

WEB PORTAL - TÀI LIỆU GIẢI THÍCH CHI TIẾT

(Kiến trúc, hàm, quy trình xử lý)

Phiên bản: 1.0 – Clean Architecture (.NET 8, Blazor Server, SQL Server, Redis, gRPC)

1. Mục tiêu và phạm vi

Tài liệu này giải thích chi tiết cách solution WebPortal được triển khai theo yêu cầu hệ thống: quản lý danh mục link theo Category, cá nhân hóa theo từng user (Pin/Unpin, sắp xếp, ViewMode), phân quyền RBAC theo Role Prefix, cache Redis để tăng tốc, UI Blazor Server với thao tác kéo-thả, tìm kiếm nhanh hotkey '/', và cơ chế bảo mật URL.

Yêu cầu được đáp ứng

- Tập trung hóa: quản lý nhiều link dịch vụ tại 1 nơi.
- Cá nhân hóa: mỗi user có Workspace riêng (favorites + order + view mode).
- RBAC: link chỉ hiển thị nếu role của user match RolePrefix.
- Bảo mật: chặn URL độc hại (javascript:, data:), link ngoài mở tab mới với rel="noopener noreferrer".
- Hiệu năng: cache catalog + workspace qua Redis (fallback Memory).
- Stack: .NET 8, Blazor Server, SQL Server, Redis, gRPC.

2. Kiến trúc solution và phụ thuộc

Solution gồm 6 project chính (đúng theo phân tầng):

Project	Vai trò	Tham chiếu
WebPortal.Domain	Entity/Enum/Guard (luật domain)	Không phụ thuộc project khác
WebPortal.Application	Use case + Abstractions (Interfaces)	Tham chiếu Domain
WebPortal.Infrastructure	EF Core mapping + Repository + Redis cache	Tham chiếu Application + Domain
WebPortal.Web	Blazor Server UI (Portal + Admin)	Tham chiếu Application + Infrastructure
WebPortal.Grpc.Contracts	Định nghĩa gRPC (.proto) + generate stubs	Độc lập

WebPortal.Grpc	gRPC host (PortalGrpc/AdminGrpc)	Tham chiếu Application + Infrastructure + Contracts
----------------	-------------------------------------	---

Nguyên tắc đặt Interface:

Interfaces (IUserRepository, IWorkspaceRepository, ICacheService, IUserContext...) đặt tại Application để đảm bảo Dependency Inversion: Application định nghĩa nhu cầu, Infrastructure chỉ implement.

3. Mapping database (SQL Server)

DB schema bám theo script bạn cung cấp: Users, Categories, Links, FavoriteLinks, LinkSequence, CategorySequence.

3.1 Ý nghĩa từng bảng

- Users: Thông tin user + ViewMode (Grid/List). UpdatedAt được update ở code khi thay đổi.
- Categories: Nhóm link. Có Priority (thứ tự mặc định) và IsActive để soft delete.
- Links: Metadata của link:
Name/Url/Icon/Color/Tags/Priority/RolePrefix/IsActive/CategoryId. Có check constraint chặn javascript:/data:.
- FavoriteLinks: Pin/Unpin per user (PK: UserId+LinkId). PinnedAt phục vụ sort/hiển thị.
- LinkSequence: Order tùy chỉnh của user cho link (PK: UserId+LinkId, có OrderIndex).
- CategorySequence: Order tùy chỉnh của user cho category (PK: UserId+CategoryId, có OrderIndex).

3.2 EF Core mapping nằm ở đâu?

File/Folder chính:

- WebPortal.Infrastructure/Persistence/WebPortalDbContext.cs
- WebPortal.Infrastructure/Persistence/*Configuration.cs

4. Luồng xử lý chính (Use case)

4.1 Load Portal – GetPortalAsync(searchText)

1. Đảm bảo user tồn tại trong DB: IUserRepository.EnsureUserAsync(userId, userName,...).
2. Lấy catalog Categories/Links từ cache: key catalog:categories:active và catalog:links:active (TTL 5 phút).
3. Lấy workspace (favorites + sequences) từ cache: key workspace:{userId} (TTL 5 phút).
4. Filter security: bỏ link bị block bởi UrlSecurityGuard (javascript:, data:).
5. RBAC filter: chỉ giữ link có role match RolePrefix (RolePrefixGuard.IsMatchedByPrefix).
6. Áp dụng search: match theo Link.Name, Link.Tags hoặc Category.Name.

7. Sắp xếp Category: ưu tiên order tùy chỉnh (CategorySequence), nếu không có thì theo Category.Priority.
8. Sắp xếp Link trong từng Category: ưu tiên order tùy chỉnh (LinkSequence), nếu không có thì theo Link.Priority.
9. Tạo output PortalModel gồm: ViewMode, PinnedLinks (favorites), Categories + Links.

Hàm nằm tại: WebPortal.Application/Services/PortalService.cs

4.2 Pin/Unpin

Pin:

10. Validate: link tồn tại, IsActive=true, URL không bị block.
11. WorkspaceRepository.PinAsync(UserId, LinkId) -> insert FavoriteLinks nếu chưa có.
12. Invalidate cache workspace:{userId} để lần load sau cập nhật.

Unpin:

13. WorkspaceRepository.UnpinAsync(UserId, LinkId) -> delete FavoriteLinks nếu có.
14. Invalidate cache workspace:{userId}.

4.3 Đổi thứ tự Category (Drag & Drop)

UI Blazor cập nhật ngay (optimistic), sau đó gọi server lưu:

15. Người dùng kéo Category A thả vào vị trí Category B.
16. UI reorder list Categories trong bộ nhớ (optimistic update).
17. Gọi IPortalService.SetCategoryOrderAsync(listCategoryIds).
18. Server lưu CategorySequence (UserId+CategoryId -> OrderIndex).
19. Nếu lỗi: UI rollback (load lại portal và hiển thị error).

4.4 Đổi thứ tự Link trong Category (Drag & Drop)

20. Người dùng kéo Link trong cùng Category để đổi thứ tự.
21. UI reorder list Links (optimistic).
22. Gọi IPortalService.SetLinkOrderAsync(listLinkId).
23. Server lưu LinkSequence (UserId+LinkId -> OrderIndex).

4.5 Đổi ViewMode (Grid/List)

24. User bấm toggle Grid>List.
25. UI cập nhật ngay (optimistic).
26. Gọi IPortalService.SetViewModeAsync(mode).
27. Server update Users.ViewMode và Users.UpdatedAt.

4.6 Admin CRUD Category/Link

Admin UI nằm ở:

- WebPortal.Web/Pages/Admin/Categories.razor

- WebPortal.Web/Pages/Admin/Links.razor

Các rule quan trọng:

- Create/Update Link luôn validate URL chống javascript:/data: (Domain guard + DB check constraint).
- Upsert xong sẽ invalidate cache catalog (ActiveCategories/ActiveLinks) để portal load lại ngay.

5. RBAC và bảo mật

5.1 RBAC Role Prefix

Quy tắc: user nhìn thấy link nếu có ít nhất 1 role bắt đầu bằng RolePrefix của link. Ví dụ RolePrefix="SMS_" thì role "SMS_ADMIN" được phép.

Hàm: WebPortal.Domain/Guards/RolePrefixGuard.cs

5.2 Bảo mật URL và mở tab mới

Chặn URL độc hại bằng 2 lớp:

- 1) Domain guard + Application validation (khi admin tạo/sửa link)
- 2) DB check constraint CK_Links_Url_Security (chặn javascript:/data:)

Tất cả link ngoài được render với target="_blank" và rel="noopener noreferrer" để tránh tab-nabbing.

6. Cache (Redis) và hiệu năng

Cache key chính:

- catalog:categories:active (TTL 5 phút)
- catalog:links:active (TTL 5 phút)
- workspace:{userId} (TTL 5 phút)

Cache implementation: WebPortal.Infrastructure/Caching/HybridCacheService.cs. Nếu cấu hình ConnectionStrings:Redis có giá trị, system dùng Redis; nếu không, fallback MemoryCache.

7. gRPC (internal communication)

Project WebPortal.Grpc cung cấp 2 service:

- PortalGrpc: GetPortal/Pin/Unpin/SetViewMode/SetOrder
- AdminGrpc: List/Upsert/SetActive

Proto: WebPortal.Grpc.Contracts/Protos/portal.proto

Dev headers (nếu chưa cấu hình auth): x-user-id, x-user-name, x-roles

8. Hướng dẫn chạy local

28. Tạo DB bằng script SQL của bạn (WebPortalDB5).
29. Cập nhật ConnectionStrings:WebPortalDb trong WebPortal.Web/appsettings.json.
30. (Tuỳ chọn) bật Redis: set ConnectionStrings:Redis = "localhost:6379".
31. Chạy WebPortal.Web (Blazor Server) để xem UI.
32. Chạy WebPortal.Grpc (gRPC) nếu cần tích hợp nội bộ.

9. Lỗi thường gặp và cách xử lý

- Không connect được DB: Kiểm tra connection string, SQL Server instance, quyền truy cập.
- Insert Link bị lỗi check constraint: URL chứa javascript: hoặc data: (bị chặn).
- Không thấy link: User thiếu role phù hợp với RolePrefix của link.
- Kéo-thả không lưu: Kiểm tra log server và bảng LinkSequence/CategorySequence đã được update chưa.
- UpdatedAt không đổi: Trong DB default chỉ áp cho INSERT. Code đã update Users.UpdatedAt khi đổi ViewMode; các entity khác không có UpdatedAt trong schema hiện tại.