IAsyncAuthorizationFilter ApiKeyAuthorizationFilter - ApiKeyService : IApiKeyService {get;} + ApiKeyAuthorizationFilter(IApiKeyService apiKeyService) + OnAuthorizationAsync(AuthorizationFilterContext context) : async Task **FileUtilities** Services + FileExists(this string file) : bool + CreateFileIfNotExists(this string pathFileName, Encoding encoding) <<interface>>
IOxygenMeasurementService <<interface>> **IApiKeyService** IOxygenMeasurementSystemService LockObj : static readonly object + GetOxygenMeasurementByIdAsync(int id) : Task<OxygenMeasurementResponseDto?> + ValidateApiKey(int systemId, string apiKey) : Task<bool> + GetOxygenMeasurementSystemByIdAsync(int id) : Task<List<OxygenMeasurementSystemResponseDto?>> LogDirectory : static readonly string + AddOxygenMeasurementAsync(AddOxygenMeasurementRequestDto oxygenMeasurement) : Task<OxygenMeasurementResponseDto?> + AddOxygenMeasurementSystemAsync(AddOxygenMeasurementSystemRequestDto addRequestDto): Task<OxygenMeasurementSystemRequestDto?> + GetSystemNotificationAdvisors(int systemId) : Task<List<string>> + LogAsync(string message) : static async DirectoryUtilities + DirectoryExists(this string directory) : bool **ApiKeyService** + CreateDirectoryIfNotExists(this string directory) - oxygenDbContext : readonly IOxygenDbContext + ApiKeyService(IOxygenDbContext dbContext) OxygenMeasurementService OxygenMeasurementSystemService - oxygenDbContext : readonly IOxygenDbContext - oxygenDbContext : readonly IOxygenDbContext + OxygenMeasurementSystemService(IOxygenDbContext dbContext) + OxygenMeasurementService(IOxygenDbContext dbContext) DTO AddOxygenMeasurementSystemRequestDto OxygenMeasurementSystemResponseDto + SystemName : string {get; init} + Location : string {get; init;} + Id : int {get; init;} + SystemNotificationAdvisors : List<SystemNotificationAdvisorDto> {get; init;} + SystemName : String {get; set;} + Location : string {get; set;} + SystemNoificationAdvisors : List<SystemNotificationAdvisorResponseDto> {get; set;} + ApiKeyValue : string {get; init;} AddSystemNotificationAdvisorRequestDto OxygenMeasurementResponseDto + Email : string {get; init;} + Id : Int {get; set;} + OxygenValue : decimal {get; set;} + MeasurementTime : DateTime {get; set;} AddOxygenMeasurementRequestDto + OxygenMeasurementSystemId : int {get; set;} SystemNotificationAdvisorResponseDto + SystemName : string {get; set;} + OxygenValue : decimal {get; init;} + SystemLocation : string {get; set;} + Id : int {get; set;} + OxygenMeasurementSystemId : int {get; init;} + Email : string {get; set;} <<static>>
DbEntityToDtoResponseMapper GreaterThanZeroAttribute + ToResponse(this OxygenMeasurementSystem oxygenMeasurementSystem): OxygenMeasurementSystemResponseDto? + ToResponse(this SystemNotificationAdvisor systemNotificationAdvisor) : SystemNotificationAdvisorResponseDto + ToResponseList(this IEnumerable<SystemNotificationAdvisor> systemNotificationAdvisors) : List<SystemNotificationAdvisorResponseDto> + ToResponse(this OxygenMeasurement oxygenMeasurement) : OxygenMeasurementResponseDto? <<Controller>>
OxygenSystemController <<Controller>>
OxygenSystemController <<static>>
ApiKeyGenerator - SystemService : IOxygenMeasurementSystemService {get;} - oxygenMeasurementService : readonly IOxygenMeasurementService - MailHandler : MailHandler {get;} + GenerateNewApiKey(int keyLength) : string + OxygenSystemController(IOxygenMeasurementSystemService systemService) OxygenMail : OxygenMail {get; set;} + PostOxygenMeasurementSystem(AddOxygenMeasurementSystemRequestDto oxygenMeasurementSystemRequestDto): async Task<IActionResult> - ConfigLoader : SmtpConfigLoader {get; set;} + GetOxygenMeasurementSystem(int id) : async Task<lActionResult> <<interface>>
IOxygenDbContext + OxygenController(IOxygenMeasurementService oxygenMeasurementService, SmtpConfigLoader configLoader) + PostOxygenMeasurement(AddOxygenMeasurementRequestDto addOxygenMeasurement) : async Task<IActionResult> + OxygenMeasurements : DbSet<OxygenMeasurement> {get; set;} <<static>>
RequestToDbEntityMapper + GetOxygenMeasurementById(int id) : async Task<IActionResult> Microsoft.EntityFrameworkCore + ApiKeys : DbSet<ApiKey> {get; set;} + OxygenMeasurementSystems : DbSet<OxygenMeasurementSystem> {get; set;} + ToDbEntity(this AddOxygenMeasurementSystemRequestDto systemRequestDto) : OxygenMeasurementSystem DbContext + ToDbEntityList(this IEnumerable<SystemNotificationAdvisorDto> systemAdvisors) : List<SystemNotificationAdvisor> + SystemNotificationAdvisor : DbSet<SystemNotificationAdvisor> {get; set;} + ToDbEntity(this AddSystemNotificationAdvisorRequestDto addSystemNotificationAdvisorRequestDto) : SystemNotificationAdvisor + SaveChangesAsync(CancellationToken cancellationToken = default) SmtpConfigLoader SmtpConfig Mail library + ToDbEntity(this AddOxygenMeasurementRequestDto requestDto) : OxygenMeasurement - Configuration : IConfiguration {get;} + SmtpUser : string {get; set;} MailHandler OxygenMail ← 0..1→ + SmtpConfig : SmtpConfig? {get; set;} + SmtpPassword : string {get; set;} + SmtpHost : string {get; set;} + SmtpConfigLoader(IConfiguration configuration) OxygenDbContext + Port : int {get; set;} - LoadSmtpConfig() + OxygenDbContext(DbContextOptions<OxygenDbContext> options) GlobalExceptionMiddleware - next : readonly RequestDelegate - GetExceptionDetails(Exception exception) : static (HttpStatusCode, string) - HandleExceptionAsync : static async task - GlobalExceptionMiddleware(RequestDelegate) Entity framework Entities + InvokeAsync(HttpContext) : async task OxygenMeasurement + ApiKeyId : int {get; set;} Vi har brugt Entity framework og + ApiKeyValue : string {get; set;} + Id : int {get; set;} for at kunne lave tabler med entity framework skal man bruge + OxygenMeasurementSystem : virtual OxygenMeasurement {get; set;} + OxygenValue : decimal {get; set;} "Reference Navigation Property" (Entity framework) + MeasurementTime : DateTime {get; set;} + OxygenMeasurementSystemId : int {get; set;} ArgumentOutOfRangeException Entity framework (Reference + OxygenMeasurementSystem : OxygenMeasurementSystem {get; set;} navigation properrty) OxygenMeasurementSystem + Id : int {get; set} SystemNotificationAdvisor Exceptions + SystemName : string {get; set} + Zipcode : string {get; set} + Id : int {get; set;} NotFoundException ←1——0..\*→ + Location : string {get; set} + Email : string {get; set;} + ApiKeyId : int {get; set} + NotFoundException(string message) + OxygenMeasurementSystemId : int {get; set;} + ApiKey : virtual ApiKey {get; set} + OxygenMeasurementSystem : virtual OxygenMeasurementSystem BadRequestException + OxygenMeasurements : virtual ICollection<OxygenMeasurement> {get; set} + SystemNotificationAdvisors : virtual ICollection<SystemNotificationAdvisors> {get; set;} + BadRequestException(string message) CustomDbException + CustomDbException(string message) CustomArgumentOutOfRangeException + CustomArgumentOutOfRangeException (string paramName, object actualValue, string message)

Microsoft.AspNetCore.Mvc.Filters