File: server/vite.ts

Type: ts | **Size:** 2254 bytes

```ts
import express, { type Express } from "express";
import fs from "fs";
import path from "path";
import { createServer as createViteServer, createLogger } from "vite";
import { type Server } from "http";
import viteConfig from "../vite.config";
import { nanoid } from "nanoid";
const viteLogger = createLogger();
export function log(message: string, source = "express") {
  const formattedTime = new Date().toLocaleTimeString("en-US", {
    hour: "numeric",
    minute: "2-digit",
    second: "2-digit",
    hour12: true,
  });
  console.log(`${formattedTime} [${source}] ${message}`);
}
export async function setupVite(app: Express, server: Server) {
  const serverOptions = {
    middlewareMode: true,
    hmr: { server },
    allowedHosts: true,
  };
  const vite = await createViteServer({
    ...viteConfig,
    configFile: false,
    customLogger: {
      ...viteLogger,
      error: (msg, options) => {
        viteLogger.error(msg, options);
        process.exit(1);
      },
    },
    server: serverOptions,
    appType: "custom",
  });
  app.use(vite.middlewares);
  app.use("*", async (req, res, next) => {
    const url = req.originalUrl;
    try {
      const clientTemplate = path.resolve(
        import.meta.dirname,
        "..",
        "client",
        "index.html",
      );
      // always reload the index.html file from disk incase it changes
      let template = await fs.promises.readFile(clientTemplate, "utf-8");
      template = template.replace(
        `src="/src/main.tsx"`,
        `src="/src/main.tsx?v=${nanoid()}"`,
      );
      const page = await vite.transformIndexHtml(url, template);
      res.status(200).set({ "Content-Type": "text/html" }).end(page);
    } catch (e) {
      vite.ssrFixStacktrace(e as Error);
      next(e);
    }
  });
}
export function serveStatic(app: Express) {
  const distPath = path.resolve(import.meta.dirname, "public");
  if (!fs.existsSync(distPath)) {
    throw new Error(
      `Could not find the build directory: ${distPath}, make sure to build the client first`,
    );
  }
  app.use(express.static(distPath));
  // fall through to index.html if the file doesn't exist
  app.use("*", (_req, res) => {
    res.sendFile(path.resolve(distPath, "index.html"));
  });
}
```