APIs

Stop clicking buttons like a mortal.

Jayrup Nakawala

What is an API?

API = Application Programming Interface

For example, the weather bureau’s software system contains daily weather data. The weather app on your phone “talks” to this system via APIs and shows you daily weather updates on your phone.

Basically a set of rules for how two programs talk to each other.

How does a browser work?

Your browser is just a UI wrapper.

Every button you click is just a text message sent to a server.

If you can send that message yourself, you don’t need the browser.

The Anatomy of a Request

HTTP is just Text

It’s not magic. It’s a text file sent to port 443.

```
POST /api/rewards/claim HTTP/1.1
Host: three.co.uk
Authorization: Bearer abc123xyz
Content-Type: application/json

{
  "reward_id": 5599
}
```

If you send this text, the server thinks you are the browser.

The Verbs (Intent)

- **GET**: “Give me data” (Read).
- **POST**: “Take this data” (Write/Action).
- **PUT/PATCH**: “Update this” (Modify).
- **DELETE**: “Destroy this” (Dangerous).

The Headers (The Metadata)

- **Content-Type**: “I am speaking JSON.”
- **User-Agent**: “I am Chrome” (The lie).
- **Authorization**: “I am Jayrup.”

The **Authorization** header is your digital passport. If I have your token, I *am* you.

The Tools

We are using:

- **DevTools**: To spy on the browser.
- **cURL**: To send requests.
- **jq**: To clean up the mess.

Case Study 1: Free rewards

The Scenario

I have a Three sim card. Every week there are free reward. I kept forgetting to claim it.

I reversed engineered it.

The “Copy as cURL” Trick

1. Open DevTools (F12) -> Network Tab.
2. Click the button in the UI.
3. Right click the request -> **Copy as cURL**.

Paste it into your terminal.

The Cleanup

The browser sends too much garbage.

- `User-Agent`? Delete.
- `Referer`? Delete.
- `Cookies`? Keep only what’s needed.

Result:

```
curl -X POST "[https://simcompanies.com/api/v2/claim](https://simcompanies.com/
     -H "Authorization: Bearer my_token"
```

Status Codes (The Attitude)

- **200**: Success.
- **300**: Go look over there (Redirect).
- **400**: You messed up (Client Error).
- **500**: I messed up (Server Error).

> ⚠ **Warning**
>
> **401 vs 403** 401: Who are you? (Login failed) 403: I know you, but no. (Permission denied)

Advanced: Moodle

Moodle is Old

Modern APIs are REST (`/users/1`). Moodle is RPC (`server.php?function=get_user`).

It’s ugly. It returns HTML inside JSON. But we can tame it.

The Goal

“I want my deadlines in my terminal.”

We need:

1. The Mobile App Token (The Key).
2. The `core_calendar` function.
3. `jq` to strip the HTML tags.

The Script

```
TOKEN="xxxxxxxx" # Don't share this
HOST="[https://moodle.uel.ac.uk](https://moodle.uel.ac.uk)"

curl -s "$HOST/webservice/rest/server.php" \
  --data-urlencode "wstoken=$TOKEN" \
  --data-urlencode "wsfunction=core_calendar_get_action_events_by_courses" \
  --data-urlencode "moodlewsrestformat=json" \
  | jq -r '.groupedbycourse[].events[] | "\(.name) :: \(.formattedtime | sub("<
```

Do I know what this does?

Ofc Not.

The Result

Instead of logging into Moodle:

```
Module Name :: Tuesday, 14 Nov, 11:59 PM
Module Name :: Friday, 17 Nov, 5:00 PM
```

Automated. Clean. Fast.

Summary

1. The Browser is a Lie.
2. The Network Tab is the Truth.
3. `Copy as cURL` is your best friend.
4. If it has a button, it has an API.

Questions?

Go home and break something.