

Here is the explanation of CORS and the "Preflight" mechanism, using the context of your note.

1. What is CORS? (The Security Guard)

Browsers have a strict security rule called the **Same-Origin Policy**.

- **The Rule:** If you are on google.com, the browser trusts you to talk to google.com. But if you try to send data to amazon.com (a "Cross-Origin"), the browser panics. It thinks a malicious script might be trying to steal data.
- **CORS:** This is the "ID Badge" system that allows exceptions. It stands for **Cross-Origin Resource Sharing**. It is the server's way of telling the browser: *"It's okay, I know this guy. Let him in."*

2. What is a "Preflight"? (The "May I?" Check)

Not all requests are treated equally.

- **Simple Requests:** If you send basic data (like a standard HTML form), the browser sends it immediately. It assumes the server can handle it safely.
- **Complex Requests:** If you do something "fancy"—like setting the format to JSON (application/json) or adding custom headers—the browser gets suspicious.

Before sending your **actual** data (the POST), the browser sends a hidden, empty test message called an **OPTIONS request**. This is the **Preflight**.

Browser: "Hey Server, I have a user trying to send application/json. Is that allowed?"

Server: "Yes, that is allowed."

Browser: "Okay, sending the real data now."

3. Why removing the header "bypasses" it

Your note suggests removing headers: { 'Content-Type': 'application/json' }.

- **With the Header:** You are explicitly telling the server "This is JSON." The browser sees this as a **Complex Request** and forces a Preflight check. If your API Gateway isn't configured to answer that check (the OPTIONS method), the request fails immediately.
- **Without the Header:** The browser defaults to text/plain or standard form data. This is considered a **Simple Request**. The browser **skips the Preflight** and sends the data straight to the Lambda.

Summary: The "Professional" Way vs. The "Hack"

Method	What happens?	Pros/Cons
--------	---------------	-----------

Removing the Header	You trick the browser into thinking it's a "Simple" text request. The browser blindly sends the data without asking permission first.	<p>Pros: It works instantly without extra server config.</p> <p>Cons: Your Lambda receives "text" instead of "JSON", which might break parsing logic. It is less secure.</p>
The Professional Way	You keep application/json and configure API Gateway to answer the Preflight (OPTIONS) request with "Yes, come on in."	<p>Pros: Robust, secure, and follows modern API standards.</p> <p>Cons: Requires extra setup in Terraform (adding the OPTIONS method).</p>

Would you like me to generate the Terraform code to add that OPTIONS method so you can solve this the professional way?