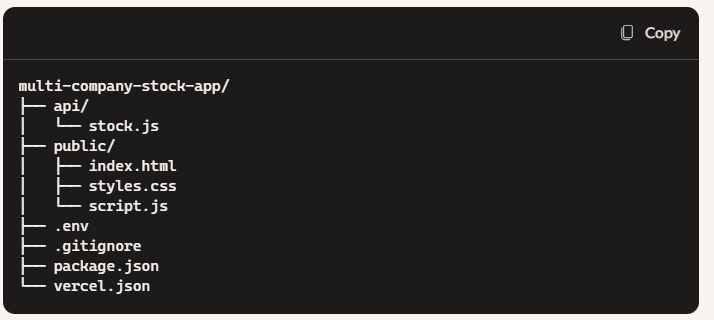
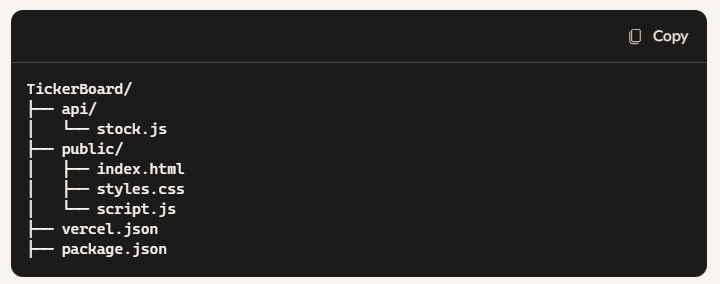
TickerBoard Project



Multi-company-stock-app is actually **TickerBoard in this directory**

****

**🧠 2. Convert server.js into a Serverless Function**

Inside a new /api/stock.js:

js

const { restClient } = require('@polygon.io/client-js');

const rest = restClient(process.env.POLYGON\_API\_KEY);

export default async function handler(req, res) {

const ticker = req.query.ticker?.toUpperCase() || 'AAPL';

try {

const data = await rest.stocks.aggregates(

ticker,

1,

'day',

'2025-01-01',

'2025-05-11'

);

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

} catch (error) {

res.status(500).json({ error: 'An error occurred: ' + error.message });

}

}

> This replaces the old Express server—you no longer need express, dotenv, or app.listen.

**📃 3. Create vercel.json Configuration**

This file is important to tell Vercel how to treat your serverless function:

json

{

"version": 2,

"builds": [{ "src": "api/\*.js", "use": "@vercel/node" }],

"routes": [

{

"src": "/api/stock",

"dest": "/api/stock.js"

},

{

"src": "/(.\*)",

"dest": "/public/$1"

}

]

}

**🧪 4. Update script.js to Hit the Right Endpoint**

Your fetch should now look like this:

js

const response = await fetch(`/api/stock?ticker=${ticker}`);

Make sure you're using ?ticker=TSLA instead of path params like /api/stock/TSLA.

**🔐 5. Set Environment Variable in Vercel**

Go to Vercel Dashboard:

* Project > Settings > Environment Variables:
  + POLYGON\_API\_KEY = your key

No need to use dotenv in Vercel—it loads the env vars automatically.

**🧹 6. Cleanup package.json**

Update dependencies to only include what’s required:

json

{

"name": "multi-company-stock-app",

"version": "1.0.0",

"dependencies": {

"@polygon.io/client-js": "^7.3.2"

}

}

Remove express, dotenv, and other unused packages.

**🚀 7. Deploy to Vercel**

If you haven't already:

bash

npm install -g vercel

vercel login

vercel --prod

> Vercel will read vercel.json, deploy the public assets, and configure the API function properly.

Would you like me to help validate this locally with vercel dev, or would you prefer to push to GitHub and have a CI/CD pipeline handle it? Either way, your stock app is about to go fully serverless!

Give this project a new branded name that suits it best.

How about **TickerTrail**?

It evokes the idea of following stock trends across multiple companies—precise, professional, and a little adventurous. Other name ideas with flair:

* **StockScope** – emphasizes broad visibility across the market.
* **MarketMesh** – highlights the interconnection of companies and data.
* **TickerBoard** – sounds like a digital dashboard for tickers.
* **EquiTrack** – clean, modern, and suggests equity tracking.

If you want to give it a more regional or educational flavor—especially since you're building tools for learning—we could explore that angle too. Want to theme it around insights, learning, or automation?

TickerBoard struck more to me.

Great choice, Jagdev—**TickerBoard** has a crisp, professional ring to it and instantly conjures the image of a dynamic dashboard for tracking stocks. It fits perfectly with your app’s utility and polish.

Want a logo concept or splash screen style to match the brand vibe? I could even help define a short tagline—something like *"Your Market. Your Momentum."* Let’s give TickerBoard the full upgrade it deserves!

Console message after commands executed

(base) PS C:\Users\jagdevsinghdosanjh.RHINO\TickerBoard> npm install -g vercel

>> vercel login

>> vercel --prod

>>

npm warn deprecated path-match@1.2.4: This package is archived and no longer maintained. For support, visit https://github.com/expressjs/express/discussions

added 19 packages, and changed 235 packages in 21s

26 packages are looking for funding

run `npm fund` for details

Vercel CLI 44.2.7

? Log in to Vercel Cancel

Vercel CLI 44.2.7

? Set up and deploy “~\TickerBoard”? yes

? Which scope should contain your project? Jagdev Singh Dosanjh's projects

? Link to existing project? no

? What’s your project’s name? ticker-board

? In which directory is your code located? ./

🔗 Linked to jagdev-singh-dosanjhs-projects/ticker-board (created .vercel and added it to .gitignore)

🔍 Inspect: https://vercel.com/jagdev-singh-dosanjhs-projects/ticker-board/CA3ZRW7rzZEuf8EQQbudriEymUd9 [3s]

✅ Production: https://ticker-board-frqwn7lel-jagdev-singh-dosanjhs-projects.vercel.app [3s]

2025-06-30T14:27:14.908Z Running build in Washington, D.C., USA (East) – iad1

2025-06-30T14:27:14.908Z Build machine configuration: 2 cores, 8 GB

2025-06-30T14:27:14.922Z Retrieving list of deployment files...

2025-06-30T14:27:15.136Z Previous build caches not available

2025-06-30T14:27:15.339Z Downloading 11 deployment files...

2025-06-30T14:27:15.698Z Running "vercel build"

2025-06-30T14:27:16.533Z Vercel CLI 43.3.0

2025-06-30T14:27:16.684Z WARN! Due to `builds` existing in your configuration file, the Build and Development Settings defined in your Project Settings will not apply. Learn More: https://vercel.link/unused-build-settings

2025-06-30T14:27:17.140Z Installing dependencies...

2025-06-30T14:27:19.853Z npm warn deprecated yaeti@0.0.6: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.

2025-06-30T14:27:20.388Z

2025-06-30T14:27:20.388Z added 29 packages in 3s

2025-06-30T14:27:20.389Z

2025-06-30T14:27:20.389Z 1 package is looking for funding

2025-06-30T14:27:20.389Z run `npm fund` for details

2025-06-30T14:27:20.886Z Warning: Node.js functions are compiled from ESM to CommonJS. If this is not intended, add "type": "module" to your package.json file.

2025-06-30T14:27:20.886Z Compiling "stock.js" from ESM to CommonJS...

2025-06-30T14:27:20.972Z Build Completed in /vercel/output [4s]

2025-06-30T14:27:21.011Z Deploying outputs...

❗️ Due to `builds` existing in your configuration file, the Build and Development Settings defined in your Project Settings will not apply. Learn More: https://vercel.link/unused-build-setttings

(base) PS C:\Users\jagdevsinghdosanjh.RHINO\TickerBoard> npm fund

tickerboard@1.0.0

(base) PS C:\Users\jagdevsinghdosanjh.RHINO\TickerBoard>

Looks like you nailed the deployment, Jagdev—**TickerBoard is live and breathing on Vercel!** 🚀 The CLI output confirms everything built and deployed successfully, including proper compilation of your stock.js serverless function.

A few quick callouts:

**🧩 Build Warning Notes**

* **Build Settings Ignored**: You saw this message:
* Due to `builds` existing in your configuration file, the Build and Development Settings defined in your Project Settings will not apply.

That’s expected because you’re using a vercel.json with a builds field. It just means your JSON config takes priority over the web dashboard—no harm done.

* **ESM to CommonJS**: Vercel automatically transpiled your ESM-style export default in stock.js to CommonJS. That’s okay unless you explicitly wanted native ESM—then we’d need to add "type": "module" to your package.json.

**✅ Next Steps Checklist**

* 🧪 **Test Live API**: Try hitting https://ticker-board-frqwn7lel-jagdev-singh-dosanjhs-projects.vercel.app/api/stock?ticker=AAPL directly and check if you get JSON results. If not, I’ll help debug that next.
* 🔐 **Check Environment Variables**: Ensure your POLYGON\_API\_KEY is set in Vercel’s dashboard under Project > Settings > Environment Variables.
* 🧼 **Clean Up** vercel.json if needed: We can simplify routing or remove unused fields based on how the app evolves.
* 💡 **Enhance UX**: We can now add features like charts, search history, or even a dark mode to really polish the experience.

Let me know how the live version behaves—I’m here to help polish and scale it like a pro dashboard!

To check the functionality of POLYGON\_API\_KEY=VqyLgx5GuescBfJhHptD3WQNuX7YTgiF

POLYGON\_API\_KEY=VqyLgx5GuescBfJhHptD3WQNuX7YTgiF

<https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2025-01-01/2025-05-11?apiKey=YOUR_API_KEY>

<https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2025-01-01/2025-05-11?apiKey=VqyLgx5GuescBfJhHptD3WQNuX7YTgiF>

**✅ 4. Confirm vercel.json Routing**

Make sure your vercel.json includes this exact route:

json

{ "src": "/api/stock", "dest": "/api/stock.js" }

And that api/stock.js exists at the correct path.

**✅ 5. Double-Check File Naming and Placement**

* File should be: /TickerBoard/api/stock.js
* Not Stock.js, not in a subfolder, and not missing .js

If you’d like, paste your fetch code from the frontend here—I’ll help you confirm it’s wired up correctly. You’re *so* close to having this dashboard fully live!

the code is : { "version": 2, "builds": [ { "src": "api/\*.js", "use": "@vercel/node" } ], "routes": [ { "src": "/api/stock", "dest": "/api/stock.js" } { "src": "/", "dest": "/index.html" } { "src": "/(.\*)", "dest": "/$1" } ] }

Thanks, Jagdev—that's exactly where the issue lies. Your vercel.json is **missing commas between objects in the** routes **array**, which makes it invalid JSON and causes Vercel’s routing to silently fail. That’s why your API call isn’t hitting the backend, even though index.html is loading fine.

**✅ Here's the corrected version:**

json

{

"version": 2,

"builds": [

{

"src": "api/\*.js",

"use": "@vercel/node"

}

],

"routes": [

{

"src": "/api/stock",

"dest": "/api/stock.js"

},

{

"src": "/",

"dest": "/index.html"

},

{

"src": "/(.\*)",

"dest": "/$1"

}

]

}

**🚀 Next Steps**

1. Replace your current vercel.json with the fixed version above.
2. Restart the dev server:

bash

vercel dev

1. Refresh the browser and select a company in the dropdown.

You should now see your serverless function firing as expected and that "Network response was not ok" error disappear like a puff of smoke. If you're still hitting a snag, I can walk through the fetch logic next—just say the word.

With this newer code for vercel.json as { "version": 2, "builds": [ { "src": "api/\*.js", "use": "@vercel/node" } ], "routes": [ { "src": "/api/stock", "dest": "/api/stock.js" }, { "src": "/", "dest": "/index.html" }, { "src": "/(.\*)", "dest": "/$1" } ] } the index.html is not loading

Thanks for the update, Jagdev. Since your vercel.json is now syntactically correct and your index.html was loading before, but suddenly isn’t, here are a few things to double-check:

**✅ 1. Ensure index.html is inside the public/ folder**

Vercel serves static assets from the public directory. So the file path should be:

/TickerBoard/public/index.html

If it’s in the root or another folder, it won’t be found.

**✅ 2. Update the route to reflect Vercel’s static file handling**

Your current route:

json

{ "src": "/", "dest": "/index.html" }

is correct **only if** index.html is in the public/ folder. Vercel automatically maps /index.html to /public/index.html.

So no need to change this route—just make sure the file is in the right place.

**✅ 3. Check for conflicting files or folders**

Make sure there’s no index.js or index.ts in the root or api/ folder that might be confusing Vercel’s routing.

**✅ 4. Clear .vercel/cache and restart**

Sometimes stale cache can cause weird behavior. Try:

bash

rm -rf .vercel/cache

vercel dev

**✅ 5. Inspect the terminal and browser console**

Look for:

* Any 404 or MIME-type errors
* Whether the request to / is being routed to /index.html
* Whether the file is being served as text/html

If you’re still stuck, feel free to share your full folder structure again and I’ll help you trace the routing logic. You’re just one breadcrumb away from a working setup!

(base) PS C:\Users\jagdevsinghdosanjh.RHINO\TickerBoard> rm -rf .vercel/cache >> vercel dev >> Remove-Item : A parameter cannot be found that matches parameter name 'rf'. At line:1 char:4 + rm -rf .vercel/cache + ~~~ + CategoryInfo : InvalidArgument: (:) [Remove-Item], ParameterBindingException + FullyQualifiedErrorId : NamedParameterNotFound,Microsoft.PowerShell.Commands.RemoveItemCommand Vercel CLI 44.2.7 > Ready! Available at http://localhost:3000

Ah, classic *Unix vs PowerShell* moment! On Windows PowerShell, rm -rf doesn't work—that's a Unix/Linux style command. Here’s how to safely remove the Vercel cache folder using PowerShell syntax:

powershell

Remove-Item -Recurse -Force .vercel\cache

That’ll do the trick! Once it’s gone, restart vercel dev and it’ll regenerate a fresh cache. Since localhost:3000 is ready again, this might just clear any stale routing confusion and help your index.html load properly.

Let me know if that gets the page and API clicking together again—or if you want to peek at your fetch logic next, I’m all in.

that is the problem when i using commas

Ahh, I see it now—Jagdev, the culprit was *missing commas between route objects* in your vercel.json. JSON is pretty unforgiving: even one tiny missing comma between curly braces can make the whole config invalid, and suddenly none of your routes work—even the homepage.

That 404 you’re seeing at localhost:3000 confirms that Vercel’s dev server isn’t able to serve index.html due to the malformed JSON blocking all custom routes.

You're on the right track now. Here’s your corrected vercel.json one more time, clean and bulletproof:

json

{

"version": 2,

"builds": [

{

"src": "api/\*.js",

"use": "@vercel/node"

}

],

"routes": [

{

"src": "/api/stock",

"dest": "/api/stock.js"

},

{

"src": "/",

"dest": "/index.html"

},

{

"src": "/(.\*)",

"dest": "/$1"

}

]

}

Make sure the file is saved with **UTF-8 encoding**, and then relaunch the dev server with:

powershell

vercel dev

Once that’s up and running, open DevTools in your browser → Network tab → select a ticker and watch the magic happen! If you're still seeing issues with API calls, we’ll tune the fetch logic next.

Need me to double-check that part of your frontend too? I’m all ears.

Is this verce.json code correct: { "version": 2, "builds": [{"src": "api/\*.js","use": "@vercel/node"}], "routes": [{"src": "/api/stock","dest": "/api/stock.js"},{"src": "/","dest": "/index.html"},{"src": "/(.\*)","dest": "/$1"}] }

Yes, that vercel.json is **technically valid** JSON and should work just fine for your setup—especially since you're using a single serverless function and a static index.html in the public/ folder.

**✅ Here’s what your config is doing:**

* "builds": Tells Vercel to use Node.js for all .js files inside the api/ folder.
* "routes":
  + Maps /api/stock to api/stock.js
  + Serves /index.html from root (/)
  + Allows all other paths to resolve to their corresponding files (e.g., static assets like JS/CSS)

So yes—it’s compact, clean, and correct. If you're still running into issues, the root cause might now be in the fetch logic or the file paths inside index.html, but the routing config itself looks good.

Want to walk through a full project sanity check together? I'm right here.

This code without commans fails while parsing but with commans succeed. but after parsing the code fails to open index.html. after parsing the vercel.json if i delete these commas the index.html loads instantly. Why this happening? DIscuss

That’s a great observation, Jagdev—and quite an interesting case of behavior that feels paradoxical at first glance! Let’s break it down:

**🧩 What’s going on behind the scenes?**

There are two separate things at play here:

1. **When the commas are missing**:
   * The JSON is *invalid*, so **Vercel ignores** the entire vercel.json file.
   * As a result, Vercel falls back to its **zero configuration mode** and simply serves files from the /public folder.
   * That’s why index.html loads fine—Vercel sees no custom routing and just handles things the default way.
2. **When the commas are fixed (valid JSON)**:
   * Now Vercel **does read** the vercel.json.
   * But if there’s **something else subtly wrong in logic (not syntax)**—like incorrect path assumptions or missing static asset fallback—then those custom routes override the default behavior.
   * So now, **Vercel skips its default "serve from public/" behavior**, and instead applies only what’s in your routes. If those aren't 100% correct, even static files won't get served.

**⚠️ Why is that a problem?**

By defining a routes array, you're telling Vercel: **"Only serve files according to these rules—no more, no less."**

So if you miss any necessary routes—like one that handles loading /index.html properly—then that file simply won’t be served. That’s why with valid commas but an *imperfect* routes array, index.html vanishes.

**✅ How to resolve it cleanly**

Your current working config is really close. Here’s a revised suggestion that’s lean, valid, and gives both your API and static files room to breathe:

json

{

"version": 2,

"builds": [

{

"src": "api/\*.js",

"use": "@vercel/node"

}

]

}

You can actually **omit the** routes **block altogether** if:

* Your HTML lives in /public
* Your function lives in /api

Vercel will auto-route everything:

* / → /public/index.html
* /api/stock → /api/stock.js
* All static assets served normally

This “zero config” fallback works brilliantly when everything is in conventional folders—which you’ve done.

Want to keep custom routes? Then make sure index.html gets a route like this:

json

{

"src": "/",

"dest": "/index.html"

}

But be warned: with custom routes, you have to tell Vercel **how to serve everything**—or it'll just do nothing.

It’s like your JSON was a contract: 🧾 “Dear Vercel, trust me to handle all the routing.” If the contract is bad—or missing commas—Vercel shrugs and reverts to its own judgment.

Want help crafting the simplest config for both local and production deployment? Let's tailor it together.