Developer Onboarding

Developer Onboarding Guide

> **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

Welcome to Bilko

Bilko is a cloud accounting SaaS for Balkan SMBs (Serbia, Bosnia, Croatia). It handles invoicing, expense tracking, VAT reporting, and financial bookkeeping. This guide gets a new developer from zero to productive in one day.

Critical context: Bilko handles real financial data. Bugs in VAT calculations, double-entry bookkeeping, or NUMERIC precision are not just bugs — they can cause tax compliance failures for users. Read the accounting fundamentals section before writing any financial logic.

Day 1 Checklist

Access & Accounts

Contact Alem Bašić (alem@alai.no) to be granted:

- ☐ GitHub: Added to `alai-holding` organization, `bilko` repository

- [ ] Railway: Added to Bilko project (staging environment access only)
- [ ] Vercel: Added to ALAI team (staging/preview access only)
- [ ] Slack: Added to #bilko-dev channel
- [ ] Sentry: Added to bilko-frontend and bilko-backend projects (view only)

Local Setup

- [ ] Follow local-development-setup.md — estimated 45 minutes
- [ ] Run `pnpm run dev` — both frontend (port 3000) and backend (port 4000) start
- [ ] Run `pnpm run test:unit` — all tests pass
- [ ] Open http://localhost:3000 — dashboard loads (mock data at MVP stage)
- [ ] Open http://localhost:5555 (Prisma Studio) — database is visible

Reading (complete on Day 1)

- [ ] **BILKO CLAUDE.md** (`~/ALAI/products/Bilko/CLAUDE.md`) — project overview, dev rules
- [ ] **Coding Standards** (`docs/templates/DEVELOPER-EXPERIENCE/coding-standards.md`) — rules you must follow
- [ ] **Database Schema** (`packages/database/prisma/schema.prisma`) — the 15 models
- [ ] **Test Strategy** (`docs/templates/TESTING/test-strategy.md`) — how we test

Tech Stack

Frontend (apps/web/)

Technology	Version	Purpose
Next.js	15.0.0	React framework, SSR
React	19.0.0	UI library
TypeScript	5.3.0	Type safety (strict mode)
Tailwind CSS	4.0.0	Styling
shadcn/ui	Latest	Component library (Radix UI + Tailwind)
Zustand	4.5.0	State management

Technology	Version	Purpose
Recharts	2.15.0	Charts (revenue, expenses)
Lucide React	Latest	Icons

Backend (apps/api/)

Technology	Version	Purpose
Express	Latest	API framework
TypeScript	5.3.0	Type safety (strict mode)
Prisma	Latest	ORM + migrations
PostgreSQL	15	Database
Passport.js	Latest	Authentication strategy
Zod	Latest	Request validation
Helmet	Latest	Security headers
bcrypt	Latest	Password hashing (12 rounds)
jsonwebtoken	Latest	JWT tokens
decimal.js	Latest	NUMERIC precision arithmetic

Monorepo

Tool	Purpose
Turborepo	Build orchestration, incremental builds
pnpm workspaces	Package management
packages/database	Prisma schema + client (shared)
packages/ui	Shared UI components (empty scaffold at MVP)

Project Structure

```
Bilko/
├── apps/
│   ├── web/              # Next.js 15 frontend (bilko.io)
│   │   ├── app/          # Next.js App Router pages
│   │   ├── components/   # React components
```

```
│   │   ├── lib/          # Utilities, API client
│   │   └── lib/mock-data.ts  # MOCK DATA — replace with real API calls
│   └── api/              # Express backend (api.bilko.io)
│       ├── src/
│       │   ├── routes/   # Express route handlers
│       │   ├── services/ # Business logic
│       │   ├── utils/    # Utilities (VAT, double-entry, currency)
│       │   ├── middleware/  # Auth, validation, error handling
│       │   └── test/     # Test setup, factories, helpers
│       └── CLAUDE.md     # Backend-specific AI instructions
├── packages/
│   ├── database/         # Prisma schema + generated client
│   │   └── prisma/
│   │       ├── schema.prisma  # 15 models — READ THIS FIRST
│   │       └── migrations/    # All DB migrations
│   └── ui/               # Shared components (scaffold only)
└── docs/                 # Documentation
    ├── infrastructure/   # CI/CD, deployment, environment setup
    ├── testing/          # Testing guide, test inventory
    └── templates/        # Document templates (this file is here)
```

Accounting Fundamentals (Required Reading)

You don't need to be an accountant, but you need to understand these concepts:

Double-Entry Bookkeeping

Every financial transaction has two sides: a debit and a credit. They must always balance (debit = credit). Example:

- Invoice of 60,000 RSD:
  - DEBIT: Accounts Receivable 60,000 RSD
  - CREDIT: Revenue 50,000 RSD + VAT Payable 10,000 RSD

Bilko enforces this in the `Transaction` + `TransactionEntry` models. You CANNOT create a transaction where debit ≠ credit.

NUMERIC(19,4) — Never Use float for Money

```
// WRONG — float arithmetic has precision errors
const vat = 100 * 0.20;  // Could be 20.000000000000004

// CORRECT — use Decimal from decimal.js
import { Decimal } from 'decimal.js';
const vat = new Decimal('100').mul('0.20');  // Exactly 20.0000
```

All monetary columns in PostgreSQL are `NUMERIC(19,4)`. All monetary values in TypeScript must use `Decimal`.

Multi-Currency with Rate Locking

When a transaction is created in EUR but the base currency is RSD, the exchange rate is locked at the transaction date — not today's rate. This is required for accurate historical financial reports.

VAT by Country

- Serbia (RS): Standard 20%, Reduced 10%, Zero 0% (exports)
- Bosnia (BA): Standard 17%, Zero 0% (exports). No reduced rate.
- Croatia (HR): Standard 25%, Reduced 13%, Super-reduced 5%

Development Workflow

Starting Work on a Feature

```
# 1. Pull latest main
git checkout main && git pull

# 2. Create feature branch
git checkout -b feature/INV-123-invoice-pdf-generation

# 3. Make changes
# ... code ...

# 4. Test locally
```

```
pnpm run lint
pnpm run type-check
pnpm run test:unit -- --coverage
pnpm run test:integration

# 5. Commit
git add apps/api/src/services/invoice.service.ts
git add apps/api/src/services/invoice.service.test.ts
git commit -m "feat: add PDF generation for invoices (INV-123)"

# 6. Push and create PR
git push origin feature/INV-123-invoice-pdf-generation
# Create PR on GitHub — CI runs automatically
```

Commit Message Format

```
type: description (issue-id)

Types: feat, fix, refactor, test, docs, chore
Examples:
  feat: add Serbian VAT calculation with 10% reduced rate
  fix: correct decimal precision in VAT calculation (BILKO-456)
  test: add unit tests for double-entry validation
  refactor: extract invoice total calculation to service layer
```

PR Requirements

- Description explains what and why
- Links to GitHub issue or Mission Control task
- All CI checks green
- At least 1 reviewer approval
- Tests written for all new business logic

Running the Application

```
# Install dependencies
pnpm install
```

```
# Run database migrations
cd packages/database && npx prisma migrate dev && npx prisma generate

# Start all services (from root)
pnpm run dev
# Frontend: http://localhost:3000
# Backend: http://localhost:4000
# Prisma Studio: npx prisma studio (port 5555)

# Run all tests
pnpm run test

# Build for production (verify before deploy)
pnpm run build
```

Key Contacts

Person	Role	Contact
Alem Bašić	Founder/CEO (Human)	alem@alai.no, +47 40 47 42 51
John	AI Director (Claude Opus)	Handles coordination, tasks, infra questions

Slack channels:

- `#bilko-dev` — development discussion
- `#bilko-deploys` — deployment notifications and alerts

Getting Help

1. **Code question:** Check `CLAUDE.md` files (root + `apps/api/CLAUDE.md`) first
2. **Architecture question:** Check `docs/` directory
3. **Database schema:** `packages/database/prisma/schema.prisma`
4. **Still stuck:** Ask in #bilko-dev Slack or create a GitHub Discussion

---

First Week Milestones

Day	Goal
1	Dev environment running, tests passing, codebase read
2-3	First small PR merged (bug fix or test)
4-5	First feature PR opened
End of week 1	Comfortable with invoice + expense flow

Related Documents

- Local Development Setup
- Coding Standards
- Test Strategy
- BILKO CLAUDE.md
- ENVIRONMENT.md

Approval

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

Revision #3
Created 2026-02-24 23:11:27 UTC by John
Updated 2026-05-31 20:04:20 UTC by John