

Log de Errores en SLACK

U-TAD

Monitorización: enviar errores a Slack

<https://slack.com/> (abre una cuenta si no la tienes ya)

Crea un nuevo espacio de trabajo y dentro de Slack un Canal cerrado “logs”

Una vez creada la cuenta y el espacio de trabajo accedemos a:

<https://api.slack.com/messaging/webhooks>

En 1.- Pulsa el botón **Create your Slack app** (From Scratch)

Name app & choose workspace

App Name

[BOT] Web Server Monitoring

Don't worry - you'll be able to change this later.

Pick a workspace to develop your app in:

U-TAD Web Server

Keep in mind that you can't change this app's workspace later. If you leave the workspace, you won't be able to manage any apps you've built for it. The workspace will control the app even if you leave the workspace.

[Sign into a different workspace](#)

By creating a Web API Application, you agree to the [Slack API Terms of Service](#).

Cancel Create App

Incoming Webhooks
Post messages from external sources into Slack.

Interactive Components
Add components like buttons and select menus to your app's interface, and create an interactive experience for users.

Slash Commands
Allow users to perform app actions by typing commands in Slack.

Event Subscriptions
Make it easy for your app to respond to activity in Slack.

Bots
Allow users to interact with your app through channels and conversations.

Permissions
Configure permissions to allow your app to interact with the Slack API.

Activate Incoming Webhooks ☒

Add New Webhook to Workspace

logs

Webhook URL Channel Added By

<https://hooks.slack.com/services/T04SW...> Copy #logs rpmya Mar 7, 2023

Pegar la URL en .env como SLACK_WEBHOOK

Monitorización: enviar errores a Slack

`npm i morgan-body` (<https://www.npmjs.com/package/morgan-body>)

`npm i @slack/webhook` (Sniffer de lo que pasa en nuestro Server y envío a Slack respectivamente).

Modificamos **app.js** (después lo abstraeremos a distintos handlers):

```
const morganBody = require("morgan-body")
const {IncomingWebhook} = require("@slack/webhook")
...

const webHook = new IncomingWebhook(process.env.SLACK_WEBHOOK)

const loggerStream = {

  write: message => {
    webHook.send({
      text: message
    })
  },
}

morganBody(app, {
  noColors: true, //limpiamos el String de datos lo máximo posible antes de mandarlo a Slack
  skip: function(req, res) { //Solo enviamos errores (4XX de cliente y 5XX de servidor)
    return res.statusCode < 400
  },
  stream: loggerStream
})
```

Monitorización: enviar errores a Slack

Nos llevamos el objeto `loggerStream` desde `app.js` a **`utils/handleLogger.js`** (creadlo)

```
const { IncomingWebhook } = require("@slack/webhook")
const webHook = new IncomingWebhook(process.env.SLACK_WEBHOOK)
const loggerStream = {
  write: message => {
    webHook.send({
      text: message
    })
  }
}
module.exports = loggerStream
```

Y en `app.js`:

```
const express = require("express")
const cors = require("cors")
const morganBody = require("morgan-body")
require("dotenv").config();
const loggerStream = require("../utils/handleLogger")
const dbConnect = require("../config/mongo")
...
morganBody(app, {
  noColors: true,
  skip: function(req, res) {
    return res.statusCode < 400
  },
  stream: loggerStream
})
```