Asperguide Backend - Module Dependency Graph

Layer 6: Entry Point
- server_main.py

Layer Isolation:
Each layer only depends on layers below it. This creates a clean separation of concerns and prevents circular dependencies.

RuntimeManager Pattern:
Services use RuntimeManager.get() to access dependencies at runtime, avoiding hard imports and enabling lazy initialization.

FinalSingleton Pattern:
Most services extend FinalSingleton to ensure single instance and controlled lifecycle via RuntimeManager.

Layer 5: Application Logic
- Server
- EndpointManager
- DocumentationHandler
- PathManager

Layer 4: Service Components
- MailManagement
- OAuthAuthentication
- Crons
- BackgroundTasks

Layer 3: Data Access
- Bucket
- SQL
- Redis

Layer 2: Utility Components
- ServerHeaders
- BoilerplateIncoming
- BoilerplateResponses
- PasswordHandling
- HttpCodes
- ServerManagement

Layer 1: Core Infrastructure
- FinalClass
- FinalSingleton
- RuntimeManager
- RuntimeControl

Layer 0: External Dependencies
- Boto3
- SQLAlchemy
- FastAPI
- Redis-py
- APScheduler
- Uvicorn