Debugging 'Invalid Host Header' Error in a Next.js + Express Setup

Step 1: Understand the Error

The 'Invalid Host Header' error typically occurs when the development server (Webpack Dev Server or Next.js) receives requests from a host that it doesn't recognize or allow by default. This is often the case when using port forwarding, cloud IDEs (like Coder or Replit), or reverse proxies.

Step 2: Check Your Environment

- Are you using a custom domain or external tunneling service (like Coder, ngrok, Vercel preview, etc.)?
- Is your frontend running on a different domain than localhost (like pit-1.try.coder.app)?

Step 3: Next.js Specific Configuration

Update the next.config.js to explicitly allow the external host:

```
module.exports = {
  devIndicators: {
    autoPrerender: false,
  },
  webpackDevMiddleware: config => {
    config.watchOptions = {
      poll: 1000,
      aggregateTimeout: 300,
    };
    return config;
  },
  reactStrictMode: true,
  experimental: {
    appDir: true,
  },
};
```

Step 4: Add Allowed Host Header

You can run Next.js with a custom host header allowed:

```
HOST=0.0.0.0 next dev
```

Or, add this in your package.json:

```
"scripts": {
  "dev": "next dev -H 0.0.0.0 -p 3000"
}
```

Step 5: If Using Express Proxy or Middleware

If you're using a reverse proxy or Express middleware, make sure to forward the correct headers. Also set in Express:

```
app.set('trust proxy', true);
```

Step 6: DevServer Configuration for Webpack (Optional)

If you're not using Next.js but Webpack directly, configure devServer:

```
devServer: {
  host: '0.0.0.0',
  allowedHosts: ['all'],
}
```

Step 7: Test with Localhost

Try running both frontend and backend locally using localhost:3000 and localhost:5000 (or 8000) to confirm whether the issue only happens in cloud setups.

Step 8: Use Console & Network Logs

Open browser DevTools -> Network tab -> Look for blocked requests, 400 errors, CORS issues.

Also, check the browser console and terminal output (where npm run dev or yarn dev is run).

Step 9: Common Fix

Run frontend with allowed hosts manually like:

```
next dev -H 0.0.0.0
```

Then visit:

https://3000--main--your-project-url.pit-1.try.coder.app

You can also use .env.local:

```
HOST=0.0.0.0
PORT=3000
```

Step 10: Document the Fix

Once fixed, write a short README or fix doc so the team can reapply the same steps in future deployments or cloud dev environments.