

MuscleMap

Architecture Documentation

Auto-generated

2026-01-06

Contents

1 Overview	2
1.1 Design Principles	2
2 Technology Stack	2
3 Packages	2
4 Frontend Pages	2
5 API Endpoints	3
6 Data Flow	4
7 Build Process	4
8 Scripts	4

1 Overview

MuscleMap is a cross-platform fitness tracking application with real-time muscle visualization. The architecture follows a modular, layered approach with clear separation of concerns.

1.1 Design Principles

- **Single Source of Truth:** PostgreSQL is the only data store
- **Cross-Platform:** One codebase serves web and mobile
- **Modular:** Clear package boundaries
- **Type-Safe:** TypeScript throughout

2 Technology Stack

Layer	Technology	Purpose
Frontend Web	React + Vite + TailwindCSS	Single-page application
Frontend Mobile	React Native + Expo	iOS & Android apps
API Server	Fastify + TypeScript	REST/GraphQL API
Database	PostgreSQL	Primary data store
Cache	Redis (optional)	Session/query caching
Reverse Proxy	Caddy	HTTPS termination

Table 1: Technology Stack

3 Packages

The monorepo contains 5 packages:

Package	Description
@musclemap/client	Shared client-side business logic for MuscleMap
@musclemap/core	Shared domain types and utilities for MuscleMap
@musclemap/plugin-sdk	SDK for building MuscleMap plugins
@musclemap/shared	Shared utilities and constants for MuscleMap apps
@musclemap/ui	Shared cross-platform UI components for MuscleMap

Table 2: Monorepo Packages

4 Frontend Pages

The application contains 34 pages:

Page	Protected	Description
AdminController	No	AdminController page
AdminIssues	Yes	Admin Issues Page Admin dashboard for m
CommunityDashboard	Yes	CommunityDashboard Page Comprehensive c
Competitions	No	Competitions page
Credits	No	Credits page

Page	Protected	Description
Dashboard	Yes	Dashboard - MuscleMap Liquid Glass Design
Design	No	Design Page Showcases MuscleMap's design
DesignSystem	No	Design System Showcase Demonstrates the
DevUpdates	No	Dev Updates Page Development updates, a
Docs	No	Docs page
Exercises	No	Exercises page
Features	No	Features Page Showcases MuscleMap features
HighFives	No	HighFives page
IssueDetail	Yes	Issue Detail Page Single issue view with
Issues	Yes	Issues Page Bug and issue tracker with:
Journey	Yes	Journey page
Landing	No	Landing page
Locations	No	Locations page
Login	Yes	Login page
Messages	No	Messages page
MyIssues	Yes	My Issues Page User's submitted issues:
NewIssue	Yes	New Issue Page Create a new bug report,
Onboarding	Yes	Onboarding page
Privacy	No	Privacy Policy Page Required for App Store
Profile	Yes	Profile page
Progression	No	Progression page
Roadmap	No	Roadmap Page Public roadmap showing: -
Science	No	Science Page Explains the science behind
Settings	No	Settings page
Signup	Yes	Signup page
SkinsStore	No	SkinsStore page
Technology	No	Technology Stack Page Showcases MuscleMap
Wallet	No	Wallet page
Workout	No	Workout page

Table 3: Frontend Pages

5 API Endpoints

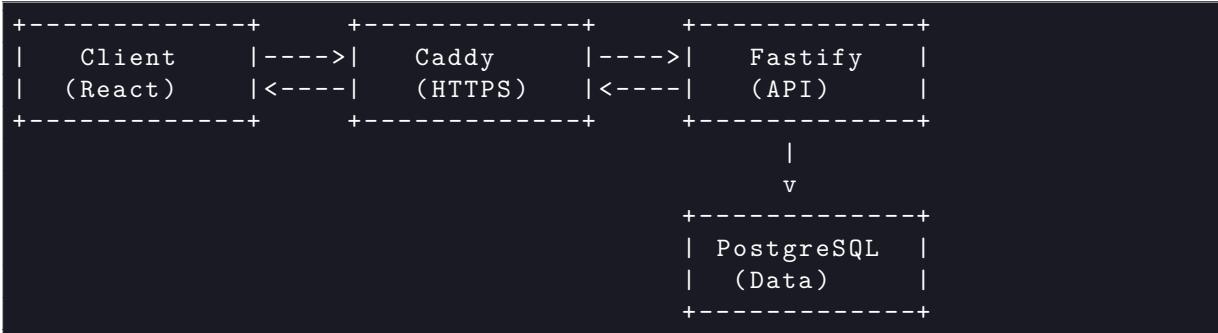
The API exposes 112 endpoints:

Method	Path	Handler
GET	/__routes	misc
GET	/admin/issues	issues
POST	/admin/issues/bulk	issues
GET	/alternatives/low-impact	misc
GET	/alternatives/seated	misc
GET	/archetypes	journey
GET	/archetypes/:id/levels	journey
POST	/archetypes/select	journey
POST	/auth/login	auth
GET	/auth/me	auth
GET	/auth/me/capabilities	auth

Method	Path	Handler
POST	/auth/register	auth
GET	/community/feed	community
GET	/community/monitor	community
GET	/community/percentile	community
GET	/community/presence	community
POST	/community/presence	community
GET	/community/stats	community
GET	/community/ws	community
GET	/competitions	misc
GET	/competitions/:id	misc
GET	/credits/balance	economy
GET	/economy/actions	economy
GET	/economy/balance	economy
POST	/economy/charge	economy
GET	/economy/history	economy
GET	/economy/pricing	economy
GET	/economy/transactions	economy
GET	/economy/wallet	economy
GET	/entitlements	misc

Table 4: API Endpoints

6 Data Flow



7 Build Process

```
# Build order
pnpm build:packages    # shared -> core -> plugin-sdk -> client -> ui
pnpm build:api          # API server
pnpm build               # Frontend (Vite)

# Full deployment
./deploy.sh "commit message"
```

8 Scripts

Script	Description
<code>deploy-branch.sh</code>	# deploy-branch.sh - Full deployment: commit, push
<code>deploy.sh</code>	Deployment helper for MuscleMap
<code>errors.sh</code>	errors.sh
<code>generate-docs.cjs</code>	MuscleMap Documentation Generator Analyzes the co
<code>generate-icons.cjs</code>	App Icon Generator for MuscleMap Mobile App Gener
<code>logs.sh</code>	Unified Log Viewer for MuscleMap
<code>maintain.sh</code>	— preflight —
<code>merge-all.sh</code>	# merge-all.sh - Merge all worktree branches into
<code>prepare-app-store.cjs</code>	MuscleMap App Store Preparation Script Automates
<code>production-deploy.sh</code>	# production-deploy.sh - Deploy script for MuscleM
<code>publish-app.sh</code>	# MuscleMap App Publishing Script
<code>repo-cleanup.sh</code>	If caller sets DRY_RUN/APPLY explicitly, we do not
<code>reset-devbox.sh</code>	— toggles (set by menu) —
<code>test.sh</code>	Create a competition to get an ID (avoid hardcoded
<code>tidy-root-js.sh</code>	Collect file basenames referenced by systemd units

Table 5: Available Scripts