// import { execa } from 'execa';

// import \* as fs from 'fs';

// import \* as path from 'path';

// import { config } from '../config';

// import { getRandomUserAgent } from './userAgents';

// export interface SubtitleDownloadResult {

// filePath: string;

// langCode: string;

// }

// export async function downloadSubtitles(

// videoId: string,

// outputDir: string

// ): Promise<SubtitleDownloadResult> {

// const baseUrl = `https://www.youtube.com/watch?v=${videoId}`;

// const output = path.join(outputDir, `${videoId}.%(ext)s`);

// const cookiesFile = config.COOKIES\_PATH;

// if (!fs.existsSync(outputDir)) {

// fs.mkdirSync(outputDir, { recursive: true });

// }

// const buildCommonArgs = (): string[] => {

// const userAgent = getRandomUserAgent();

// const args = [

// '--cookies', cookiesFile,

// '--user-agent', userAgent,

// '--no-check-certificate',

// '--write-auto-sub',

// '--write-sub',

// '--skip-download',

// '-o', output,

// baseUrl,

// ];

// if (config.PROXY) {

// args.unshift('--proxy', config.PROXY);

// }

// return args;

// };

// const tryDownload = async (lang: string, maxRetries = 3): Promise<string | null> => {

// for (let attempt = 1; attempt <= maxRetries; attempt++) {

// try {

// await execa('./bin/yt-dlp', ['--sub-lang', lang, ...buildCommonArgs()]);

// const match = lang === 'en' ? '.en.vtt' : '.vtt';

// const subtitleFile = fs

// .readdirSync(outputDir)

// .find(f => f.startsWith(videoId) && f.endsWith(match));

// if (subtitleFile) {

// return subtitleFile;

// }

// } catch (err: any) {

// if (attempt === maxRetries) {

// throw new Error(`Failed to download subtitles after ${maxRetries} attempts: ${err.message}`);

// }

// console.warn(`[Retry ${attempt}] Failed to download ${lang} subtitles for ${videoId}. Retrying...`);

// await new Promise(res => setTimeout(res, 2000));

// }

// }

// return null;

// };

// const enSubtitle = await tryDownload('en');

// if (enSubtitle) {

// return { filePath: path.join(outputDir, enSubtitle), langCode: 'en' };

// }

// const fallbackSubtitle = await tryDownload('best');

// if (fallbackSubtitle) {

// const langMatch = fallbackSubtitle.match(/\.(\w+)\.vtt$/);

// const detectedLang = langMatch?.[1] ?? 'unknown';

// return { filePath: path.join(outputDir, fallbackSubtitle), langCode: detectedLang };

// }

// throw new Error(`No subtitles found for video: ${videoId}`);

// }