```
const express = require('express');
const cors = require('cors');
const rateLimit = require('express-rate-limit');
const fs = require('fs');
const path = require('path');
require('dotenv').config();

const app = express();
const PORT = process.env.PORT || 3000;

// ── ALLOWED ORIGINS ──────────────────────────────────────────
const ALLOWED_ORIGINS = (process.env.ALLOWED_ORIGINS || 'http://localhost:8080')
  .split(',')
  .map(o => o.trim());

// ── CORS ─────────────────────────────────────────────────────
app.use(cors({
  origin: function (origin, callback) {
    if (!origin) return callback(null, false);
    if (ALLOWED_ORIGINS.includes(origin)) return callback(null, true);
    return callback(new Error('Not allowed by CORS'));
  },
  methods: ['GET', 'POST'],
  credentials: false,
}));

app.use(express.json({ limit: '10kb' }));

// ── RATE LIMITING ────────────────────────────────────────────
const limiter = rateLimit({
  windowMs: 60 * 1000,
  max: 30,
  message: { error: 'Too many requests, slow down.' },
  standardHeaders: true,
  legacyHeaders: false,
});
app.use('/api/', limiter);

// ── LOGGER HELPER ────────────────────────────────────────────
const LOG_FILE = process.env.LOG_FILE || path.join(__dirname, 'deployments.log');
function writeLog(entry) {
  const line = JSON.stringify(entry) + '\n';
  fs.appendFile(LOG_FILE, line, (err) => {
    if (err) console.error('[logger] Failed to write log:', err.message);
  });
  console.log('[log]', JSON.stringify(entry));
}

// ── HEALTH CHECK ─────────────────────────────────────────────
app.get('/', (req, res) => {
  res.json({ status: 'ok', service: 'SolMint API', version: '1.0.0' });
});

// ── GET CONFIG ───────────────────────────────────────────────
app.get('/api/config', (req, res) => {
  const origin = req.headers.origin || req.headers.referer || '';
  const allowed = ALLOWED_ORIGINS.some(o => origin.startsWith(o));
  if (!allowed) return res.status(403).json({ error: 'Forbidden' });

  if (!process.env.OWNER_WALLET) {
    console.error('[config] OWNER_WALLET not set in environment!');
    return res.status(500).json({ error: 'Server not configured' });
  }

  res.json({
    feeAddress: process.env.OWNER_WALLET,
    network: process.env.NETWORK || 'mainnet',
    baseFee: parseFloat(process.env.BASE_FEE || '0.2'),
    optionFees: {
      'revoke-mint':   parseFloat(process.env.FEE_REVOKE_MINT || '0.1'),
      'revoke-freeze': parseFloat(process.env.FEE_REVOKE_FREEZE || '0.1'),
      'immutable':     parseFloat(process.env.FEE_IMMUTABLE || '0.1'),
      'update-auth':   parseFloat(process.env.FEE_UPDATE_AUTH || '0.1'),
      'tax':           parseFloat(process.env.FEE_TAX || '0.2'),
      'lpburn':        parseFloat(process.env.FEE_LPBURN || '0.1'),
      'creator':       parseFloat(process.env.FEE_CREATOR || '0.1'),
    },
  });
});

// ── LOG DEPLOYMENT ───────────────────────────────────────────
app.post('/api/log', (req, res) => {
  const origin = req.headers.origin || '';
  const allowed = ALLOWED_ORIGINS.some(o => origin.startsWith(o));
  if (!allowed) return res.status(403).json({ error: 'Forbidden' });

  const { mintAddress, network, options } = req.body || {};

  // Validate mint address format
  if (!mintAddress || !/^[1-9A-HJ-NP-Za-km-z]{32,44}$/.test(mintAddress)) {
    return res.status(400).json({ error: 'Invalid mint address' });
  }

  const safeNetwork = ['mainnet', 'devnet'].includes(network) ? network : 'unknow
  const VALID_OPTIONS = ['revoke-mint', 'revoke-freeze', 'immutable', 'update-aut
  const safeOptions = Array.isArray(options)
    ? options.filter(o => VALID_OPTIONS.includes(o))
    : [];

  writeLog({
    ts: new Date().toISOString(),
    mintAddress,
    network: safeNetwork,
    options: safeOptions,
    ip: req.ip,
  });

  res.json({ ok: true });
});

// ── 404 ───────────────────────────────────────────────────────
app.use((req, res) => res.status(404).json({ error: 'Not found' }));

// ── ERROR HANDLER ─────────────────────────────────────────────
app.use((err, req, res, next) => {
  if (err.message === 'Not allowed by CORS') {
    return res.status(403).json({ error: 'Origin not allowed' });
  }
  console.error(err);
  res.status(500).json({ error: 'Internal server error' });
});

app.listen(PORT, () => {
  console.log(`✅ SolMint API running on port ${PORT}`);
  console.log(`   Allowed origins: ${ALLOWED_ORIGINS.join(', ')}`);
  console.log(`   Owner wallet: ${process.env.OWNER_WALLET ? '***SET***' : '⚠️  N
  console.log(`   Network: ${process.env.NETWORK || 'mainnet'}`);
});
```