

PSScript Support and Operations

This guide defines how to get help, how issues are triaged, and what operational checks should be performed before escalation.

Support entry points

Channel	Purpose	Best for
GitHub Issues	Bug reports and feature requests	Product issues and reproducible defects
Docs and Training	Self-service guides	Onboarding, workflows, and troubleshooting
Logs and health checks	Operational diagnostics	Service availability and performance

Before you open an issue

- Confirm the problem is reproducible
- Capture steps, expected behavior, and actual behavior
- Include logs from backend and AI service
- Include screenshots if UI related

Issue triage checklist

Step	Owner	Outcome
Reproduce	Reporter	Clear reproduction steps
Classify	Maintainer	Bug, feature, or support
Assign	Maintainer	Owner and priority set

Step	Owner	Outcome
Verify fix	Maintainer	Regression or unit test added

Operational checks

Health endpoints

- Backend: `http://localhost:4000/health`
- Backend API: `http://localhost:4000/api/health`
- AI service: `http://localhost:8000/health`

Log locations

- Backend logs: `src/backend/logs/`
- AI service logs: `src/ai/logs/` (if configured)
- Docker logs: `docker-compose logs -f`

Common recovery actions

```
# Restart all services
./restart-all.sh

# Restart backend only
./restart-backend.sh
```

Severity guide

Severity	Description	Example
Sev 1	Production down or data loss	Cannot log in, data missing
Sev 2	Critical workflow blocked	Upload or analysis failing
Sev 3	Degraded experience	Slow responses, partial UI
Sev 4	Cosmetic or enhancement	UI polish, copy changes

Security reporting

If you believe you have found a security issue, do not open a public issue. Share details privately with the project owner.

Reference docs

- `docs/GETTING-STARTED.md`
- `docs/training-suite/README.md`
- `docs/training-suite/TRAINING-GUIDE.md`

Generated 2026-01-12 11:31 UTC