

Developer Onboarding Guide

Developer Onboarding Guide

“ **Project:** {{PROJECT_NAME}} **Version:** {{VERSION}} **Date:** {{DATE}}
Author: {{AUTHOR}} **Status:** Draft | In Review | Approved **Reviewers:**
{{REVIEWERS}}

Document History

Version	Date	Author	Changes
0.1	{{DATE}}	{{AUTHOR}}	Initial draft

Welcome

Welcome to **{{PROJECT_NAME}}**! This guide will help you go from zero to a fully working development environment and your first contribution.

If anything in this guide is unclear or out of date, please update it as you go — good documentation is a team responsibility. Your onboarding buddy is **{{BUDDY}}** — reach out to them anytime.

Your first week at a glance:

- Day 1: Set up your environment and get the app running locally
- Day 2-3: Read the architecture overview, explore the codebase
- Day 4-5: Work through your first small ticket with your buddy's guidance

1. Prerequisites

Hardware Requirements

Component	Minimum	Recommended
CPU	{{MIN_CPU}}	{{REC_CPU}}
RAM	{{MIN_RAM}}GB	{{REC_RAM}}GB
Disk	{{MIN_DISK}}GB free	SSD with {{REC_DISK}}GB free
OS	{{SUPPORTED_OS}}	

Accounts You Need

Account	Why	How to Get Access
GitHub / GitLab	Code repository	Ask {{GITHUB_ADMIN}}
{{CLOUD_PROVIDER}}	Cloud resources	Request via {{CLOUD_REQUEST_PROCESS}}
{{CI_CD_PLATFORM}}	CI/CD pipeline	Auto-provisioned on org invite
{{ERROR_TRACKER}}	Error monitoring	Ask {{SENTRY_ADMIN}}
{{PROJECT_MGMT}}	Tickets / tasks	Ask {{PM_ADMIN}}
{{COMMUNICATION}}	Team communication	Ask your manager
Vault / Secrets	Development secrets	Ask {{VAULT_ADMIN}}

Expected setup time for all accounts: {{ACCOUNT_SETUP_TIME}} days

Access Requests

Access Type	Contact	Process	SLA
GitHub organization	{{CONTACT}}	{{PROCESS}}	{{SLA}}
Cloud console (read-only dev)	{{CONTACT}}	{{PROCESS}}	{{SLA}}
VPN	{{CONTACT}}	{{PROCESS}}	{{SLA}}
Production read access	{{CONTACT}}	Requires security training first	{{SLA}}

2. Development Environment Setup

2.1 macOS Setup

```
# Install Homebrew (if not already installed)
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# Install required tools
brew install {{TOOLS_LIST}}

# e.g.: brew install git node nvm docker kubectl terraform

# Install nvm (Node version manager)
brew install nvm
echo 'export NVM_DIR="$HOME/.nvm"' >> ~/.zshrc
echo '[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"' >> ~/.zshrc
source ~/.zshrc

# Install required Node version
nvm install {{NODE_VERSION}}
nvm use {{NODE_VERSION}}
nvm alias default {{NODE_VERSION}}
```

2.2 Linux Setup

```
# Update package index
sudo apt update && sudo apt upgrade -y

# Install required tools
sudo apt install -y {{LINUX_TOOLS}}

# Install nvm
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v{{NVM_VERSION}}/install.sh | bash
source ~/.bashrc

# Install required Node version
nvm install {{NODE_VERSION}}
```

2.3 Windows Setup

```
# Enable WSL2
wsl --install
```

```
# After WSL2 is set up, follow the Linux setup instructions inside WSL2
# Recommended: Use VS Code with Remote WSL extension
```

2.4 Required Software & Versions

Software	Required Version	Install
Node.js	{{NODE_VERSION}}	<code>nvm install {{NODE_VERSION}}</code>
npm / yarn / pnpm	{{PKG_MGR_VERSION}}	Included with Node
Docker	{{DOCKER_VERSION}}+	docker.com/get-docker
Docker Compose	{{COMPOSE_VERSION}}+	Included with Docker Desktop
{{OTHER_TOOL}}	{{VERSION}}	{{INSTALL_CMD}}

Check your versions:

```
node --version # Should be v{{NODE_VERSION}}
docker --version # Should be {{DOCKER_VERSION}}+
```

2.5 IDE Setup

Recommended IDE: {{IDE}}

Required extensions:

```
# VS Code – install all recommended extensions
code --install-extension {{EXTENSION_1}}
code --install-extension {{EXTENSION_2}}
code --install-extension {{EXTENSION_3}}
# OR: Open project in VS Code → "Install recommended extensions" prompt will appear
```

Recommended extensions:

Extension	Purpose
{{EXT_1}}	{{PURPOSE}}
{{EXT_2}}	{{PURPOSE}}
{{EXT_3}}	{{PURPOSE}}

2.6 Environment Variables

```
# Copy the example env file
cp .env.example .env

# Fill in your local values
# Ask your onboarding buddy for any values marked {{SECRET}}
```

Key variables to set:

Variable	What to put	Where to get it
<code>DATABASE_URL</code>	<code>postgresql://localhost:5432/{{APP}}_dev</code>	Local setup
<code>{{SECRET_VAR}}</code>	Your dev secret	Ask {{BUDDY}}

3. Repository Setup

```
# Step 1: Clone the repository
git clone {{REPO_URL}}
cd {{REPO_NAME}}

# Step 2: Install dependencies
{{INSTALL_CMD}} # npm install / yarn / pnpm install

# Step 3: Set up the database
{{DB_SETUP_CMD}} # e.g., npm run db:create && npm run db:migrate

# Step 4: Seed development data
{{SEED_CMD}} # e.g., npm run db:seed

# Step 5: Configure environment (see 2.6)
cp .env.example .env
# Edit .env with your values

# Step 6: Start the application
{{DEV_CMD}} # e.g., npm run dev

# Step 7: Open the application
open http://localhost:{{PORT}}
```

4. Architecture Overview

High-level architecture: See [deployment-architecture.md](#)

In brief:

- **Frontend:** `{{FRONTEND_STACK}}` — located in `{{FRONTEND_DIR}}`
- **Backend:** `{{BACKEND_STACK}}` — located in `{{BACKEND_DIR}}`
- **Database:** `{{DB_ENGINE}}` — schema in `{{SCHEMA_DIR}}`
- **Key external services:** `{{EXTERNAL_SERVICES}}`

Codebase tour (key directories):

Directory	Purpose
<code>{{DIR_1}}</code>	<code>{{PURPOSE_1}}</code>
<code>{{DIR_2}}</code>	<code>{{PURPOSE_2}}</code>
<code>{{DIR_3}}</code>	<code>{{PURPOSE_3}}</code>
<code>{{DIR_4}}</code>	<code>{{PURPOSE_4}}</code>

5. Key Documentation Links

Document	Link	Purpose
Architecture	deployment-architecture.md	How the system is structured
API Documentation	<code>{{API_DOCS_LINK}}</code>	API reference
Design System	<code>{{DESIGN_SYSTEM_LINK}}</code>	UI components
Coding Standards	coding-standards.md	How we write code
Test Strategy	test-strategy.md	How we test
CI/CD Pipeline	cicd-pipeline.md	How deployments work
Tech Decisions (ADRs)	<code>{{ADR_DIR}}</code>	Why we made key decisions

6. Coding Standards Reference

Standards document: [coding-standards.md](#)

Quick reference:

- **Commit messages:** Conventional Commits format (`feat:`, `fix:`, `docs:`, etc.)
- **Branch naming:** `{{BRANCH_FORMAT}}` (e.g., `feature/TICKET-123-short-description`)
- **PR size:** Aim for `< {{PR_SIZE}}` lines changed per PR
- **Testing requirement:** All PRs require tests for new code

7. Git Workflow & Branching Strategy

```
# Create a feature branch
git checkout -b feature/{{TICKET}}-description

# Make changes, then commit
git add {{FILES}}
git commit -m "feat({{SCOPE}}): describe what you did"

# Push your branch
git push origin feature/{{TICKET}}-description

# Open a PR via GitHub/GitLab UI
```

Branching strategy: `{{STRATEGY}}` — see [CI/CD Pipeline](#)

PR requirements:

1. Title follows format: `type(scope): description`
2. PR description filled in (template auto-loaded)
3. Tests included for new code
4. CI pipeline passing before requesting review
5. At least `{{REVIEW_COUNT}}` reviewer(s) approved

8. First PR Checklist

Your first PR goal: A small, low-risk change to confirm everything works end-to-end.

- Development environment running locally
- Can run the full test suite: `{{TEST_CMD}}`
- All tests passing locally

- First branch created following naming convention
- Change made (start small — a doc fix or minor improvement is perfect)
- Tests written for any logic change
- PR opened with filled description
- CI pipeline passing on your PR
- Review requested from your onboarding buddy
- First PR merged ☐☐

9. Team Rituals & Meetings

Meeting	Frequency	When	Purpose	Required
Daily standup	Daily	{{TIME}}	Status, blockers	Yes
Sprint planning	{{SPRINT_FREQ}}	{{TIME}}	Plan upcoming sprint	Yes
Sprint review	{{SPRINT_FREQ}}	{{TIME}}	Demo completed work	Yes
Retrospective	{{SPRINT_FREQ}}	{{TIME}}	Process improvement	Yes
Architecture review	Monthly	{{TIME}}	Tech decisions	Optional
Social / team coffee	{{SOCIAL_FREQ}}	{{TIME}}	Team building	Encouraged

10. Communication Channels

Channel	Platform	Purpose
#{{GENERAL_CHANNEL}}	{{PLATFORM}}	General team communication
#{{DEV_CHANNEL}}	{{PLATFORM}}	Developer discussions, code help
#{{INCIDENTS_CHANNEL}}	{{PLATFORM}}	Incident response
#{{DEPLOYMENTS_CHANNEL}}	{{PLATFORM}}	Deployment notifications
Direct message	{{PLATFORM}}	Quick questions to teammates
{{EMAIL_LIST}}	Email	Formal announcements

Rule: Default to public channels over DMs. Questions asked in channels help the whole team.

11. Key Contacts

Role	Name	Contact	Help With
Onboarding Buddy	{{BUDDY}}	{{CONTACT}}	Everything in the first week
Tech Lead	{{TECH_LEAD}}	{{CONTACT}}	Technical decisions, architecture
Engineering Manager	{{ENG_MGR}}	{{CONTACT}}	Career, process, team
DevOps / Platform	{{DEVOPS}}	{{CONTACT}}	Infrastructure, deployment, CI
QA Lead	{{QA_LEAD}}	{{CONTACT}}	Testing questions

Related Documents

- [Local Development Setup](#)
- [Coding Standards](#)
- [Developer Offboarding Guide](#)

Approval

Role	Name	Date	Signature
Author			
Reviewer			
Approver			

Revision #3

Created 2026-02-24 14:54:28 UTC by John

Updated 2026-05-25 07:34:50 UTC by John