Microsoft.AspNetCore.Mvc.Filters 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>>
IOxygenMeasurementSystemService <<interface>> **IApiKeyService** LockObj : static readonly object + GetOxygenMeasurementByIdAsync(int id) : Task<OxygenMeasurementResponseDto?> + GetOxygenMeasurementSystemByIdAsync(int id) : Task<List<OxygenMeasurementSystemResponseDto?>> + ValidateApiKey(int systemId, string apiKey) : Task<bool> LogDirectory : static readonly string + GetAllOxygenMeasurementsAsync(): Task<List<OxygenMeasurementResponseDto?>> + GetAllOxygenMeasurementSystemAsync(): Task<List<OxygenMeasurementSystemResponseDto?>> + AddOxygenMeasurementSystemAsync(AddOxygenMeasurementSystemRequestDto addRequestDto): Task<OxygenMeasurementSystemRequestDto?> + GetAllSystemOxygenMeasurementsAsync(int systemId) : Task<List<OxygenMeasurementResponseDto?>> + LogAsync(string message) : static async DirectoryUtilities + AddOxygenMeasurementAsync(AddOxygenMeasurementRequestDto oxygenMeasurement) : Task<OxygenMeasurementResponseDto?> + DirectoryExists(this string directory) : bool + GetSpecificAmountOfOxygenMeasurementsAsync(int systemId, int amount) : Task<List<OxygenMeasurementResponseDto?>> **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 OxygenMeasurementSystemsResponseDto OxygenMeasurementSystemResponseDto + SystemName : string {get; init} + Location : string {get; init;} + Id : int {get; set;} + Id : int {get; init;} + SystemNotificationAdvisors : List<SystemNotificationAdvisorDto> {get; init;} + SystemName : string {get; set;} + SystemName : String {get; set;} + Location : 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> + ToResponseList(this IEnumerable<OxygenMeasurement> dbMeasurements): List<OxygenMeasurementResponseDto?> + ToResponse(this OxygenMeasurement oxygenMeasurement): OxygenMeasurementResponseDto? + ToSimpleResponse(this OxygenMeasurementSystem oxygenMeasurementSystem) : OxygenMeasurementSystemResponseDto? + ToResponseList(this IEnumerable<OxygenMeasurementSystem> dbSystems) : List<OxygenMeasurementSystemResponseDto?> <<Controller>>
OxygenSystemController <<Controller>>
OxygenController **ApiKeyGenerator** - SystemService : IOxygenMeasurementSystemService {get;} - oxygenMeasurementService : readonly IOxygenMeasurementService + GenerateNewApiKey(int keyLength) : string + OxygenSystemController(IOxygenMeasurementSystemService systemService) + OxygenController(IOxygenMeasurementService oxygenMeasurementService) + CreateOxygenMeasurementSystem(AddOxygenMeasurementSystemRequestDto oxygenMeasurementSystemRequestDto): async Task<IActionResult> + GetOxygenMeasurementSystem(int id) : async Task<lActionResult> · CreateOxygenMeasurement(AddOxygenMeasurementRequestDto addOxygenMeasurement) : async Task<IActionResult> <<interface>> <<static>>
RequestToDbEntityMapper + GetOxygenMeasurementById(int id) : async Task<IActionResult> IOxygenDbContext + GetAllOxygenMeasurements() : async Task<IActionResult> + OxygenMeasurements : DbSet<OxygenMeasurement> {get; set;} + ToDbEntity(this AddOxygenMeasurementSystemRequestDto systemRequestDto): OxygenMeasurementSystem + GetAllOxygenMeasurements(int systemId) : async Task<IActionResult> Microsoft.EntityFrameworkCore + ApiKeys : DbSet<ApiKey> {get; set;} + ToDbEntityList(this IEnumerable<SystemNotificationAdvisorDto> systemAdvisors) : List<SystemNotificationAdvisor> GetSpecificAmountOfOxygenMeasurements(int systemId, int amount) : async Task<IActionResult> + OxygenMeasurementSystems : DbSet<OxygenMeasurementSystem> {get; set;} + ToDbEntity(this SystemNotificationAdvisorDto systemNotificationAdvisorDto) : SystemNotificationAdvisor DbContext + SystemNotificationAdvisor : DbSet<SystemNotificationAdvisor> {get; set;} + ToDbEntity(this AddOxygenMeasurementRequestDto requestDto) : OxygenMeasurement + SaveChangesAsync(CancellationToken cancellationToken = default) OxygenDbContext + OxygenDbContext(DbContextOptions<OxygenDbContext> options) OxygenMeasurementExtensions 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)