';
const client = require('twilio')(accountSid, authToken);

client.messages
  .create({
     body: 'Hola, este es un mensaje de prueba',
     from: '+1415XXXXXX', // tu número de Twilio
     to: '+52155XXXXXXX' // número al que quieres enviar el mensaje
   })
  .then(message => console.log(message.sid));
```

Twilio's SDK allows you to integrate its API in different programming languages.

3. PostMortem

Overall, we are pleased with the work done in the course, as we have successfully implemented all the geometries, materials and lights. However, we are not completely satisfied with the visual presentation, as we consider that it does not do justice to all the effort invested.

The main idea was to represent a terrain with moving water, small islands scattered around the map and an illuminated lighthouse. We succeeded in implementing our vision, although it did not turn out as well as we had imagined.

We emphasised the strength of our project in the water, which we were particularly proud of due to its realistic appearance and the reflections of the sun in relation to the directional light.

We encountered difficulties when trying to incorporate light into the objects we carried.

If we were to start the project again, we would organise the start of the scene better, separating each element into its own function and then calling them from the "init" method. This would allow us to have a neater and clearer structure.

We encountered problems when assigning the focal light to the ball rotating around the lighthouse, as we had to build the model matrix of the ball to get its position in the world, given that it rotates around an object.

The object loading and the interaction of the light with our geometries has been excellent. However, we have not been able to get the same results with the objects themselves.

With more time, we would like to implement the use of LUA to be able to move objects in the scene by clicking on them, instead of relying on the ImGui window.

The APIS used are the following:

- **OpenGL**: This API is the fundamental base on which our project is based. It is a graphical interface that has given us the ability to render all the objects in our scene. Although it took us some time to become familiar with it, as we progressed in its development, its use became more fluid and understandable. Overall, we consider OpenGL to be an extremely powerful tool and an ideal choice for those wishing to get into the graphical realm of programming.

- **ImGui**: We show the value of important variables such as position, rotation and scaling of an object and the possibility to modify these values at runtime. This not only serves to give the user the possibility to move the objects as they wish, but also to adjust each object to the position we want. We have not had much difficulty in implementing it because we had already used it previously in other projects.

- **GLFW**: We use the GLFW library for the implementation of PS4 and XBOX controllers. Its integration has been quite simple, because once the controller is connected you only have to collect the input you receive and separate it by each of the buttons to process it properly. We have had to test buttons for the XBOX controller because it doesn't receive the input in the same way as the PS4 controller.

- **STB Image**: We use this library to load images, especially those used in the cube map. It's a simple API and it didn't take us too much work to adapt to it and include it in the project.

- **Perlin Samplex Noise**: Thanks to this API, we have been able to generate both the terrain on the water and the water itself to form small or large irregularities in the various terrains to give it a more realistic aesthetic. Its integration has not been very costly as it was already integrated in the engine and our teacher taught us how to use it.

- **GLEW (OpenGL Extension Wrangler Library)**: By using GLEW, we have had access to a wide range of OpenGL extensions, which has allowed us to take full advantage of the advanced functionalities that this library offers. The integration of the GPU manager has been an essential part of our project, allowing us to manage and optimise graphics processing efficiently.

  The use of this API has not been without its challenges, as we have encountered multiple approaches and options to accomplish our tasks. However, it has not been overly difficult or complex either.

- **ESAT**: We have used the ESAT library for mathematical functions, in particular for the use of the time elapsed between frames. Its implementation has been simple as it is included in the engine and we have been using it for a long time in previous courses and in this one.

- **OXML**: We have used our own library for mathematical functions, such as vectors and matrices. Its implementation has been simple as it is our own API and we know perfectly well how it works.

4. BIbliography

Twilio [online information] twilio.com/es-mx
[consulted 02/06/2023]. Available on: https://www.twilio.com/es-mx

Chat GPT[online information] openai.com/blog/chatgpt [consulted 02/06/2023].
Available on: https://openai.com/blog/chatgpt