**✅ What We'll Do:**

* Use Vercel to host a simple Node.js API endpoint.
* This endpoint will forward requests to Hugging Face (with your API key).
* Your frontend will call **your Vercel function**, which has no CORS issues.

**🛠 Step 1: Set Up a Vercel Account**

1. Go to <https://vercel.com>
2. Sign up with GitHub.
3. Create a **new project** and link it to a GitHub repo (you can make a new empty one).

**📁 Step 2: Create a Project Structure Locally**

Create a new folder and add the following files:

pgsql

CopyEdit

my-hf-proxy/

├── api/

│ └── hf.js

├── package.json

└── vercel.json

**📦 Step 3: package.json**

json

CopyEdit

{

"name": "hf-proxy",

"version": "1.0.0",

"main": "index.js",

"type": "module"

}

**🧠 Step 4: Create the API Proxy – api/hf.js**

This is your serverless function. It handles requests and forwards them to Hugging Face.

js

CopyEdit

export default async function handler(req, res) {

if (req.method !== 'POST') {

return res.status(405).json({ message: 'Only POST allowed' });

}

const input = req.body.inputs;

try {

const hfRes = await fetch("https://api-inference.huggingface.co/models/gpt2", {

method: "POST",

headers: {

Authorization: `Bearer ${process.env.HUGGINGFACE\_API\_KEY}`,

"Content-Type": "application/json",

},

body: JSON.stringify({ inputs: input }),

});

const data = await hfRes.json();

res.setHeader("Access-Control-Allow-Origin", "\*");

res.status(200).json(data);

} catch (err) {

res.status(500).json({ error: err.toString() });

}

}

**🧪 Step 5: Add vercel.json**

This tells Vercel that you want to use serverless functions:

json

CopyEdit

{

"version": 2,

"builds": [

{ "src": "api/hf.js", "use": "@vercel/node" }

]

}

**🔐 Step 6: Add Your Hugging Face API Key to Vercel**

1. Go to your Vercel dashboard.
2. Click your project → Settings → Environment Variables.
3. Add a variable:
   * **Key**: HUGGINGFACE\_API\_KEY
   * **Value**: Your Hugging Face key (starts with hf\_...)
4. Redeploy your project after setting the variable.

**🚀 Step 7: Deploy**

1. Push this code to GitHub (if you haven’t already).
2. Vercel will auto-deploy it and give you a URL like:

arduino

CopyEdit

https://your-project-name.vercel.app/api/hf

**🌐 Step 8: Use It in Your Frontend**

In your jeopardy.js, replace the Hugging Face URL with your Vercel one:

js

CopyEdit

const response = await fetch("https://your-project-name.vercel.app/api/hf", {

method: "POST",

headers: {

"Content-Type": "application/json",

},

body: JSON.stringify({

inputs: "Your question here",

}),

});

const data = await response.json();

Boom 💥 — no more CORS issues!

**💡 Bonus: Handle Preflight OPTIONS (if needed)**

If you're sending anything fancy (like Authorization headers from frontend), you might also want to support OPTIONS requests. Let me know if you want to add that!

Want me to help you build and test this whole thing out live?