

Router
- routes - request
+ register(string \$method, string \$route, string \$controller, string \$action): void + match(): ?array + routerRequest(\$controller, \$action): void

Request
- get - post
+ sanitizeTextArea(string \$data): string + sanitizeString(string \$data): string + sanitizeEmail(string \$data): ?string + sanitizeLogin(string \$data): string + sanitizeInteger(int \$value): ?int

Session
- session
+ getSession(): ?array + setSession(array \$addedArray): self + remove(string \$session): self + destroy(): void

Auth
- request - session - userRepo
+ user(): ?User + isLoggedIn(): bool + isAdmin(\$userId): bool + isSuperAdmin(\$userId): bool + generateToken(): string + checkToken(?string \$token): bool

View
- twig - loader
+ render(string \$view, array \$params = []): void + renderMail(string \$view, string \$block, array \$params = []): string

ErrorController
- renderer - session - auth
+ show404(): void

AccountController
- renderer - postRepo - postManager - commentRepo - commentManager - userRepo - userManager - session - auth - config
+ showLoginPage(): void + showSignInPage(): void + contactForm(Request \$request): void + loginForm(Request \$request): void + logout(): void + signInForm(Request \$request): void + activate(Request \$request): void + resendMail(): void

PostController
- renderer - postRepo - postManager - commentRepo - commentManager - session - auth
+ home(): void + showSinglePost(Request \$request): void + showPostsPage(Request \$request): void + addComment(Request \$request): void

AdminController
- renderer - postRepo - postManager - commentRepo - commentManager - userManager - userRepo - session - auth
+ showPostsManager(Request \$request): void + showEditPost(Request \$request): void + modifyPost(Request \$request): void + delete(Request \$request): void + showAddPost(): void + addPost(Request \$request): void + showCommentsManager(Request \$request): void + approve(Request \$request): void + refuse(Request \$request): void

SuperAdminController
- renderer - userRepo - userManager - session - auth
+ showUsersManager(Request \$request): void + promote(Request \$request): void + demote(Request \$request): void

PostManager
- userRepo - session
+ getSinglePost(int \$postId): ?Post + getSinglePostPager(int \$commentPage, int \$totalComments): array + getNextPostId(int \$postId): ?int + getPreviousPostId(int \$postId): ?int + getHomePagePosts(): array + getPostsPage(int \$offset, int \$limit): array + getPostsPagePager(int \$currentPage, int \$totalItems): array + getNumberOfPosts(): int + modifyPostContent(int \$postId, ?string \$title, ?string \$chapo, ?int \$authorId, ?string \$content): bool + createPost(?string \$title, ?string \$chapo, ?int \$authorId, ?string \$content): ?int + deletePost(int \$postId): bool

CommentManager
- userRepo - session
+ getApprovedComments(int \$postId, int \$offset, int \$commentsNumberLimit): array + getNotApprovedComments(int \$offset, int \$commentsNumberLimit): array + getNumberOfApprovedCommentsFromPost(int \$postId): int + getNumberOfApprovedComments(): int + getNumberOfNotApprovedComments(): int + getCommentsManagerPager(int \$commentPage, int \$totalComments): array + addCommentToPost(int \$postId, int \$authorId, ?string \$comment): bool + approveComment(int \$commentId): bool + refuseComment(int \$commentId): bool

UserManager
- userRepo - session
+ getUsersPage(int \$offset, int \$limit): array + getNumberOfUsers(): int + getUsersManagerPager(int \$userPage, int \$totalUsers): array + getAdminUsers(): array + promoteUser(int \$userId): bool + demoteUser(int \$userId): bool + login(?string \$login, ?string \$password): ?User + signIn(?string \$login, ?string \$password, ?string \$confirm, ?string \$email): ?array + signInUserFromToken(string \$previousToken): ?array

PostRepository
+ getPost(int \$postId): ?Post + getMostXRecentPosts(int \$postsNumberLimit): array + getPosts(int \$offset, int \$postsNumberLimit): array + countPosts(): int + getNextId(int \$postId): ?int + getPrevId(int \$postId): ?int + updatePost(int \$postId, string \$title, string \$chapo, int \$authorId, string \$content): bool + addPost(string \$title, string \$chapo, int \$authorId, string \$content): ?int + updatePost(int \$postId, string \$title, string \$chapo, int \$authorId, string \$content): bool + deleteOnePost(int \$postId): bool

CommentRepository
+ getCommentsFromPost(int \$postId, int \$offset, int \$commentsNumberLimit, int \$approved): array + getAllNotApprovedComments(int \$offset, int \$commentsNumberLimit): array + countComments(int \$approved): int + countNumberOfApprovedCommentsFromPost(int \$postId): int + insertCommentToPost(int \$postId, int \$authorId, string \$comment): bool + setCommentToApproved(int \$commentId): bool + deleteComment(int \$commentId): bool

UserRepository
+ getUser(int \$userId): User + countUsers(): int + getNonSuperAdminUsers(int \$offset, int \$limit): array + getAllAdminUsers(): array + updateUserType(int \$userId, string \$type): bool + checkLogin(string \$login, string \$password): ?User + userHasPosts(int \$userId): bool + userExists(string \$login): bool + emailExists(string \$email): bool + addUser(string \$login, string \$passwordHash, string \$email): ?int + insertToken(string \$token, int \$newUserId): bool + searchToken(string \$token): ?User + activateOneUser(string \$token): bool

Post
- id : int - title : string - chapo : string - content : string - created : datetime - lastUpdate : datetime - authorId : int

Comment
- id : int - content : string - approved : bool - created : datetime - lastUpdate : datetime - postId : int - authorId : int

User
- id : int - login : string - password : string - email : string - type : TypeProfile - activated : bool - token : string - created : datetime - lastUpdate : datetime

<enum>> TypeProfile
superadmin admin member

Database
- config
dbConnect(): PDO

Config
+ contactMail + serverUrl + dbName + dbUser + dbPassword