Development · Contributing

Generated Tue, Sep 22, 2026 12:12 PM · Single page

docs/development/contributing.md

Contributing

Public contribution policy for this multi-tenant CRM.

Short answer: outside contributors are welcome **via pull request**. They are not granted production access, Super Admin, or direct push to `main`.

By opening a pull request you agree to this policy, including the intellectual property terms.

Who can contribute

Who	What they may do
Anyone	Open an issue or a pull request with a **specific** bug fix or feature
Trusted teammates	Same as above; write access to a feature branch is optional and still PR-only into `main`
Maintainer (repo owner)	Reviews, merges, deploys, and holds production credentials

Contributions are accepted when they are scoped, reviewed, and safe for a multi-tenant product. Vague “give me access and I’ll help with everything” requests are declined.

You do **not** need to be a collaborator on GitHub to contribute. Fork the repository, make a branch, and open a PR.

How we accept work (PR-only)

1. Fork (external) or create a feature branch from up-to-date `main`.
   Example: `cursor/short-description-a8c9`
2. Implement with tests.
3. Run Pint + tests locally (see Before you push).
4. Open a **pull request against `main`**.
5. Address review. The maintainer merges when ready.
6. Ensure CI passes once workflows exist (CI/CD).

Not allowed

- Direct pushes to `main`
- Force-pushes to shared branches
- Merging your own PR unless the maintainer has explicitly said so
- Asking for admin, maintain, or production deploy rights as a condition of helping

Write access to this repository, if ever granted, is only for pushing **feature branches**. It is not permission to change production or skip review.

What contributors never get

This CRM stores tenant customer data, conversations, and channel secrets (WhatsApp, Meta, webhooks). Contributors work on **code**, not the live system.

Do **not** request or accept:

- Railway / hosting / SSH / database credentials
- Production or staging `.env` files
- Meta App Secret, WhatsApp tokens, webhook signing secrets, mail or payment keys
- Super Admin access on the live product
- Tenant logins, customer exports, or database dumps
- Impersonation of real companies

Use local setup from Installation and `.env.example`. Seeded / demo data is for local development only.

If you accidentally receive a secret, **do not** put it in a PR, issue, or chat log. Tell the maintainer privately so it can be rotated.

Intellectual property

The repository owner retains all rights to this CRM. There is no inbound license that makes your patch automatically open source on your terms.

By submitting a contribution (pull request, patch, or issue that includes code or docs) you confirm that:

1. The work is yours, or you have permission to submit it, and it does not include secrets or someone else's proprietary code.
2. You grant the repository owner a **perpetual, worldwide, irrevocable, royalty-free** license to use, copy, modify, merge, publish, distribute, sublicense, and **relicense** the contribution as part of this CRM, in source or compiled form, for any purpose.
3. Git history may keep your name as author; the owner is not required to credit you in the product UI or marketing.
4. The contribution does **not** entitle you to payment, equity, revenue share, or production access unless a **separate written agreement** says otherwise.
5. You can be asked to change or remove a contribution during review; the owner may decline any PR for any reason.

If you cannot agree to these terms, do not open a pull request.

Security and tenancy

This is a multi-tenant app. A bad change can leak Company A's data to Company B.

- Follow Coding standards (CompanyScope, policies, permissions).
- Cover happy path **and** cross-tenant 403/404 in feature tests.
- Never commit `.env`, tokens, or `docs/demo-credentials.local.md`.
- Keep Super Admin routes in `routes/superadmin.php`.
- Do not use `withoutCompanyScope()` unless the code truly needs it (webhooks, platform tools), and set company context before tenant writes.

See Multi-tenancy and RBAC.

Before you push

```
vendor/bin/pint
php artisan test
# if permissions config changed:
php artisan permissions:sync
```

PR checklist

- ☐ Tenant isolation considered (CompanyScope / policies)
- ☐ Permissions updated + synced if needed
- ☐ Feature tests for happy path + 403/404 cross-tenant
- ☐ No secrets committed (`.env`, tokens)
- ☐ Docs updated under `/docs` when behavior changes
- ☐ Migrations are MySQL-safe (index name lengths)
- ☐ You agree to this contribution policy (access + IP)

Channels / Meta changes

- Document setup steps in `docs/channels/`
- Fake Graph HTTP in tests
- Note queue worker requirements

Super Admin vs tenant

- Keep Super Admin routes in `routes/superadmin.php`
- Do not leak platform queries into tenant controllers without `withoutCompanyScope` intent

Commit messages

Prefer clear imperative subjects:

- `Add Inbox UI with WhatsApp reply support`
- `Fix company settings edit method`

Related

- Coding standards
- Roadmap
- Installation