


## Configuración de API Meta/developer/

Se configura la API para poder acceder a los tokens y Url para ser consumida, en nuestro formulario.



**PortafolioWeb2**  
Identificador de la aplicación: 11827642638495  
Modo: En desarrollo  
Tipo: Empresa  
Empresa: Laxy games

Administrador

## Aquí es donde se configuran los accesos para poder consumir el API

**Identificador de acceso**

Copiar Generar identificador de acceso

**Enviar y recibir mensajes**

**Paso 1: Selecciona números de teléfono**

**Desde**  
Los números de teléfono de prueba te permiten enviar mensajes de forma gratuita durante 90 días. Puedes usar tu propio número de teléfono, sujeto a límites y precios. [Información sobre los precios](#)

Número de prueba: +1 555 176 [redacted]

Identificador del número de teléfono: 525591510641821 Copiar    Identificador de la cuenta de WhatsApp Business: 562977493561352 Copiar

**A**

+57 317 [redacted]

**Paso 2: Envía mensajes con la API**

Para enviar un mensaje de prueba, copia este comando, pégalo en Terminal y pulsa Intro. Para crear tu propia plantilla de mensajes, haz clic [aquí](#).

```
1 curl -i -X POST `
2 https://graph.facebook.com/v21.0/[redacted]/messages `
3 -H 'Authorization: Bearer <access token>' `
4 -H 'Content-Type: application/json' `
5 -d '{ "messaging_product": "whatsapp", "to": "57317[redacted]", "type": "template", "template": { "name":
"hello_world", "language": { "code": "en_US" } } }'
```

Ejecutar en Postman Enviar mensaje

**Identificador de acceso**

Copiar Generar Identificador de acceso

**Enviar y recibir mensajes**

**Paso 1: Selecciona números de teléfono**

**Desde**  
Los números de teléfono de prueba te permiten enviar mensajes de forma gratuita durante 90 días. Puedes usar tu propio número de teléfono, sujeto a límites y precios. [Información sobre los precios](#)

Número de prueba: +1 555 176 [redacted]

Identificador del número de teléfono: 525591510641821 Copiar    Identificador de la cuenta de WhatsApp Business: 562977493561352 Copiar

**A**

+57 317 [redacted]

**Paso 2: Envía mensajes con la API**

Para enviar un mensaje de prueba, copia este comando, pégalo en Terminal y pulsa Intro. Para crear tu propia plantilla de mensajes, haz clic [aquí](#).

```
1 curl -i -X POST `
2 https://graph.facebook.com/v21.0/[redacted]/messages `
3 -H 'Authorization: Bearer <access token>' `
4 -H 'Content-Type: application/json' `
5 -d '{ "messaging_product": "whatsapp", "to": "57317[redacted]", "type": "template", "template": { "name":
  "hello_world", "language": { "code": "en_US" } } }'
```

Ejecutar en Postman Enviar mensaje

**Formulario Principal, se encargara de recibir la Data que será Procesada.**

CONTACTO  
**CONTACTAME**

Esta es una Prueba e la API para la Documentación|

**Envío del Formulario, se envía el Formulario y procede hacer procesado por JavaScript**

# CONTACTO

## CONTACTAME

Tu mensaje fue enviado correctamente. ¡Gracias por contactarnos! ×

Tu nombre

Tu correo

Telefono / WhatsApp

Lo que me quieras decir

### **Proceso de Datos con JavaScript.**

El código valida un formulario de contacto con jqBootstrapValidation, envía los datos vía AJAX a mail/contact.php, y muestra mensajes de éxito o error según la respuesta del servidor. Deshabilita temporalmente el botón de envío para evitar duplicados, gestiona errores de conexión y permite cambiar de pestañas con clics. También limpia mensajes al enfocar el campo de nombre.

```
man > contact.js > @ 30 callback > @ submitSuccess
You, 2 days ago | 1 author (You)
1 $(function () {
2     $("#contactForm input, #contactForm textarea").jqBootstrapValidation({
3         preventSubmit: true,
4         submitError: function ($form, event, errors) {
5             // Manejo de errores
6         },
7         submitSuccess: function ($form, event) {
8             event.preventDefault(); // Evita el envío estándar del formulario
9             var name = $("#inputName").val();
10            var email = $("#inputEmail").val();
11            var subject = $("#inputSubject").val();
12            var message = $("#textareaMessage").val();
13
14            var $this = $("#sendMessageButton");
15            $this.prop("disabled", true);
16
17            // Realiza la llamada AJAX al servidor PHP
18            $.ajax({
19                url: "mail/contact.php", // Ruta al archivo PHP
20                type: "POST",
21                dataType: "json", // Esperamos respuesta en formato JSON
22                data: {
23                    name: name,
24                    email: email,
25                    subject: subject,
26                    message: message,
27                },
28                success: function (response) {
29                    if (response.status === "success") {
30                        $("#success").html("<div class='alert alert-success'>");
31                        $("#success > .alert-success")
32                            .html("<button type='button' class='close' data-dismiss='alert' aria-hidden='true'>&times;</button>")
33                            .append("<strong> " + response.message + "</strong>")
34                            .append("</div>");
35                        $("#contactForm").trigger("reset");
36                    } else {
37                        $("#success").html("<div class='alert alert-danger'>");
38                        $("#success > .alert-danger")
39                            .html("<button type='button' class='close' data-dismiss='alert' aria-hidden='true'>&times;</button>")
40                            .append("<strong> " + response.message + "</strong>")
41                            .append("</div>");
42                    }
43                },
44                error: function () {
45                    $("#success").html("<div class='alert alert-danger'>");
46                    $("#success > .alert-danger")
47                        .html("<button type='button' class='close' data-dismiss='alert' aria-hidden='true'>&times;</button>")
48                        .append("<strong>Hubo un error al intentar enviar el mensaje. Por favor intenta nuevamente.</strong>")
49                        .append("</div>");
50                },
51                complete: function () {
52                    $this.prop("disabled", false);
53                    console.log("aquí leigo...");
54                },
55            });
56        },
57        filter: function () {
58            return $(this).is(":visible");
59        },
60    });
61
62    $("#a[data-toggle='tab']").click(function (e) {
63        e.preventDefault();
64        $(this).tab("show");
65    });
66
67    $("#name").focus(function () {
68        $("#success").html("");
69    });
70 });
```

## Manejo de API Backend

Este código PHP recibe datos de un formulario, los valida y sanitiza, y luego envía un mensaje de WhatsApp a través de la API de Facebook utilizando cURL. Responde en formato JSON con mensajes de éxito o error según el resultado de la solicitud.

```

<?php
header(header: 'Content-Type: application/json');

// Validar los datos enviados
if (empty($_POST['name']) || empty($_POST['subject']) || empty($_POST['message']) || !filter_var(value: $_POST['email'], filter: FILTER_VALIDATE_EMAIL)) {
    http_response_code(response_code: 400); // Código 400: Solicitud incorrecta
    echo json_encode(value: ['status' => 'error', 'message' => 'Datos incompletos o inválidos.']);
    exit();
}

// Sanitizar los datos recibidos
$name = strip_tags(string: htmlspecialchars(string: $_POST['name']));
$email = strip_tags(string: htmlspecialchars(string: $_POST['email']));
$subject = strip_tags(string: htmlspecialchars(string: $_POST['subject']));
$message = strip_tags(string: htmlspecialchars(string: $_POST['message']));

// Preparar el mensaje para WhatsApp
$message = "Hola Larry! " . $name . " quiere hablar contigo. Su correo es " . $email . ", su número de WhatsApp es " . $subject . ", y su mensaje es: " . $message . "";

// Configuración de la API de Facebook para enviar mensajes de WhatsApp
$token = "EAJQz8T8W...";
$telefono = "57314470828";
$url = "https://graph.facebook.com/v21.0/525591510C.../messages";

// Configuración del cuerpo del mensaje
$message = json_encode(value: [
    'messaging_product' => 'whatsapp',
    'to' => $telefono,
    'type' => 'text',
    'text' => [
        'body' => $message
    ]
]);

// Cabeceras para la solicitud HTTP
$header = [
    'Authorization: Bearer ' . $token,
    'Content-Type: application/json',
];

// Inicializar curl
$curl = curl_init();
curl_setopt(handle: $curl, option: CURLOPT_URL, value: $url);
curl_setopt(handle: $curl, option: CURLOPT_POSTFIELDS, value: $message);
curl_setopt(handle: $curl, option: CURLOPT_HTTPHEADER, value: $header);
curl_setopt(handle: $curl, option: CURLOPT_RETURNTRANSFER, value: true);

// Ejecutar la solicitud y obtener la respuesta
$response = curl_exec(handle: $curl);
$status_code = curl_getinfo(handle: $curl, option: CURLINFO_HTTP_CODE);

// Manejar errores en la solicitud curl
if (curl_errno(handle: $curl)) {
    http_response_code(response_code: 500);
    echo json_encode(value: [
        'status' => 'error',
        'message' => 'Error al intentar enviar el mensaje. Detalles: ' . curl_error(handle: $curl)
    ]);
    curl_close(handle: $curl);
    exit();
}

// Cerrar curl
curl_close(handle: $curl);

// Analizar la respuesta de la API
$responseData = json_decode(json: $response, associative: true);

// Manejar la respuesta según el código de estado
if ($status_code >= 200 && $status_code < 300) {
} else {

```

## Resultado de API

