BAKI 2017-ci\_ il

Proqram təminatının arxitekturası

Aztelekom MMC

Kimin üçün proqram təminatı

v 1.0.0

**NİS**

|  |  |  |  |
| --- | --- | --- | --- |
| Versiya | Tarix(dd.mm.yyyy) | Redaktor | Şəhrlər |
| 001 | 20.06.2016 | İsmayıl İsmayılov | Nümunə sənədin birinci versiyası |
| 002 | 24.06.2016 | Cəbrayıl Hüseynov | Buraxılma prosesi bölməsinə şərhlər |
| 003 | 26.06.2016 | İsmayıl İsmayılov | Buraxılma prosesi bölməsinin işlənilməsi |
| 004 | 30.06.2016 | İsmayıl İsmayılov | İnstallation çıxarldı |

Mündəricat

[**Məqsəd və proqramın tətbiqi şərtləri** 1](#_Toc493770270)

[**Proqramın xüsusiyyətləri** 1](#_Toc493770271)

[Server 1](#_Toc493770272)

[Token 1](#_Toc493770273)

[{Metodlar sadalanır} 2](#_Toc493770274)

[Date 5](#_Toc493770278)

[Client 6](#_Toc493770279)

[İstifadəçi interfeysi 7](#_Toc493770280)

[Etibarlılıq və təhlükəsizlik 7](#_Toc493770281)

[**Proqramın buraxılma prosesi** 7](#_Toc493770282)

[Environment 7](#_Toc493770283)

[**Proqram mesajları** 7](#_Toc493770284)

# **Məqsəd və proqramın tətbiqi şərtləri**

**Məqsədi**:

{Burada hansı məqsədlə Program yaranır və kimin sifarişi ilə}

**Proqramın tətbiqi şərtləri**:

{Proqram AztelekomNet , İKT və İT dəstək üzrə şöbəsi tərəfindən qəbul edilib istifadə ediləcək}

# **Proqramın xüsusiyyətləri**

Proqramın ümumi iş prinsipi klient-server texnologiyadan istifadə olunur

## Server

**{Hansı progrtamm dil və əlavə paketlərdən istifadə olunur( frameworklar və kitabxanalar. Hansı Data Base istifadə olunur.}**

Proqramda proqramlaşdırma dili olaraq c#, Platform Asp.Net Web Api , DB MsSql istifadə olunur. Əlavə paketlər :  
Db connection üçün EntityFramework CodeFirst, mapping üçün Automapper, logging üçün Log4Net, Real time üçün SignalR, Gmail listener üçün İmapX, authentication ve authorization üçün oauth2 kitabxanalarından istifadə olunur .

**{ Serverda yerləşən metodlar və onlar baradə sıxcam məlumat }**

Proqram strukturunda çoxqatlı arxitekturadan(n-Tier architecture ) və kod dizayn model (design pattern) olaraq “Repository pattern” istifadə edilir. Proqram daxilində 6 proyekt (NİS.Operator, NIS.Service, NIS.Bl, NIS.DAL, NIS.Security, NIS.Utils) mövcuddur.

* **NIS.Operator**. Client tərəfə aid olan bütün kodlar bu proyektdə yazılır.
* **NIS.Service**. Web Api funksiyasin heyata kecirir. Clientdən gələn bütün sorğular bu proyekt vasitəsi ilə filtrlanıb qəbul olunur və NIS.BL proyektinə ötürülür və qayıdan dataya və statusa uyğun Clientə cavab qaytarir.
  + **Controllerler**
    - **Region**
      * **Models**
        + RegionViewModel

int Id

string Name

int? RegionPrefix

DateTime? CreatedDate

string Description

int? DefaultUserId

int? TotalCount

List<AreaDto> Area

* + - * + RegionUpdateModel

int Id

string Name

int? RegionPrefix

string Description

int? DefaultUserId

* + - * + RegionFindModel

int? Id

string Name

int? RegionPrefix

DateTime? CreatedDate

string Description

int? DefaultUserId

int? PageSize

int? PageNumber

* + - * + RegionDeleteModel

int Id

* + - * + RegionCreateModel

string Name

int? RegionPrefix

string Description

int? DefaultUserId

* + - * **Actions**
        + **GetAll**

**Request model**

Filter Filter

**Response model**

IEnumerable<RegionViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter Filter

**Response model**

Task<IEnumerable<RegionViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

RegionViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task<RegionViewModel>

* + - * + **Find**

**Request model**

RegionFindModel parameter

**Response model**

RegionViewModel

* + - * + **FindAsync**

**Request model**

RegionFindModel parameter

**Response model**

Task<RegionViewModel>

* + - * + **FindAll**

**Request model**

RegionFindModel parameter

**Response model**

ICollection<RegionViewModel>

* + - * + **FindAllAsync**

**Request model**

RegionFindModel parameter

**Response model**

Task<ICollection<RegionViewModel>>

* + - * + **Add**

**Request model**

RegionCreateModel parameter

**Response model**

RegionViewModel

* + - * + **AddAsync**

**Request model**

RegionCreateModel parameter

**Response model**

RegionViewModel

* + - * + **AddRangeAsync**

**Request model**

List<RegionCreateModel> parameter

**Response model**

Task<List<RegionViewModel>>

* + - * + **Update**

**Request model**

RegionUpdateModel parameter

**Response model**

RegionViewModel

* + - * + **UpdateAsync**

**Request model**

RegionUpdateModel parameter

**Response model**

Task<RegionViewModel>

* + - **Area**
      * **Models**
        + AreaViewModel

int Id

int RegionId

string Name

int? AreaPrefix

DateTime CreatedDate

string Description

int? TotalCount

RegionDto Region

List<NodeDto> Node

* + - * + AreaUpdateModel

int Id

int RegionId

string Name

int? AreaPrefix

string Description

* + - * + AreaFindModel

int? Id

int? RegionId

string Name

int? AreaPrefix

DateTime? CreatedDate

string Description

int? PageSize

int? PageNumber

* + - * + AreaDeleteModel

int Id

* + - * + AreaCreateModel

int RegionId

string Name

int? AreaPrefix

string Description

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable<AreaViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable<AreaViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

AreaViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task<AreaViewModel>

* + - * + **Find**

**Request model**

AreaFindModel parameter

**Response model**

AreaViewModel

* + - * + **FindAsync**

**Request model**

AreaFindModel parameter

**Response model**

Task<AreaViewModel>

* + - * + **FindAll**

**Request model**

AreaFindModel parameter

**Response model**

Icollection<AreaViewModel>

* + - * + **FindAllAsync**

**Request model**

AreaFindModel parameter

**Response model**

Task<Icollection<AreaViewModel>>

* + - * + **Add**

**Request model**

AreaCreateModel parameter

**Response model**

AreaViewModel

* + - * + **AddAsync**

**Request model**

AreaCreateModel parameter

**Response model**

Task<AreaViewModel>

* + - * + **AddRangeAsync**

**Request model**

List<AreaCreateModel> parameter

**Response model**

Task<List<AreaViewModel>>

* + - * + **Update**

**Request model**

AreaUpdateModelparameter

**Response model**

AreaViewModel

* + - * + **UpdateAsync**

**Request model**

AreaUpdateModel parameter

**Response model**

Task<AreaViewModel>

* + - **Node**
      * **Models**
        + NodeViewModel

int Id

int? AreId

string Name

string Description

int? TotalCount

AreaDto Area

* + - * + NodeUpdateModel

int Id

int? AreId

int? Name

string Description \

* + - * + NodeFindModel

int? Id

int? AreId

string Name

string Description

int? PageSize

int? PageNumber

* + - * + NodeDeleteModel

int Id

* + - * + NodeCreateModel

int? AreId

int? Name

string Description

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable< NodeViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable< NodeViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

NodeViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task< NodeViewModel>

* + - * + **Find**

**Request model**

AreaFindModel parameter

**Response model**

NodeViewModel

* + - * + **FindAsync**

**Request model**

NodeFindModel parameter

**Response model**

Task< NodeViewModel>

* + - * + **FindAll**

**Request model**

NodeFindModel parameter

**Response model**

Icollection< NodeViewModel>

* + - * + **FindAllAsync**

**Request model**

NodeFindModel parameter

**Response model**

Task<Icollection< NodeViewModel>>

* + - * + **Add**

**Request model**

AreaCreateModel parameter

**Response model**

NodeViewModel

* + - * + **AddAsync**

**Request model**

NodeCreateModelparameter

**Response model**

Task< NodeViewModel>

* + - * + **AddRangeAsync**

**Request model**

List< NodeCreateModel> parameter

**Response model**

Task<List< NodeViewModel>>

* + - * + **Update**

**Request model**

NodeUpdateModel parameter

**Response model**

NodeViewModel

* + - * + **UpdateAsync**

**Request model**

NodeUpdateModel parameter

**Response model**

Task< NodeViewModel>

* + - **Department**
      * **Models**
        + DepartmentViewModel

int Id

string Name

int? DepartmentPrefix

DateTime? CreatedDate

string Description

string DepartmentType

int? DefaultUserId

int? TotalCount

string Email { get; set;

int Phone

List<AreaDto> Area

* + - * + DepartmentUpdateModel

int Id

string Name

int? DepartmentPrefix

string Description

int? DefaultUserId

string Email

string DepartmentType

int Phone

* + - * + DepartmentFindModel

string DepartmentType

int? Id

string Name

int? DepartmentPrefix

DateTime? CreatedDate

string Email

int? Phone

string Description

int? DefaultUserId

int? PageSize

int? PageNumber

* + - * + DepartmentDeleteModel

int Id

* + - * + DepartmentCreateModel

string Name

int? DepartmentPrefix

string Description

int? DefaultUserId

string Email

int Phone

string DepartmentType

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable< DepartmentViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable< DepartmentViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

DepartmentViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task< DepartmentViewModel>

* + - * + **Find**

**Request model**

DepartmentFindModel parameter

**Response model**

DepartmentViewModel

* + - * + **FindAsync**

**Request model**

DepartmentFindModel parameter

**Response model**

Task< DepartmentViewModel>

* + - * + **FindAll**

**Request model**

DepartmentFindModel parameter

**Response model**

Icollection< DepartmentViewModel>

* + - * + **FindAllAsync**

**Request model**

DepartmentFindModel parameter

**Response model**

Task<Icollection< DepartmentViewModel>>

* + - * + **Add**

**Request model**

DepartmentCreateModel parameter

**Response model**

DepartmentViewModel

* + - * + **AddAsync**

**Request model**

DepartmentCreateModelparameter

**Response model**

Task< DepartmentViewModel>

* + - * + **AddRangeAsync**

**Request model**

List< DepartmentCreateModel> parameter

**Response model**

Task<List< DepartmentViewModel>>

* + - * + **Update**

**Request model**

DepartmentUpdateModel parameter

**Response model**

DepartmentViewModel

* + - * + **UpdateAsync**

**Request model**

DepartmentUpdateModel parameter

**Response model**

Task< DepartmentViewModel>

* + - **MainTask**
      * **Models**
        + TaskViewModel

int Id

int GeneratedUserId

int? CustomerRequestId

int? ExecutorUserId

int? InjuryTypeId

int? RealInjuryTypeId

DateTime CreatedDate

DateTime? EndDate

DateTime? StartDate

string Description

string Note

int? Priority

int? DepartmentId

int? TotalCount

int TaskStatusId

int? ProjectId

int? MainTaskId

UserDto GeneratedUser

UserDto ExecutorUser

RequestDto CustomerRequest

DepartmentDto Department

ICollection<TagDto> Tag

RatingDto Rating

ProjectDto Project

TaskStatusDto TaskStatus

ICollection<TaskDto> ChildTasks

ICollection<CommentDto> Comment

ICollection<AttachmentDto> Attachment

TaskDto ParentTask

InjuryTypeDto InjuryType

InjuryTypeDto RealInjuryType

List<TaskStatusHistoryDto> TaskStatusHistory

* + - * + TaskUpdateModel

public int Id { get; set; }

public int? CustomerRequestId { get; set; }

public int? ExecutorUserId { get; set; }

public int? InjuryTypeId { get; set; }

public int? RealInjuryTypeId { get; set; }

public string Description { get; set; }

public int? DepartmentId { get; set; }

public int? TaskStatusId { get; set; }

public int? ProjectId { get; set; }

public int? MainTaskId { get; set; }

public DateTime? StartDate { get; set; }

public DateTime? EndDate { get; set; }

public string Note { get; set; }

public int? Priority { get; set; }

public RatingUpdateModel Rating { get; set; }

public ICollection<TagCreateModel> Tag { get; set; }

* + - * + TaskFindModel

public bool AscId { get; set; } = false;

public int? Id { get; set; }

public int? GeneratedUserId { get; set; }

public int? CustomerRequestId { get; set; }

public int? ExecutorUserId { get; set; }

public int? InjuryTypeId { get; set; }

public int? RealInjuryTypeId { get; set; }

public string Description { get; set; }

public int? DepartmentId { get; set; }

public List<int> TaskStatusId { get; set; }

public int? ProjectId { get; set; }

public int? RatingId { get; set; }

public DateTime? StartDateFrom { get; set; }

public DateTime? StartDateTo { get; set; }

public DateTime? EndDateFrom { get; set; }

public DateTime? EndDateTo { get; set; }

public DateTime? CreatedDateFrom { get; set; }

public DateTime? CreatedDateTo { get; set; }

public string Note { get; set; }

public int? Priority { get; set; }

public int? PageSize { get; set; }

public int? PageNumber { get; set; }

* + - * + TaskDeleteModel

public int Id { get; set; }

* + - * + TaskCreateModel

public int? CustomerRequestId { get; set; }

public int? ExecutorUserId { get; set; }

public int? MainTaskId { get; set; }

public int? ProjectId { get; set; }

public int? InjuryTypeId { get; set; }

public int? RealInjuryTypeId { get; set; }

public string Description { get; set; }

public int? DepartmentId { get; set; }

public DateTime? StartDate { get; set; }

public string Note { get; set; }

public int? Priority { get; set; }

public ICollection<TagCreateModel> Tag { get; set; }

public ICollection<AttachmentCreateModel> Attachment { get; set; }

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable< TaskViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable< TaskViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

TaskViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task< TaskViewModel>

* + - * + **Find**

**Request model**

TaskFindModel parameter

**Response model**

TaskViewModel

* + - * + **FindAsync**

**Request model**

TaskFindModel parameter

**Response model**

Task< TaskViewModel>

* + - * + **FindAll**

**Request model**

TaskFindModel parameter

**Response model**

Icollection< TaskViewModel>

* + - * + **FindAllAsync**

**Request model**

TaskFindModel parameter

**Response model**

Task<Icollection< TaskViewModel>>

* + - * + **Add**

**Request model**

TaskCreateModel parameter

**Response model**

TaskViewModel

* + - * + **AddAsync**

**Request model**

TaskCreateModelparameter

**Response model**

Task< TaskViewModel>

* + - * + **AddRangeAsync**

**Request model**

List< TaskCreateModel> parameter

**Response model**

Task<List< TaskViewModel>>

* + - * + **Update**

**Request model**

TaskUpdateModel parameter

**Response model**

TaskViewModel

* + - * + **UpdateAsync**

**Request model**

TaskUpdateModel parameter

**Response model**

Task< TaskViewModel>

* + - **TaskStatus**
      * **Models**
        + TaskStatusViewModel

public string Name { get; set; }

public System.DateTime CreatedDate { get; set; }

public string Description { get; set; }

public int? TotalCount { get; set; }

public ICollection<TaskStatusHistoryDto> TaskStatusHistory { get; set; }

* + - * + TaskStatusUpdateModel

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

* + - * + TaskStatusFindModel

public int? Id { get; set; }

public string Name { get; set; }

public System.DateTime? CreatedDate { get; set; }

public string Description { get; set; }

public int? PageSize { get; set; }

public int? PageNumber { get; set; }

* + - * + TaskStatusDeleteModel

public int Id { get; set; }

* + - * + TaskStatusCreateModel

public string Name { get; set; }

public string Description { get; set; }

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable< TaskStatusViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable< TaskStatusViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

TaskStatusViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task< TaskStatusViewModel>

* + - * + **Find**

**Request model**

TaskStatusFindModel parameter

**Response model**

TaskStatusViewModel

* + - * + **FindAsync**

**Request model**

TaskStatusFindModel parameter

**Response model**

Task< TaskStatusViewModel>

* + - * + **FindAll**

**Request model**

TaskStatusFindModel parameter

**Response model**

Icollection< TaskStatusViewModel>

* + - * + **FindAllAsync**

**Request model**

TaskStatusFindModel parameter

**Response model**

Task<Icollection< TaskStatusViewModel>>

* + - * + **Add**

**Request model**

TaskStatusCreateModel parameter

**Response model**

TaskStatusViewModel

* + - * + **AddAsync**

**Request model**

TaskStatusCreateModelparameter

**Response model**

Task< TaskStatusViewModel>

* + - * + **AddRangeAsync**

**Request model**

List< TaskStatusCreateModel> parameter

**Response model**

Task<List< TaskStatusViewModel>>

* + - * + **Update**

**Request model**

TaskUStatuspdateModel parameter

**Response model**

TaskStatusViewModel

* + - * + **UpdateAsync**

**Request model**

TaskStatusUpdateModel parameter

**Response model**

Task< TaskStatusViewModel>

* + - **Report**
      * **Models**
        + **Request**

**Department**

RequestByDepartmentViewModel

**public int? Id { get; set; }**

**public string Name { get; set; }**

**public int? Value { get; set; }**

RequestDepartmentReport

**public List<RequestByDepartmentViewModel> Data { get; set; };**

**public int? TotalCount { get; set; }**

**RequestStatus**

RequestByRequestStatusViewModel

**public int? Id { get; set; }**

**public string Name { get; set; }**

**public int? Value { get; set; }**

RequestRequestStatusReport

**public List<RequestByRequestStatusViewModel> Data { get; set; ;**

**public int? TotalCount { get; set; }**

**RequestType**

RequestByRequestTypeViewModel

**public int? Id { get; set; }**

**public string Name { get; set; }**

**public int? Value { get; set; }**

RequestRequestTypeReport

**public List< RequestByRequestTypeViewModel> Data { get; set; }**

**public int? TotalCount { get; set; }**

* + - * + **Task**

**ExecutorUser**

TaskByExecutorUserViewModel

**public int? Id { get; set; }**

**public string Name { get; set; }**

**public int? Value { get; set; }**

TaskExecutorUserReport

**public List<TaskByExecutorUserViewModel> Data { get; set; }**

**public int? TotalCount { get; set; }**

**TaskStatus**

TaskByTaskStatusViewModel

**public int? Id { get; set; }**

**public string Name { get; set; }**

**public int? Value { get; set; }**

TaskTaskStatusReport

**public List<TaskByTaskStatusViewModel> Data { get; set;**

**public int? TotalCount { get; set; }**

* + - * **Actions**
        + **RequestByDepartment**

**Request model**

RequestFindModel parameters

**Response model**

RequestDepartmentReport

* + - * + **RequestByRequestType**

**Request model**

RequestFindModel parameters

**Response model**

RequestRequestTypeReport

* + - * + **RequestByRequestStatus**

**Request model**

RequestFindModel parameters)

**Response model**

RequestRequestStatusReport

* + - * + **TaskByTaskStatus**

**Request model**

TaskFindModel parameters

**Response model**

TaskTaskStatusReport

* + - * + **TaskByExecutorUser**

**Request model**

TaskFindModel parameters

**Response model**

TaskExecutorUserReport

* + - **CustomerRequest**
      * **Models**
        + RequestViewModel

public int Id { get; set; }

public int CustomerRequestTypeId { get; set; }

public int? DepartmentId { get; set; }

public int? RegionId { get; set; }

public string CreatedUserId { get; set; }

public string CustomerName { get; set; }

public int? CustomerNumber { get; set; }

public string Text { get; set; }

public DateTime? StartDate { get; set; }

public DateTime? CreatedDate { get; set; }

public string Description { get; set; }

public string ContactNumber { get; set; }

public int? TotalCount { get; set; }

public int SourceTypeId { get; set; }

public int RequestStatusId { get; set; }

public RequestTypeDto CustomerRequestType { get; set; }

public DepartmentDto Department { get; set; }

public RegionDto Region { get; set; }

public SourceTypeDto SourceType { get; set; }

public RequestStatusDto RequestStatus { get; set; }

public virtual UserDto CreatedUser { get; set; }

public virtual RatingDto Rating { get; set; }

public List<TaskDto> MainTask { get; set; }

public List<AttachmentDto> Attachment { get; set; }

* + - * + RequestUpdateModel

public int Id { get; set; }

public int CustomerRequestTypeId { get; set; }

public int? DepartmentId { get; set; }

public int? RegionId { get; set; }

public string CustomerName { get; set; }

public int? CustomerNumber { get; set; }

public string Text { get; set; }

public DateTime? StartDate { get; set; }

public string Description { get; set; }

public string ContactNumber { get; set; }

public int RequestStatusId { get; set; }

public int SourceTypeId { get; set; }

public List<AttachmentCreateModel> Attachment { get; set; }

* + - * + RequestFindModel

public int? Id { get; set; }

public int? CustomerRequestTypeId { get; set; }

public int? DepartmentId { get; set; }

public int? RegionId { get; set; }

public int? CreatedUserId { get; set; }

public string CustomerName { get; set; }

public int? CustomerNumber { get; set; }

public string Text { get; set; }

public DateTime? StartDateFrom { get; set; }

public DateTime? StartDateTo { get; set; }

public DateTime? CreatedDateFrom { get; set; }

public DateTime? CreatedDateTo { get; set; }

public string Description { get; set; }

public string ContactNumber { get; set; }

public int? SourceTypeId { get; set; }

public int? RequestStatusId { get; set; }

public int? PageSize { get; set; }

public int? PageNumber { get; set; }

* + - * + RequestDeleteModel

public int Id { get; set; }

* + - * + RequestCreateModel

public int CustomerRequestTypeId { get; set; }

public int? DepartmentId { get; set; }

public int? RegionId { get; set; }

public string CustomerName { get; set; }

public int? CustomerNumber { get; set; }

public string Text { get; set; }

public DateTime? StartDate { get; set; }

public string Description { get; set; }

public string ContactNumber { get; set; }

public int SourceTypeId { get; set; }

public List<AttachmentCreateModel> Attachment { get; set; }

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable< RequestViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable< RequestViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

RequestViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task< RequestViewModel>

* + - * + **Find**

**Request model**

RequestFindModel parameter

**Response model**

RequestViewModel

* + - * + **FindAsync**

**Request model**

RequestFindModel parameter

**Response model**

Task< RequestViewModel>

* + - * + **FindAll**

**Request model**

RequestFindModel parameter

**Response model**

Icollection< RequestViewModel>

* + - * + **FindAllAsync**

**Request model**

RequestFindModel parameter

**Response model**

Task<Icollection< RequestViewModel>>

* + - * + **Add**

**Request model**

RequestCreateModel parameter

**Response model**

RequestViewModel

* + - * + **AddAsync**

**Request model**

RequestCreateModelparameter

**Response model**

Task< RequestViewModel>

* + - * + **AddRangeAsync**

**Request model**

List< RequestCreateModel> parameter

**Response model**

Task<List< RequestViewModel>>

* + - * + **Update**

**Request model**

RequestUpdateModel parameter

**Response model**

RequestViewModel

* + - * + **UpdateAsync**

**Request model**

RequestUpdateModel parameter

**Response model**

Task<RequestViewModel>

* + - **CustomerRequestType**
      * **Models**
        + RequestTypeViewModel

public int Id { get; set; }

public int? DepartmentId { get; set; }

public string Name { get; set; }

public int ExecutionDay { get; set; }

public string Description { get; set; }

public DateTime CreatedDate { get; set; }

public int? TotalCount { get; set; }

public int SourceTypeId { get; set; }

public int? ParentRequestTypeId { get; set; }

public virtual RequestTypeDto ParentRequestType { get; set; }

public DepartmentDto Department { get; set; }

public SourceTypeDto SourceType { get; set; }

public virtual ICollection<RequestTypeDto> ChildRequestTypes { get; set; }

* + - * + RequestTypeUpdateModel

public int Id { get; set; }

public int? DepartmentId { get; set; }

public string Name { get; set; }

public int ExecutionDay { get; set; }

public string Description { get; set; }

public int SourceTypeId { get; set; }

public int? ParentRequestTypeId { get; set; }

* + - * + RequestTypeFindModel

public int? Id { get; set; }

public int? DepartmentId { get; set; }

public string Name { get; set; }

public int? ExecutionDay { get; set; }

public string Description { get; set; }

public DateTime? CreatedDate { get; set; }

public int? ParentRequestTypeId { get; set; }

public int? PageSize { get; set; }

public int? PageNumber { get; set; }

public int? SourceTypeId { get; set; }

* + - * + RequestTypeDeleteModel

public int Id { get; set; }

* + - * + RequestTypeCreateModel

public int? DepartmentId { get; set; }

public string Name { get; set; }

public int ExecutionDay { get; set; }

public int SourceTypeId { get; set; }

public string Description { get; set; }

public int? ParentRequestTypeId { get; set; }

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable< RequestTypeViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable< RequestTypeViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

RequestTypeViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task< RequestTypeViewModel>

* + - * + **Find**

**Request model**

RequestTypeFindModel parameter

**Response model**

RequestTypeViewModel

* + - * + **FindAsync**

**Request model**

RequestTypeFindModel parameter

**Response model**

Task< RequestTypeViewModel>

* + - * + **FindAll**

**Request model**

RequestTypeFindModel parameter

**Response model**

Icollection< RequestTypeViewModel>

* + - * + **FindAllAsync**

**Request model**

RequestTypeFindModel parameter

**Response model**

Task<Icollection< RequestTypeViewModel>>

* + - * + **Add**

**Request model**

RequestTypeCreateModel parameter

**Response model**

RequestTypeViewModel

* + - * + **AddAsync**

**Request model**

RequestTypeCreateModelparameter

**Response model**

Task< RequestTypeViewModel>

* + - * + **AddRangeAsync**

**Request model**

List< RequestTypeCreateModel> parameter

**Response model**

Task<List< RequestTypeViewModel>>

* + - * + **Update**

**Request model**

RequestTypeUpdateModel parameter

**Response model**

RequestTypeViewModel

* + - * + **UpdateAsync**

**Request model**

RequestTypeUpdateModel parameter

**Response model**

Task<RequestTypeViewModel>

* + - **RequestStatus**
      * **Models**
        + RequestStatusViewModel

public int Id { get; set; }

public string Name { get; set; }

public DateTime CreatedDate { get; set; }

public string Description { get; set; }

public int? TotalCount { get; set; }

* + - * + RequestStatusUpdateModel

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

* + - * + RequestStatusFindModel

public int? Id { get; set; }

public string Name { get; set; }

public DateTime? CreatedDate { get; set; }

public string Description { get; set; }

public int? PageSize { get; set; }

public int? PageNumber { get; set; }

* + - * + RequestStatusDeleteModel

public int Id { get; set; }

* + - * + RequestStatusCreateModel

public string Name { get; set; }

public string Description { get; set; }

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable< RequestTypeViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable< RequestStatusViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

RequestStatusViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task< RequestStatusViewModel>

* + - * + **Find**

**Request model**

RequestStatusFindModel parameter

**Response model**

RequestStatusViewModel

* + - * + **FindAsync**

**Request model**

RequestStatusFindModel parameter

**Response model**

Task< RequestStatusViewModel>

* + - * + **FindAll**

**Request model**

RequestStatusFindModel parameter

**Response model**

Icollection< RequestStatusViewModel>

* + - * + **FindAllAsync**

**Request model**

RequestStatusFindModel parameter

**Response model**

Task<Icollection< RequestStatusViewModel>>

* + - * + **Add**

**Request model**

RequestStatusCreateModel parameter

**Response model**

RequestStatusViewModel

* + - * + **AddAsync**

**Request model**

RequestStatusCreateModelparameter

**Response model**

Task< RequestStatusViewModel>

* + - * + **AddRangeAsync**

**Request model**

List< RequestStatusCreateModel> parameter

**Response model**

Task<List< RequestStatusViewModel>>

* + - * + **Update**

**Request model**

RequestStatusUpdateModel parameter

**Response model**

RequestStatusViewModel

* + - * + **UpdateAsync**

**Request model**

RequestStatusUpdateModel parameter

**Response model**

Task<RequestStatusViewModel>

* + - **User**
      * **Models**
        + UserViewModel

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string UserName { get; set; }

public string Email { get; set; }

public string MobileNumber { get; set; }

public string PhoneNumber { get; set; }

public int? AccessFailedCount { get; set; }

public DateTime LastLoginDate { get; set; }

public string ConnectedId { get; set; }

public virtual UserDto ParentUser { get; set; }

public virtual DepartmentDto Department { get; set; }

public virtual ICollection<UserClaimDto> UserClaim { get; set; }

public virtual ICollection<RoleDto> Role { get; set; }

public virtual ICollection<UserDto> ChildUsers { get; set; }

* + - * + UserUpdateModel

public int Id { get; set; }

public string Email { get; set; }

public string UserName { get; set; }

public string Password { get; set; }

public string Salt { get; set; }

public string PhoneNumber { get; set; }

public string MobileNumber { get; set; }

public int? AccessFailedCount { get; set; }

* + - * + UserFindModel

public int? Id { get; set; }

public DateTime? LastLoginDate { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; set; }

public string UserName { get; set; }

public string Password { get; set; }

public string Salt { get; set; }

public string PhoneNumber { get; set; }

public string MobileNumber { get; set; }

public int? AccessFailedCount { get; set; }

public string ConnectedId { get; set; }

public int? PageSize { get; set; }

public int? PageNumber { get; set; }

* + - * + UserDeleteModel

public int Id { get; set; }

* + - * + UserCreateModel

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; set; }

public string UserName { get; set; }

public string Password { get; set; }

public string Salt { get; set; }

public string PhoneNumber { get; set; }

public string MobileNumber { get; set; }

public int? AccessFailedCount { get; set; }

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable< UserViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable< UserViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

UserViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task< UserViewModel>

* + - * + **Find**

**Request model**

UserFindModel parameter

**Response model**

UserViewModel

* + - * + **FindAsync**

**Request model**

UserFindModel parameter

**Response model**

Task< UserViewModel>

* + - * + **FindAll**

**Request model**

UserFindModel parameter

**Response model**

Icollection< UserViewModel>

* + - * + **FindAllAsync**

**Request model**

UserFindModel parameter

**Response model**

Task<Icollection< UserViewModel>>

* + - * + **Add**

**Request model**

UserCreateModel parameter

**Response model**

UserViewModel

* + - * + **AddAsync**

**Request model**

UserCreateModelparameter

**Response model**

Task< UserViewModel>

* + - * + **AddRangeAsync**

**Request model**

List< UserCreateModel> parameter

**Response model**

Task<List< UserViewModel>>

* + - * + **Update**

**Request model**

UserUpdateModel parameter

**Response model**

UserViewModel

* + - * + **UpdateAsync**

**Request model**

UserUpdateModel parameter

**Response model**

Task<UserViewModel>

* + - **Project**
      * **Models**
        + ProjectViewModel

public int Id { get; set; }

public int DepartmentId { get; set; }

public string Name { get; set; }

public DateTime CreatedDate { get; set; }

public string Description { get; set; }

public DepartmentDto Department { get; set; }

* + - * + ProjectUpdateModel

public int Id { get; set; }

public int DepartmentId { get; set; }

public string Name { get; set; }

public string Description { get; set; }

* + - * + ProjectFindModel

public int? Id { get; set; }

public int? DepartmentId { get; set; }

public string Name { get; set; }

public DateTime? CreatedDate { get; set; }

public string Description { get; set; }

public int? PageSize { get; set; }

public int? PageNumber { get; set; }

* + - * + ProjectDeleteModel

public int Id { get; set; }

* + - * + ProjectCreateModel

public int DepartmentId { get; set; }

public string Name { get; set; }

public string Description { get; set; }

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable< ProjectViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable< ProjectViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

ProjectViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task< ProjectViewModel>

* + - * + **Find**

**Request model**

ProjectFindModel parameter

**Response model**

ProjectViewModel

* + - * + **FindAsync**

**Request model**

ProjectFindModel parameter

**Response model**

Task< ProjectViewModel>

* + - * + **FindAll**

**Request model**

ProjectFindModel parameter

**Response model**

Icollection< ProjectViewModel>

* + - * + **FindAllAsync**

**Request model**

ProjectFindModel parameter

**Response model**

Task<Icollection< ProjectViewModel>>

* + - * + **Add**

**Request model**

ProjectCreateModel parameter

**Response model**

ProjectViewModel

* + - * + **AddAsync**

**Request model**

ProjectCreateModelparameter

**Response model**

Task< ProjectViewModel>

* + - * + **AddRangeAsync**

**Request model**

List< ProjectCreateModel> parameter

**Response model**

Task<List< ProjectViewModel>>

* + - * + **Update**

**Request model**

ProjectUpdateModel parameter

**Response model**

ProjectViewModel

* + - * + **UpdateAsync**

**Request model**

ProjectUpdateModel parameter

**Response model**

Task<ProjectViewModel>

* + - **Comment**
      * **Models**
        + CommentCreateModel

public int MainTaskId { get; set; }

public int UserId { get; set; }

public string Content { get; set; }

public List<AttachmentCreateModel> Attachment { get; set; }

* + - * + CommentDeleteModel

public int Id { get; set; }

* + - * + CommentViewModel

public int Id { get; set; }

public int MainTaskId { get; set; }

public int? UserId { get; set; }

public DateTime? CreatedDate { get; set; }

public string Content { get; set; }

public virtual TaskDto MainTask { get; set; }

public virtual UserDto User { get; set; }

public List<AttachmentDto> Attachment { get; set; }

* + - * + CommentFindModel

public int? Id { get; set; }

public int? MainTaskId { get; set; }

public int? UserId { get; set; }

public DateTime? CreatedDate { get; set; }

public string Content { get; set; }

public int? PageSize { get; set; }

public int? PageNumber { get; set; }

* + - * + CommentUpdateModel

public int Id { get; set; }

public int MainTaskId { get; set; }

public string UserId { get; set; }

public string Content { get; set; }

* + - * **Actions**
        + **GetAll**

**Request model**

Filter filter

**Response model**

IEnumerable< CommentViewModel>

* + - * + **GetAllAsync**

**Request model**

Filter filter

**Response model**

Task<IEnumerable< CommentViewModel>>

* + - * + **GetById**

**Request model**

int id

**Response model**

CommentViewModel

* + - * + **GetByIdAsync**

**Request model**

int id

**Response model**

Task< CommentViewModel>

* + - * + **Find**

**Request model**

CommentFindModel parameter

**Response model**

CommentViewModel

* + - * + **FindAsync**

**Request model**

CommentFindModel parameter

**Response model**

Task< CommentViewModel>

* + - * + **FindAll**

**Request model**

CommentFindModel parameter

**Response model**

Icollection< CommentViewModel>

* + - * + **FindAllAsync**

**Request model**

CommentFindModel parameter

**Response model**

Task<Icollection< CommentViewModel>>

* + - * + **Add**

**Request model**

CommentCreateModel parameter

**Response model**

CommentViewModel

* + - * + **AddAsync**

**Request model**

CommentCreateModelparameter

**Response model**

Task< CommentViewModel>

* + - * + **AddRangeAsync**

**Request model**

List< CommentCreateModel> parameter

**Response model**

Task<List< CommentViewModel>>

* + - * + **Update**

**Request model**

CommentUpdateModel parameter

**Response model**

CommentViewModel

* + - * + **UpdateAsync**

**Request model**

CommentUpdateModel parameter

**Response model**

Task<CommentViewModel>

* + - **Attachment**
      * **Models**
        + AttachmentCreateModel

public string FilePath { get; set; }

public int? CustomerRequestId { get; set; }

public int? MainTaskId { get; set; }

public int? CommentId { get; set; }

public int? FileType { get; set; }

public string Description { get; set; }

public string MediaType { get; set; }

public string Container { get; set; }

public string Extension { get; set; }

* + - * + AttachmentViewModel

public int Id { get; set; }

public string FilePath { get; set; }

public int? CustomerRequestId { get; set; }

public int? MainTaskId { get; set; }

public int? CommentId { get; set; }

public int? FileType { get; set; }

public string Description { get; set; }

public DateTime? CreatedDate { get; set; }

* + - * **Actions**
        + **Add**

**Request model**

AttachmentCreateModel parameter

**Response model**

AttachmentViewModel

* + - * + **AddAsync**

**Request model**

AttachmentCreateModelparameter

**Response model**

Task< AttachmentViewModel>

* **NIS.BL** **(Business Logic).** Bu proyekt NIS.Service-dən daxil olan data üzərində əməliyyatlar( məs: gələn datanın kim tərəfindən əlavə edildiyi məlumatının dataya əlavə olunması və s.) həyata keçirib Bazaya yazılacaq formaya uyğun hazır dataya çevirir və bu datanı NIS.DAL proyektinə ötürür və qayidan dataya görə NIS.Service proyektinə cavab qaytarır. Əməliyatlar Logic-lər tərəfindən həyata keçirilir.
  + **Logiclər**
    - Region
    - Area
    - Node
    - Department
    - MainTask
    - TaskStatus
    - Report
    - CustomerRequest
    - CustomerComment
    - RequestStatus
    - Comment
    - Comment
    - Comment
    - Attachment
  + **Methodlar**
    - GetAll
    - GetAllAsync
    - GetById
    - GetByIdAsync
    - Find
    - FindAsync
    - FindAll
    - FindAllAsync
    - Add
    - AddAsync
    - AddRangeAsync
    - Update
    - UpdateAsync
* **NIS.DAL (Data Access Layer)**. Gələn məlumatın bazaya əlave edilməsi, silinməsi, dəyişdirilməsi və oxunması əməliyatlarını həyata keçirir və NIS.BL proyektinə cavab qaytarır.
* **NIS.Security**. Authentication və authorization proseslərini həyata keçirir.
* **NIS.Utils** . əlavə yardımçı siniflərin (enum, automapper, Chashing və s.) saxlanılması üçün istifadə edilir.

### Token

{bu metod serverda yerləşir və authorization üçün istifadə olunur. Bütün sorğular token-lə göndərilməlidir. Yəni ilk olarak token controlleri çağrılır və müvafiq token alındıqdan sonra digər sorğuların header-ində bu tokeni əlavə etməklə sorğular göndərilir }

**URL :**  /token

**Type:**  POST

**Description:**  Sistemə giriş

**Request header :**  null

**Request body:**  grant\_type:password

Commentname:*Commentname*

password:*password*

**Response:**  {

"access\_token":"kGkUerPZduRCtIDghE82uRcg3ImMpfmpaTT…..",

"token\_type": "bearer",

"expires\_in": 86399,

"Comment\_name": "*Comment*",

"departmentCode": "*code*",

".issued": "Sun, 19 Jun 2016 07:58:10 GMT",

".expires": "Mon, 20 Jun 2016 07:58:10 GMT"

}

Növbəti sorğuların hər birində request body ilə yanaşı header-də bu formada sorğu göndərmək mütləqdir:

Content-Type: **application/json**

Accept: **application/json**

Authorization: **Bearer** *fyEUSBoc…..HGhRQ3ioM7IJ0lc\_a\_pqsXZfjYJPprXuO….*

{ Metodlar məsələm Department

### Department

|  |  |  |  |
| --- | --- | --- | --- |
| API type/url | Description | Request body sample | Response data sample |
| POST api/Department | Yeni müəssisə əlavə etmək | {  "departmentCode": 2,  "departmentName": "sample string 3"  } | Blank,  Code: 200,  Status: OK |
| GET  api/Department | Bütün müəssisələrin siyahısını qaytarır |  | [  {  "id": 6,  "departmentCode": 12,  "departmentName": "Abşeron"  }  ] |
| PUT api/Department/{id} | Müəssisəni dəyişdirir | {  "departmentCode": 2,  "departmentName": "sample string 4"  } | Blank,  Code: 200,  Status: OK |
| DELETE api/Department/{id} | Müəssisəni silir |  | Blank,  Code: 200,  Status: OK |
| GET api/Department/{id} | Seçilmiş müəssisənin məlmuatlarını qaytarır |  | {  "id": 6,  "departmentCode": 12,  "departmentName": "Abşeron"  } |

## Client

Proqramın client tərəfi yalnız HTML5, CSS3 və Javascript istifadə edilərək hazırlanmışdır. Proqram təminatında həmçinin Angular.Js { frameworklar adları} istifadə edilmişdir. Client side aşağıdakı qovluqları özündə birləşdirir:

Programm modullat sadalanır və hər modul hansı funksiyaları yerinə yetirir və harda yerləşir

### İstifadəçi interfeysi

Hansı modular klient interfesi yaradır və necə.

## Etibarlılıq və təhlükəsizlik

Qeyd edildiyi kimi proqram təminatı Oauth2 texnologiyası OWIN körpüsü vasitəsilə access token üzərindən çalışır. İsitfadəçinin sistemə daxil olmasından sonra token istifadəçi hüquqlarını tənzimləməyə kömək edir. Göstərilmiş controllerlərin hər biri token yoxlanılmasından sonra əlçatan olur,əks halda istifadəçiyə və ya thir party application-lara 404 not autherization mesajı göndərilir.

Proqram təminatının verilənlər bazası {Data Base adı} üzərində qurulmuşdur. Məlumat bazasının əlyetənliyi yalnız server side-da mümkündür. Bu səbəbdən client side-dan məlumat bazasına heç bir vəcdlə müdaxilə edilməsi mümkün deyildir. Digər tərəfdən server side-da məlumat bazasının zəruri cədvəllərinə lazımi manipulyasiyaları etmək üçün ayrılmış məlumat baza hesabı üzərindən bazanın digər elementləri üzərində heç bir əməliyyata icazə verilmir. Məlumat bazasında 2 (iki) aylıq transaction loglar saxlanılır və həftəlik backuplar çıxarılır.

# **Proqramın buraxılma prosesi**

Proqramın işə salınması üçün aşağıdakılar tələb olunur:

## Environment

{Hansı webserver

Hansı DataBase}

{Cədvəllər sadalanır və hər cədvəl üçün əlavə məlumat verilməlidir

Hansı məqsədlə həmən cədvəl yaranıb. Hansı hədəflər qoyulmuşdur

Kim tərəfindən orada məlumat daxil və deyişmək olur

Hər sütünün adı tipi və izahatı necə cədvəl digər cədvəllərlə alagələnir

}

# **Proqram mesajları**

Proqram aşağıdakı mesajları qaytarır.

|  |  |  |  |
| --- | --- | --- | --- |
| S.S | Code | Status | Description |
|  | 200 | OK | Sorğu düzgündür və əlavə sorğu tələb etmir |
|  | 000 | OK | Proses uğurla tamamlandı |
|  | 404 | Bad | Sizin bu sorğuya icazəniz yoxdur |
|  | 101 | Bad | Login və ya şifrə yalnışdır |
|  | 102 | Bad | Rəqəm tələn olunan sahədə mətn yazırsınız |
|  | 103 | Bad | Mətn tələb olunan sahədə rəqəm yazırsınız |
|  | 104 | Bad | Tarix formatı düzgün deyil (dd.mm.YYYY) |
|  | 105 | OK | Sorğu uğurla tamamlandı, lakin proses hələ bitməyib |