bland.ts

import { agentConfig } from '../config/agent-config';

const BLAND\_API\_KEY = process.env.NEXT\_PUBLIC\_BLAND\_API\_KEY!;

const BLAND\_API\_URL = 'https://api.bland.ai';

const BLAND\_WEB\_URL = 'https://web.bland.ai';

if (!BLAND\_API\_KEY) {

throw new Error('BLAND\_API\_KEY is not set in environment variables');

}

export async function createWebAgent() {

try {

console.log('Creating new web agent...');

const response = await fetch(`${BLAND\_API\_URL}/v1/agents`, {

method: 'POST',

headers: {

'authorization': BLAND\_API\_KEY,

'Content-Type': 'application/json'

},

body: JSON.stringify(agentConfig)

});

if (!response.ok) {

const errorText = await response.text();

console.error('Agent creation error:', errorText);

throw new Error(`Failed to create agent: ${response.status} - ${errorText}`);

}

const data = await response.json();

console.log('Created agent:', data);

return data;

} catch (error) {

console.error('Error creating agent:', error);

throw error;

}

}

export async function getSessionToken(agentId: string) {

try {

const response = await fetch(`${BLAND\_WEB\_URL}/v1/agents/${agentId}/authorize`, {

method: 'POST',

headers: {

'authorization': BLAND\_API\_KEY,

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

'Accept': 'application/json'

},

body: JSON.stringify({})

});

if (!response.ok) {

const errorText = await response.text();

throw new Error(`API responded with status: ${response.status} - ${errorText}`);

}

return response.json();

} catch (error) {

console.error('Error getting session token:', error);

throw error;

}

}