

Local Development Setup

Local Development Setup

“ **Project:** Bilko **Version:** 0.1 **Date:** 2026-02-23 **Author:** Ops Architect **Status:** Draft **Reviewers:** Tech Lead, Alem Bašić

Document History

Version	Date	Author	Changes
0.1	2026-02-23	Ops Architect	Initial draft

Overview

This guide gets a Bilko development environment running from scratch. Estimated time: 45-60 minutes on a clean machine.

Full environment documentation: `../infrastructure/ENVIRONMENT.md`

Prerequisites

Software	Version	Check	Install
Node.js	20.x (LTS)	<code>node --version</code>	https://nodejs.org
pnpm	8.x+	<code>pnpm --version</code>	<code>npm install -g pnpm</code>
PostgreSQL	15+	<code>psql --version</code>	https://postgresql.org/download or Docker
Git	Latest	<code>git --version</code>	https://git-scm.com

Optional (recommended):

- VS Code with extensions: ESLint, Prettier, Prisma, Tailwind CSS IntelliSense
 - TablePlus or DBeaver for database browsing
 - Postman or Insomnia for API testing
-

Installation Steps

Step 1: Clone Repository

```
git clone https://github.com/alai-holding/bilko.git
cd bilko
```

Step 2: Install Dependencies

```
# Install all workspace dependencies (from repo root)
pnpm install
```

This installs dependencies for:

- Root workspace (Turborepo)
- `apps/web` (Next.js 15)
- `apps/api` (Express)
- `packages/database` (Prisma)
- `packages/ui` (shared components)

Step 3: Set Up PostgreSQL

Option A: Local PostgreSQL installation

```
# Connect to PostgreSQL as admin
psql -U postgres

# Run in psql:
CREATE DATABASE bilko_dev;
CREATE USER bilko WITH PASSWORD 'bilko';
GRANT ALL PRIVILEGES ON DATABASE bilko_dev TO bilko;
\q
```

Option B: Docker (no PostgreSQL installation required)

```
docker run --name bilko-postgres \  
  -e POSTGRES_USER=bilko \  
  -e POSTGRES_PASSWORD=bilko \  
  -e POSTGRES_DB=bilko_dev \  
  -p 5432:5432 \  
  --restart unless-stopped \  
  -d postgres:15  
  
# Verify it's running  
docker ps | grep bilko-postgres
```

Step 4: Configure Environment Variables

Create `apps/api/.env`:

```
cp apps/api/.env.example apps/api/.env  
# Edit with your values
```

Content:

```
# Database  
DATABASE_URL=postgresql://bilko:bilko@localhost:5432/bilko_dev  
  
# JWT Secrets (generate with: openssl rand -base64 32)  
JWT_SECRET=dev-secret-change-in-production  
JWT_REFRESH_SECRET=dev-refresh-secret-change-in-production  
  
# Email – leave empty for local dev (invoices won't be emailed but can be created)  
SENDGRID_API_KEY=  
  
# Cloudflare R2 – leave empty for local dev (file uploads will fail gracefully)  
R2_ACCESS_KEY_ID=  
R2_SECRET_ACCESS_KEY=  
R2_BUCKET_NAME=bilko-receipts-dev  
R2_ENDPOINT=  
  
# App  
PORT=4000  
NODE_ENV=development  
ALLOWED_ORIGINS=http://localhost:3000
```

Create `apps/web/.env.local`:

```
cp apps/web/.env.local.example apps/web/.env.local
```

Content:

```
NEXT_PUBLIC_API_URL=http://localhost:4000  
NEXT_PUBLIC_APP_ENV=development
```

Step 5: Run Database Migrations

```
cd packages/database  
npx prisma migrate dev  
npx prisma generate  
cd ../../
```

This:

1. Applies all 15+ table migrations to `bilko_dev`
2. Generates the Prisma Client TypeScript types

Step 6: Seed Database (Optional)

```
cd packages/database  
npx prisma db seed  
cd ../../
```

Creates:

- Demo organization: `Test Company d.o.o.` (country: RS, currency: RSD)
- Demo user: `demo@bilko.io` / `demo123` (role: owner)
- Sample contacts, one invoice (draft state)

Running the Application

Start All Services

```
# From repo root – starts both frontend and backend via Turborepo
pnpm run dev
```

Services:

- **Frontend:** <http://localhost:3000> (Next.js dev server)
- **Backend:** <http://localhost:4000> (Express, hot reload via nodemon)

Start Individual Services

```
# Frontend only
cd apps/web && pnpm run dev

# Backend only
cd apps/api && pnpm run dev
```

Prisma Studio (Database GUI)

```
cd packages/database && npx prisma studio
# Opens at http://localhost:5555
```

Features: Browse tables, edit records, view relations, run queries.

Running Tests

All Tests

```
pnpm run test
```

Unit Tests

```
# Run
pnpm run test:unit

# Watch mode (re-run on save)
pnpm run test:unit -- --watch
```

```
# With coverage report
pnpm run test:unit -- --coverage

# Single file
pnpm run test:unit -- vat.test.ts
```

Integration Tests

Requires the test database `bilko_test`:

```
# Create test database
psql -U postgres -c "CREATE DATABASE bilko_test;"
psql -U postgres -c "GRANT ALL PRIVILEGES ON DATABASE bilko_test TO bilko;"

# Run integration tests
pnpm run test:integration

# Single file
pnpm run test:integration -- invoices-api.test.ts
```

E2E Tests

Requires both frontend and backend running:

```
# Terminal 1: Start dev servers
pnpm run dev

# Terminal 2: Run E2E tests
pnpm run test:e2e

# Headed mode (see browser)
pnpm run test:e2e -- --headed

# Debug mode (pause on failure)
pnpm run test:e2e -- --debug
```

Common Tasks

Create a Database Migration

After modifying `packages/database/prisma/schema.prisma`:

```
cd packages/database
npx prisma migrate dev --name describe_your_change
npx prisma generate
```

Reset Database (DEV ONLY — deletes all data)

```
cd packages/database
npx prisma migrate reset
# Type 'y' to confirm
```

Lint & Type Check

```
pnpm run lint          # ESLint + Prettier check
pnpm run lint -- --fix # Auto-fix formatting
pnpm run type-check    # TypeScript strict mode check
```

Build for Production

```
pnpm run build
# Builds apps/web/.next and apps/api/dist
```

VS Code Setup

Recommended Extensions

Create `.vscode/extensions.json` (commit this file):

```
{
  "recommendations": [
```

```
"dbaeumer.vscode-eslint",
"esbenp.prettier-vscode",
"bradlc.vscode-tailwindcss",
"prisma.prisma",
"ms-vscode.vscode-typescript-next"
]
}
```

Install all: `Ctrl/Cmd+Shift+P` → "Extensions: Show Recommended Extensions" → Install Workspace Recommendations

Workspace Settings

Create `.vscode/settings.json`:

```
{
  "editor.formatOnSave": true,
  "editor.defaultFormatter": "esbenp.prettier-vscode",
  "editor.codeActionsOnSave": {
    "source.fixAll.eslint": true
  },
  "typescript.tsdk": "node_modules/typescript/lib",
  "tailwindCSS.experimental.classRegex": [
    ["cva\\(((\\^)*))\\)", "[\\\"'`](\\^\\\"'`)*).*?[\\\"'`]"]
  ]
}
```

Troubleshooting

PostgreSQL Connection Failed

Error: `Can't reach database server at localhost:5432`

1. Check if PostgreSQL is running: `pg_isready -h localhost`
2. If Docker: `docker ps | grep bilko-postgres` — if not running: `docker start bilko-postgres`
3. Verify credentials in `apps/api/.env` match database setup
4. Check port 5432 not in use by another process: `lsof -i :5432`

Port Already in Use

Error: Port 3000 is already in use

```
# Kill process on port 3000
lsof -ti:3000 | xargs kill

# Or use a different port
PORT=3001 pnpm run dev
```

Prisma Client Not Generated

Error: @prisma/client did not initialize

```
cd packages/database && npx prisma generate
```

TypeScript Errors After Pulling

```
pnpm install
cd packages/database && npx prisma generate
pnpm run type-check
```

Hot Reload Not Working (Backend)

1. Kill and restart dev server
2. Check nodemon is watching the right directory

Hot Reload Not Working (Frontend)

```
# Clear Next.js cache
rm -rf apps/web/.next
pnpm run dev
```

Environment Variables Reference

See [environment-configuration.md](#) for full reference.

Quick reference for local dev (`apps/api/.env`):

Variable	Required Locally	Default
<code>DATABASE_URL</code>	Yes	<code>postgresql://bilko:bilko@localhost:5432/bilko_dev</code>
<code>JWT_SECRET</code>	Yes	Any non-empty string
<code>JWT_REFRESH_SECRET</code>	Yes	Any non-empty string
<code>SENDGRID_API_KEY</code>	No	Empty (email sending won't work locally)
<code>R2_ACCESS_KEY_ID</code>	No	Empty (file uploads won't work locally)
<code>PORT</code>	No	4000
<code>NODE_ENV</code>	No	development
<code>ALLOWED_ORIGINS</code>	No	<code>http://localhost:3000</code>

Related Documents

- [ENVIRONMENT.md](#) — authoritative environment documentation
- [Developer Onboarding Guide](#)
- [Coding Standards](#)
- [BILKO CLAUDE.md](#)

Approval

Role	Name	Date	Signature
Author	Ops Architect	2026-02-23	
Reviewer	Tech Lead		
Approver	Alem Bašić		

Revision #3

Created 2026-02-24 23:11:27 UTC by John

Updated 2026-05-31 20:04:21 UTC by John