

# Namespace Backend\_Reservation\_Service. Controllers

## Classes

### [AccountController](#)

Controller for managing user accounts.

### [NewsController](#)

a Controller for managing news

### [ReservationsController](#)

Controller for handling reservation-related actions.

### [UploadFilesController](#)

Controller for uploading files

### [VehiclesController](#)

Controller for managing vehicles

# Class AccountController

Namespace: [Backend\\_Reservation\\_Service.Controllers](#)

Assembly: Backend\_Reservation\_Service.dll

Controller for managing user accounts.

```
[Route("api/[controller]")]
[ApiController]
[Authorize]
public class AccountController : ControllerBase
```

## Inheritance

[object](#) ← [ControllerBase](#) ← AccountController

## Inherited Members

[ControllerBase.StatusCode\(int\)](#) , [ControllerBase.StatusCode\(int, object\)](#) ,  
[ControllerBase.Content\(string\)](#) , [ControllerBase.Content\(string, string\)](#) ,  
[ControllerBase.Content\(string, string, Encoding\)](#) ,  
[ControllerBase.Content\(string, MediaTypeHeaderValue\)](#) , [ControllerBase.NoContent\(\)](#) ,  
[ControllerBase.Ok\(\)](#) , [ControllerBase.Ok\(object\)](#) , [ControllerBase.Redirect\(string\)](#) ,  
[ControllerBase.RedirectPermanent\(string\)](#) ,  
[ControllerBase.RedirectPreserveMethod\(string\)](#) ,  
[ControllerBase.RedirectPermanentPreserveMethod\(string\)](#) ,  
[ControllerBase.LocalRedirect\(string\)](#) , [ControllerBase.LocalRedirectPermanent\(string\)](#) ,  
[ControllerBase.LocalRedirectPreserveMethod\(string\)](#) ,  
[ControllerBase.LocalRedirectPermanentPreserveMethod\(string\)](#) ,  
[ControllerBase.RedirectToAction\(\)](#) , [ControllerBase.RedirectToAction\(string\)](#) ,  
[ControllerBase.RedirectToAction\(string, object\)](#) ,  
[ControllerBase.RedirectToAction\(string, string\)](#) ,  
[ControllerBase.RedirectToAction\(string, string, object\)](#) ,  
[ControllerBase.RedirectToAction\(string, string, string\)](#) ,  
[ControllerBase.RedirectToAction\(string, string, object, string\)](#) ,  
[ControllerBase.RedirectToActionPreserveMethod\(string, string, object, string\)](#) ,  
[ControllerBase.RedirectToActionPermanent\(string\)](#) ,  
[ControllerBase.RedirectToActionPermanent\(string, object\)](#) ,  
[ControllerBase.RedirectToActionPermanent\(string, string\)](#) ,  
[ControllerBase.RedirectToActionPermanent\(string, string, string\)](#) ,  
[ControllerBase.RedirectToActionPermanent\(string, string, object\)](#) ,

[ControllerBase.RedirectToActionPermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string\)](#)  ,  [ControllerBase.RedirectToRoute\(object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanentPreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string\)](#)  ,  [ControllerBase.RedirectToPage\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.File\(byte\[\], string\)](#)  ,  [ControllerBase.File\(byte\[\], string, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string\)](#)  ,  [ControllerBase.File\(Stream, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(string, string\)](#)  ,  [ControllerBase.File\(string, string, bool\)](#)  ,

[ControllerBase.File\(string, string, string\)](#)  ,  [ControllerBase.File\(string, string, string, bool\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.Unauthorized\(\)](#)  ,  [ControllerBase.Unauthorized\(object\)](#)  ,  
 [ControllerBase.NotFound\(\)](#)  ,  [ControllerBase.NotFound\(object\)](#)  ,  
 [ControllerBase.BadRequest\(\)](#)  ,  [ControllerBase.BadRequest\(object\)](#)  ,  
 [ControllerBase.BadRequest\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.UnprocessableEntity\(\)](#)  ,  [ControllerBase.UnprocessableEntity\(object\)](#)  ,  
 [ControllerBase.UnprocessableEntity\(ModelStateDictionary\)](#)  ,  [ControllerBase.Conflict\(\)](#)  ,  
 [ControllerBase.Conflict\(object\)](#)  ,  [ControllerBase.Conflict\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.Problem\(string, string, int?, string, string\)](#)  ,  
 [ControllerBase.Problem\(string, string, int?, string, string, IDictionary<string, object>\)](#)  ,  
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#)  ,  
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.ValidationProblem\(\)](#)  ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#)  ,  
  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary, IDictionary<string, object>\)](#)  ,  
 [ControllerBase.Created\(\)](#)  ,  [ControllerBase.Created\(string, object\)](#)  ,  
 [ControllerBase.Created\(Uri, object\)](#)  ,  [ControllerBase.CreatedAtAction\(string, object\)](#)  ,  
 [ControllerBase.CreatedAtAction\(string, object, object\)](#)  ,  
 [ControllerBase.CreatedAtAction\(string, string, object, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(string, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(object, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(string, object, object\)](#)  ,  [ControllerBase.Accepted\(\)](#)  ,  
 [ControllerBase.Accepted\(object\)](#)  ,  [ControllerBase.Accepted\(Uri\)](#)  ,  
 [ControllerBase.Accepted\(string\)](#)  ,  [ControllerBase.Accepted\(string, object\)](#)  ,  
 [ControllerBase.Accepted\(Uri, object\)](#)  ,  [ControllerBase.AcceptedAtAction\(string\)](#)  ,

[ControllerBase.AcceptedAtAction\(string, string\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, string, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, object, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, string, object, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(object\)](#)  ,  [ControllerBase.AcceptedAtRoute\(string\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(string, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(object, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(string, object, object\)](#)  ,  [ControllerBase.Challenge\(\)](#)  ,  
 [ControllerBase.Challenge\(params string\[\]\)](#)  ,  
 [ControllerBase.Challenge\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.Forbid\(\)](#)  ,  [ControllerBase.Forbid\(params string\[\]\)](#)  ,  
 [ControllerBase.Forbid\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal\)](#)  ,  [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#)  ,  
 [ControllerBase.SignOut\(\)](#)  ,  [ControllerBase.SignOut\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.SignOut\(params string\[\]\)](#)  ,  
 [ControllerBase.SignOut\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\] \)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\] \)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#)  ,  
 [ControllerBase.TryValidateModel\(object\)](#)  ,  
 [ControllerBase.TryValidateModel\(object, string\)](#)  ,  [ControllerBase.HttpContext](#)  ,  
 [ControllerBase.Request](#)  ,  [ControllerBase.Response](#)  ,  [ControllerBase.RouteData](#)  ,  
 [ControllerBase.ModelState](#)  ,  [ControllerBase.ControllerContext](#)  ,  
 [ControllerBase.MetadataProvider](#)  ,  [ControllerBase.ModelBinderFactory](#)  ,

[ControllerBase.Url](#) ,  [ControllerBase.ObjectValidator](#) ,  
 [ControllerBase.ProblemDetailsFactory](#) ,  [ControllerBase.User](#) ,  [ControllerBase.Empty](#) ,  
 [object.Equals\(object\)](#) ,  [object.Equals\(object, object\)](#) ,  [object.GetHashCode\(\)](#) ,  
 [object.GetType\(\)](#) ,  [object.MemberwiseClone\(\)](#) ,  [object.ReferenceEquals\(object, object\)](#) ,  
 [object.ToString\(\)](#)

## Constructors

**AccountController(UserManager<User>, IAccountService, AuthenticationService)**

Initializes a new instance of the  [AccountController](#) class.

```
public AccountController(UserManager<User> userManager, IAccountService  
accountService, AuthenticationService authenticationService)
```

## Parameters

**userManager**  [UserManager](#)<User>

The user manager.

**accountService**  [IAccountService](#)

The account service.

**authenticationService**  [AuthenticationService](#)

The authentication service.

## Methods

**AdminDeleteUser(string?)**

Deletes (or marks as deleted) a user by a user with admin privliges.

```
[HttpDelete("admin-delete/{username}")]  
[Authorize(Roles = "Admin")]  
public Task<IActionResult> AdminDeleteUser(string? username)
```

## Parameters

`username` [string](#)

The username of the user to delete.

## Returns

[Task](#) <[IActionResult](#)>

An IActionResult indicating the result of the delete operation.

## Remarks

Example request:

```
DELETE /api/account/admin-delete/john_doe
```

## AssignRole(UserRole)

Assigns a role to a user.

```
[HttpPost("assign-role")]
[Authorize(Roles = "Admin")]
public Task<IActionResult> AssignRole(UserRole model)
```

## Parameters

`model` [UserRole](#)

The user role details.

## Returns

[Task](#) <[IActionResult](#)>

An IActionResult indicating the result of the role assignment.

## Remarks

Example request:

```
POST /api/account/assign-role
{
    "Username": "john_doe",
    "Role": "Admin"
}
```

## ChangePwd(UpdateUserPwdDTO)

Changes the password of the authorized user.

```
[HttpPut("user/change-pwd")]
public Task<IActionResult> ChangePwd(UpdateUserPwdDTO model)
```

### Parameters

**model** [UpdateUserPwdDTO](#)

The password change details.

### Returns

[Task](#)<[IActionResult](#)>

An IActionResult indicating the result of the password change operation.

### Remarks

Example request:

```
PUT /api/account/user/change-pwd
{
    "OldPwd": "OldPassword123!",
    "NewPwd": "NewPassword123!",
    "NewPwd2": "NewPassword123!"
}
```

## CreateRole(string)

Creates a new role.

```
[HttpPost("create-role")]
[Authorize(Roles = "Admin")]
public Task<IActionResult> CreateRole(string role)
```

## Parameters

**role** [string](#)

The role name.

## Returns

[Task](#)<[IActionResult](#)>

An IActionResult indicating the result of the role creation.

## Remarks

Example request:

```
POST /api/account/create-role
{
    "role": "Admin"
}
```

## DeleteUser()

Deletes (or marks as deleted) the authorized user.

```
[HttpDelete("user")]
public Task<IActionResult> DeleteUser()
```

## Returns

[Task](#)<[IActionResult](#)>

An IActionResult indicating the result of the delete operation.

## Remarks

Example request:

```
DELETE /api/account/user
```

## GetAllUsersAsync(int, int, bool, bool)

Gets a list of all users, including ones marked as deleted (IsDeleted = true) with pagination.  
Admin privilege required.

```
[HttpGet("users")]
[Authorize(Roles = "Admin")]
public Task<ActionResult> GetAllUsersAsync(int itemsPerPage, int pageNumber, bool
orderDesc, bool onlyDeleted)
```

### Parameters

`itemsPerPage` [int](#)

The number of items per page.

`pageNumber` [int](#)

The page number to retrieve.

`orderDesc` [bool](#)

Whether to order the results in descending order.

`onlyDeleted` [bool](#)

Whether to include only deleted users.

### Returns

[Task](#)<[ActionResult](#)>

An ActionResult containing a list of all users and pagination details.

### Remarks

Example request:

```
GET /api/account/users?itemsPerPage=10&pageNumber=1&orderDesc=true&onlyDeleted=false
```

## GetUser(string)

Gets a user by username.

```
[HttpGet("user/{username}")]
public Task<IActionResult> GetUser(string username)
```

### Parameters

username [string](#)

The username.

### Returns

[Task](#)<[IActionResult](#)>

An IActionResult containing the user details.

### Remarks

Example request:

```
GET /api/account/user/john_doe
```

## GetUserAuthorized()

Gets the authorized user's details.

```
[HttpGet("user/authorized")]
public Task<IActionResult> GetUserAuthorized()
```

### Returns

[Task](#)<[IActionResult](#)>

An IActionResult containing the authorized user's details.

## Remarks

Example request:

```
GET /api/account/user/authorized
```

## LogOut(LogOut)

Logs out the current user.

```
[HttpPost("logout")]
public Task<IActionResult> LogOut(LogOut model)
```

## Parameters

**model** [LogOut](#)

The logout details.

## Returns

[Task](#)<[IActionResult](#)>

An IActionResult indicating the result of the logout.

## Remarks

Example request:

```
POST /api/account/logout
{
    "Username": "john_doe"
}
```

## Login(Login)

Authenticates a user and generates a JWT token.

```
[HttpPost("login")]
[AllowAnonymous]
public Task<IActionResult> Login(Login model)
```

## Parameters

model [Login](#)

The login details.

## Returns

[Task](#)<[IActionResult](#)>

An IActionResult containing the JWT token if authentication is successful.

## Remarks

Example request:

```
POST /api/account/login
{
    "Username": "john_doe",
    "Password": "Password123!"
}
```

## Register(Register)

Registers a new user.

```
[HttpPost("register")]
[AllowAnonymous]
public Task<IActionResult> Register(Register model)
```

## Parameters

model [Register](#)

The registration details.

Returns

[Task](#) <[IActionResult](#)>

An IActionResult indicating the result of the registration.

Remarks

Example request:

```
POST /api/account/register
{
    "UserName": "john_doe",
    "Email": "john.doe@example.com",
    "Password": "Password123!"
}
```

## RestoreSoftDeletedUser(string?)

Restore softdeleted account If both username and id are given, will prioritize username

```
[HttpPut("restore")]
[Authorize(Roles = "Admin")]
public Task<ActionResult<FullUserDTO>> RestoreSoftDeletedUser(string? username)
```

Parameters

username [string](#)

Returns

[Task](#) <[ActionResult](#)> <[FullUserDTO](#)>>

## UpdateUser(UpdateUserDTO?)

Updates the details of the authorized user.

```
[HttpPut("user/update")]
public Task<IActionResult> UpdateUser(UpdateUserDTO? model)
```

## Parameters

model [UpdateUserDTO](#)

The user details to update.

## Returns

[Task](#)<[IActionResult](#)>

An IActionResult indicating the result of the update operation.

## Remarks

Example request:

```
PUT /api/account/user/update
{
    "FirstName": "John",
    "LastName": "Doe",
    "Email": "john.doe@example.com"
}
```

# Class NewsController

Namespace: [Backend\\_Reservation\\_Service.Controllers](#)

Assembly: Backend\_Reservation\_Service.dll

a Controller for managing news

```
[Route("api/[controller]")]
[ApiController]
public class NewsController : Controller, IActionFilter, IAsyncActionFilter,
IFilterMetadata, IDisposable
```

## Inheritance

[object](#) ← [ControllerBase](#) ← [Controller](#) ← NewsController

## Implements

[IActionFilter](#), [IAsyncActionFilter](#), [IFilterMetadata](#), [IDisposable](#)

## Inherited Members

[Controller.View\(\)](#), [Controller.View\(string\)](#), [Controller.View\(object\)](#),  
[Controller.View\(string, object\)](#), [Controller.PartialView\(\)](#), [Controller.PartialView\(string\)](#),  
[Controller.PartialView\(object\)](#), [Controller.PartialView\(string, object\)](#),  
[Controller.ViewComponent\(string\)](#), [Controller.ViewComponent\(Type\)](#),  
[Controller.ViewComponent\(string, object\)](#), [Controller.ViewComponent\(Type, object\)](#),  
[Controller.Json\(object\)](#), [Controller.Json\(object, object\)](#),  
[Controller.OnActionExecuting\(ActionExecutingContext\)](#),  
[Controller.OnActionExecuted\(ActionExecutedContext\)](#),  
[Controller.OnActionExecutionAsync\(ActionExecutingContext, ActionExecutionDelegate\)](#),  
[Controller.Dispose\(\)](#), [Controller.Dispose\(bool\)](#), [Controller.ViewData](#),  
[Controller.TempData](#), [Controller.ViewBag](#), [ControllerBase.StatusCode\(int\)](#),  
[ControllerBase.StatusCode\(int, object\)](#), [ControllerBase.Content\(string\)](#),  
[ControllerBase.Content\(string, string\)](#), [ControllerBase.Content\(string, string, Encoding\)](#),  
[ControllerBase.Content\(string, MediaTypeHeaderValue\)](#), [ControllerBase.NoContent\(\)](#),  
[ControllerBase.Ok\(\)](#), [ControllerBase.Ok\(object\)](#), [ControllerBase.Redirect\(string\)](#),  
[ControllerBase.RedirectPermanent\(string\)](#),  
[ControllerBase.RedirectPreserveMethod\(string\)](#),  
[ControllerBase.RedirectPermanentPreserveMethod\(string\)](#),  
[ControllerBase.LocalRedirect\(string\)](#), [ControllerBase.LocalRedirectPermanent\(string\)](#),  
[ControllerBase.LocalRedirectPreserveMethod\(string\)](#),  
[ControllerBase.LocalRedirectPermanentPreserveMethod\(string\)](#),

[ControllerBase.RedirectToAction\(\)](#)  ,  [ControllerBase.RedirectToAction\(string\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, object\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string\)](#)  ,  [ControllerBase.RedirectToRoute\(object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanentPreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string\)](#)  ,  [ControllerBase.RedirectToPage\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.File\(byte\[\], string\)](#)  ,  [ControllerBase.File\(byte\[\], string, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, bool\)](#)  ,

[ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string\)](#)  ,  [ControllerBase.File\(Stream, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(string, string\)](#)  ,  [ControllerBase.File\(string, string, bool\)](#)  ,  
 [ControllerBase.File\(string, string, string\)](#)  ,  [ControllerBase.File\(string, string, string, bool\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.Unauthorized\(\)](#)  ,  [ControllerBase.Unauthorized\(object\)](#)  ,  
 [ControllerBase.NotFound\(\)](#)  ,  [ControllerBase.NotFound\(object\)](#)  ,  
 [ControllerBase.BadRequest\(\)](#)  ,  [ControllerBase.BadRequest\(object\)](#)  ,  
 [ControllerBase.BadRequest\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.UnprocessableEntity\(\)](#)  ,  [ControllerBase.UnprocessableEntity\(object\)](#)  ,  
 [ControllerBase.UnprocessableEntity\(ModelStateDictionary\)](#)  ,  [ControllerBase.Conflict\(\)](#)  ,  
 [ControllerBase.Conflict\(object\)](#)  ,  [ControllerBase.Conflict\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.Problem\(string, string, int?, string, string\)](#)  ,  
 [ControllerBase.Problem\(string, string, int?, string, string, IDictionary<string, object>\)](#)  ,  
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#)  ,  
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.ValidationProblem\(\)](#)  ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#)  ,

[ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary, IDictionary<string, object>\)](#)  ,  
 [ControllerBase.Created\(\)](#)  ,  [ControllerBase.Created\(string, object\)](#)  ,  
 [ControllerBase.Created\(Uri, object\)](#)  ,  [ControllerBase.CreatedAtAction\(string, object\)](#)  ,  
 [ControllerBase.CreatedAtAction\(string, object, object\)](#)  ,  
 [ControllerBase.CreatedAtAction\(string, string, object, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(string, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(object, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(string, object, object\)](#)  ,  [ControllerBase.Accepted\(\)](#)  ,  
 [ControllerBase.Accepted\(object\)](#)  ,  [ControllerBase.Accepted\(Uri\)](#)  ,  
 [ControllerBase.Accepted\(string\)](#)  ,  [ControllerBase.Accepted\(string, object\)](#)  ,  
 [ControllerBase.Accepted\(Uri, object\)](#)  ,  [ControllerBase.AcceptedAtAction\(string\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, string\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, string, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, object, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, string, object, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(object\)](#)  ,  [ControllerBase.AcceptedAtRoute\(string\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(string, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(object, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(string, object, object\)](#)  ,  [ControllerBase.Challenge\(\)](#)  ,  
 [ControllerBase.Challenge\(params string\[\]\)](#)  ,  
 [ControllerBase.Challenge\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.Forbid\(\)](#)  ,  [ControllerBase.Forbid\(params string\[\]\)](#)  ,  
 [ControllerBase.Forbid\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal\)](#)  ,  [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#)  ,  
 [ControllerBase.SignOut\(\)](#)  ,  [ControllerBase.SignOut\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.SignOut\(params string\[\]\)](#)  ,  
 [ControllerBase.SignOut\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\] \)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#)  ,

[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string\)](#) ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,  
 [ControllerBase.TryValidateModel\(object\)](#) ,  
 [ControllerBase.TryValidateModel\(object, string\)](#) , ControllerBase.HttpContext ,  
 ControllerBase.Request , ControllerBase.Response , ControllerBase.RouteData ,  
 ControllerBase.ModelState , ControllerBase.ControllerContext ,  
 ControllerBase.MetadataProvider , ControllerBase.ModelBinderFactory ,  
 ControllerBase.Url , ControllerBase.ObjectValidator ,  
 ControllerBase.ProblemDetailsFactory , ControllerBase.User , ControllerBase.Empty ,  
 object.Equals(object) , object.Equals(object, object) , object.GetHashCode() ,  
 object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) ,  
 object.ToString()

## Constructors

### NewsController(INewsService)

```
public NewsController(INewsService newsService)
```

## Parameters

newsService [INewsService](#)

## Methods

### DeleteNews(long)

Hard deletes news.

```
[HttpDelete("{id}")]
[Authorize(Roles = "Admin")]
public Task<IActionResult> DeleteNews(long id)
```

## Parameters

`id long`

## Returns

`Task<IActionResult>`

## GetNewsList()

Gets list of latest news. List is ordered by creation datetime. Change settings value for `numberOfNewsRetrieved` to increase or decrease amount retrieved.

```
[HttpGet]  
[Authorize]  
public Task<IActionResult> GetNewsList()
```

## Returns

`Task<IActionResult>`

## GetNewsListAll()

Gets list of latest news. List is ordered by creation datetime. Change settings value for `numberOfNewsRetrieved` to increase or decrease amount retrieved.

```
[HttpGet("all")]  
[Authorize(Roles = "Admin")]  
public Task<IActionResult> GetNewsListAll()
```

## Returns

`Task<IActionResult>`

## PostNews(NewsDTO)

Create news Title is required

```
[HttpPost]  
[Authorize(Roles = "Admin")]  
public Task<IActionResult> PostNews(NewsDTO news)
```

Parameters

news [NewsDTO](#)

Returns

[Task](#)<[IActionResult](#)>

## UpdateNews(NewsDTO)

Updates news. Use NewsDTO Id field to specify what news is going to be updated

```
[HttpPut]  
[Authorize(Roles = "Admin")]  
public Task<IActionResult> UpdateNews(NewsDTO news)
```

Parameters

news [NewsDTO](#)

Returns

[Task](#)<[IActionResult](#)>

# Class ReservationsController

Namespace: [Backend\\_Reservation\\_Service.Controllers](#)

Assembly: Backend\_Reservation\_Service.dll

Controller for handling reservation-related actions.

```
[Route("api/[controller]")]
[ApiController]
[Authorize]
public class ReservationsController : Controller, IActionFilter, IAsyncActionFilter,
IFilterMetadata, IDisposable
```

## Inheritance

[object](#) ← [ControllerBase](#) ← [Controller](#) ← ReservationsController

## Implements

[IActionFilter](#), [IAsyncActionFilter](#), [IFilterMetadata](#), [IDisposable](#)

## Inherited Members

[Controller.View\(\)](#), [Controller.View\(string\)](#), [Controller.View\(object\)](#),  
[Controller.View\(string, object\)](#), [Controller.PartialView\(\)](#), [Controller.PartialView\(string\)](#),  
[Controller.PartialView\(object\)](#), [Controller.PartialView\(string, object\)](#),  
[Controller.ViewComponent\(string\)](#), [Controller.ViewComponent\(Type\)](#),  
[Controller.ViewComponent\(string, object\)](#), [Controller.ViewComponent\(Type, object\)](#),  
[Controller.Json\(object\)](#), [Controller.Json\(object, object\)](#),  
[Controller.OnActionExecuting\(ActionExecutingContext\)](#),  
[Controller.OnActionExecuted\(ActionExecutedContext\)](#),  
[Controller.OnActionExecutionAsync\(ActionExecutingContext, ActionExecutionDelegate\)](#),  
[Controller.Dispose\(\)](#), [Controller.Dispose\(bool\)](#), [Controller.ViewData](#),  
[Controller.TempData](#), [Controller.ViewBag](#), [ControllerBase.StatusCode\(int\)](#),  
[ControllerBase.StatusCode\(int, object\)](#), [ControllerBase.Content\(string\)](#),  
[ControllerBase.Content\(string, string\)](#), [ControllerBase.Content\(string, string, Encoding\)](#),  
[ControllerBase.Content\(string, MediaTypeHeaderValue\)](#), [ControllerBase.NoContent\(\)](#),  
[ControllerBase.Ok\(\)](#), [ControllerBase.Ok\(object\)](#), [ControllerBase.Redirect\(string\)](#),  
[ControllerBase.RedirectPermanent\(string\)](#),  
[ControllerBase.RedirectPreserveMethod\(string\)](#),  
[ControllerBase.RedirectPermanentPreserveMethod\(string\)](#),  
[ControllerBase.LocalRedirect\(string\)](#), [ControllerBase.LocalRedirectPermanent\(string\)](#),  
[ControllerBase.LocalRedirectPreserveMethod\(string\)](#),

[ControllerBase.LocalRedirectPermanentPreserveMethod\(string\)](#)  ,  
 [ControllerBase.RedirectToAction\(\)](#)  ,  [ControllerBase.RedirectToAction\(string\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, object\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string\)](#)  ,  [ControllerBase.RedirectToRoute\(object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanentPreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string\)](#)  ,  [ControllerBase.RedirectToPage\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.File\(byte\[\], string\)](#)  ,  [ControllerBase.File\(byte\[\], string, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string\)](#)  ,

[ControllerBase.File\(byte\[\], string, string, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string\)](#)  ,  [ControllerBase.File\(Stream, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(string, string\)](#)  ,  [ControllerBase.File\(string, string, bool\)](#)  ,  
 [ControllerBase.File\(string, string, string\)](#)  ,  [ControllerBase.File\(string, string, string, bool\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.Unauthorized\(\)](#)  ,  [ControllerBase.Unauthorized\(object\)](#)  ,  
 [ControllerBase.NotFound\(\)](#)  ,  [ControllerBase.NotFound\(object\)](#)  ,  
 [ControllerBase.BadRequest\(\)](#)  ,  [ControllerBase.BadRequest\(object\)](#)  ,  
 [ControllerBase.BadRequest\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.UnprocessableEntity\(\)](#)  ,  [ControllerBase.UnprocessableEntity\(object\)](#)  ,  
 [ControllerBase.UnprocessableEntity\(ModelStateDictionary\)](#)  ,  [ControllerBase.Conflict\(\)](#)  ,  
 [ControllerBase.Conflict\(object\)](#)  ,  [ControllerBase.Conflict\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.Problem\(string, string, int?, string, string\)](#)  ,  
 [ControllerBase.Problem\(string, string, int?, string, string, IDictionary<string, object>\)](#)  ,  
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#)  ,  
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.ValidationProblem\(\)](#)  ,

[ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#)

,

[ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary, IDictionary<string, object>\)](#)

,

[ControllerBase.Created\(\)](#)  ,  [ControllerBase.Created\(string, object\)](#)  ,

[ControllerBase.Created\(Uri, object\)](#)  ,  [ControllerBase.CreatedAtAction\(string, object\)](#)  ,

[ControllerBase.CreatedAtAction\(string, object, object\)](#)  ,

[ControllerBase.CreatedAtAction\(string, string, object, object\)](#)  ,

[ControllerBase.CreatedAtRoute\(string, object\)](#)  ,

[ControllerBase.CreatedAtRoute\(object, object\)](#)  ,

[ControllerBase.CreatedAtRoute\(string, object, object\)](#)  ,  [ControllerBase.Accepted\(\)](#)  ,

[ControllerBase.Accepted\(object\)](#)  ,  [ControllerBase.Accepted\(Uri\)](#)  ,

[ControllerBase.Accepted\(string\)](#)  ,  [ControllerBase.Accepted\(string, object\)](#)  ,

[ControllerBase.Accepted\(Uri, object\)](#)  ,  [ControllerBase.AcceptedAtAction\(string\)](#)  ,

[ControllerBase.AcceptedAtAction\(string, string\)](#)  ,

[ControllerBase.AcceptedAtAction\(string, object\)](#)  ,

[ControllerBase.AcceptedAtAction\(string, string, object\)](#)  ,

[ControllerBase.AcceptedAtAction\(string, object, object\)](#)  ,

[ControllerBase.AcceptedAtAction\(string, string, object, object\)](#)  ,

[ControllerBase.AcceptedAtRoute\(object\)](#)  ,  [ControllerBase.AcceptedAtRoute\(string\)](#)  ,

[ControllerBase.AcceptedAtRoute\(string, object\)](#)  ,

[ControllerBase.AcceptedAtRoute\(object, object\)](#)  ,

[ControllerBase.AcceptedAtRoute\(string, object, object\)](#)  ,  [ControllerBase.Challenge\(\)](#)  ,

[ControllerBase.Challenge\(params string\[\]\)](#)  ,

[ControllerBase.Challenge\(AuthenticationProperties\)](#)  ,

[ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#)  ,

[ControllerBase.Forbid\(\)](#)  ,  [ControllerBase.Forbid\(params string\[\]\)](#)  ,

[ControllerBase.Forbid\(AuthenticationProperties\)](#)  ,

[ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#)  ,

[ControllerBase.SignIn\(ClaimsPrincipal\)](#)  ,  [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#)  ,

[ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#)  ,

[ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#)  ,

[ControllerBase.SignOut\(\)](#)  ,  [ControllerBase.SignOut\(AuthenticationProperties\)](#)  ,

[ControllerBase.SignOut\(params string\[\]\)](#)  ,

[ControllerBase.SignOut\(AuthenticationProperties, params string\[\]\)](#)  ,

[ControllerBase.TryUpdateModelAsync<TModel>\(TModel\)](#)  ,

[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string\)](#)  ,

[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#)  ,

[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\] \)](#)  ,

[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string\)](#) ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,  
 [ControllerBase.TryValidateModel\(object\)](#) ,  
 [ControllerBase.TryValidateModel\(object, string\)](#) , ControllerBase.HttpContext ,  
ControllerBase.Request , ControllerBase.Response , ControllerBase.RouteData ,  
ControllerBase.ModelState , ControllerBase.ControllerContext ,  
ControllerBase.MetadataProvider , ControllerBase.ModelBinderFactory ,  
ControllerBase.Url , ControllerBase.ObjectValidator ,  
ControllerBase.ProblemDetailsFactory , ControllerBase.User , ControllerBase.Empty ,  
object.Equals(object) , object.Equals(object, object) , object.GetHashCode() ,  
object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) ,  
object.ToString()

## Constructors

### ReservationsController(IReservationService)

Initializes a new instance of the [ReservationsController](#) class.

```
public ReservationsController(IReservationService reservationService)
```

#### Parameters

reservationService [IReservationService](#)

The reservation service.

## Methods

### CreateReservationAsync(CreateReservationDTO)

Creates a new reservation. Overlapping reservations and start- and enddate are checked.

```
[HttpPost("create")]
public Task<IActionResult> CreateReservationAsync(CreateReservationDTO
createReservationDTO)
```

## Parameters

createReservationDTO [CreateReservationDTO](#)

The reservation details.

## Returns

[Task](#)<[IActionResult](#)>

An [IActionResult](#) representing the result of the action.

## Remarks

User identity is required for this action, and the user must be authenticated. The service gets the username from the claims and uses it to create the reservation.

Example request:

```
POST /api/reservations/create
{
    "vehicleId": 1,
    "reservationStart": "2023-10-01",
    "reservationEnd": "2023-10-10"
}
```

Example responses:

200 OK

```
{
    "statusMessage": "Varauksen luominen onnist",
    "reservationDto": { ... }
}
```

400 Bad Request

```
{
    "statusMessage": "Error message"
```

```
}
```

401 Unauthorized

```
{
```

    "statusMessage": "Pääsy evätty!"

```
}
```

## DeleteReservationAsync(long)

Deletes a reservation.

```
[HttpDelete("delete/{deleteId}")]
public Task<IActionResult> DeleteReservationAsync(long deleteId)
```

### Parameters

deleteId [long](#)

### Returns

[Task](#)<[IActionResult](#)>

### Remarks

User must be authenticated and authorized to delete the reservation. Only the user who created the reservation can delete it. User with the "Admin" role can delete any reservation.

Example request:

```
DELETE /api/reservations/1
```

The ID of the reservation to delete. An [IActionResult](#) representing the result of the action.

## GetAllReservationsAsync(int, int, bool, int?)

Gets all reservations with pagination. Admin privileges needed.

```
[HttpGet("all")]
[Authorize(Roles = "Admin")]
public Task<IActionResult> GetAllReservationsAsync(int itemsPerPage, int pageNumber,
bool orderDesc, int? deletedIncludeMode)
```

## Parameters

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

deletedIncludeMode [int](#)?

## Returns

[Task](#)<[IActionResult](#)>

## Remarks

You must be authenticated and the user must be authorized with admin role to use this action.

Example request:

```
GET /api/reservations/all?
itemsPerPage=10&pageNumber=1&orderDesc=true&deletedIncludeMode=0
```

The number of reservations to retrieve per page. The page number to retrieve. Whether to order the results in descending order. The mode to include deleted reservations (0: include - all reservations, 1: exclude - active reservations, 2: only deleted). An [IActionResult](#) representing the result of the action.

## GetReservationDTOByIdAsync(long)

Gets a reservation by its ID.

```
[HttpGet("by/{reservationId}")]
public Task<ActionResult<ReservationDTO>> GetReservationDTOByIdAsync(long
```

```
reservationId)
```

## Parameters

reservationId [long](#)

## Returns

[Task](#) <[ActionResult](#) <[ReservationDTO](#)>>

## Remarks

Example request:

```
GET /api/reservations/by/1
```

The ID of the reservation to retrieve. An [ActionResult<TValue>](#) representing the result of the action.

## GetReservationsByUserAsync(int, int, bool, bool, bool)

Gets the reservations of the authenticated user with pagination.

```
[HttpGet("by/user")]
public Task<IActionResult> GetReservationsByUserAsync(int itemsPerPage, int
pageNumber, bool orderDesc, bool onlyActive, bool onlyPast)
```

## Parameters

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

onlyActive [bool](#)

onlyPast [bool](#)

## Returns

[Task](#) <[IActionResult](#)>

## Remarks

User identity is required for this action, and the user must be authenticated. User identity is retrieved from the claims and used to get the reservations.

Example request:

```
GET /api/reservations/by/user?  
itemsPerPage=10&pageNumber=1&orderDesc=true&onlyActive=true&onlyPast=false
```

The number of reservations to retrieve per page. The page number to retrieve. Whether to order the results in descending order. Whether to include only active reservations. Whether to include only past reservations. An [IActionResult](#) representing the result of the action.

## GetReservationsByVehicleAsync(long, int, int, bool)

Gets the active (EndDate > DateTime.Now) reservations for a specific vehicle with pagination.

```
[HttpGet("by/vehicle/{vehicleId}")]  
public Task<IActionResult> GetReservationsByVehicleAsync(long vehicleId, int  
itemsPerPage, int pageNumber, bool orderDesc)
```

## Parameters

vehicleId [long](#)

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

## Returns

[Task](#) <[IActionResult](#)>

## Remarks

Example request:

```
GET /api/reservations/byVehicle/1?itemsPerPage=10&pageNumber=1&orderDesc=true
```

The ID of the vehicle. The number of reservations to retrieve per page. The page number to retrieve. Whether to order the results in descending order. An [IActionResult](#) representing the result of the action, a list of reservations and a statusmessage

# Class UploadFilesController

Namespace: [Backend\\_Reservation\\_Service.Controllers](#)

Assembly: Backend\_Reservation\_Service.dll

Controller for uploading files

```
[Route("api/[controller]")]
[ApiController]
[Authorize(Roles = "Admin")]
public class UploadFilesController : ControllerBase
```

## Inheritance

[object](#) ← [ControllerBase](#) ← UploadFilesController

## Inherited Members

[ControllerBase.StatusCode\(int\)](#) , [ControllerBase.StatusCode\(int, object\)](#) ,  
[ControllerBase.Content\(string\)](#) , [ControllerBase.Content\(string, string\)](#) ,  
[ControllerBase.Content\(string, string, Encoding\)](#) ,  
[ControllerBase.Content\(string, MediaTypeHeaderValue\)](#) , [ControllerBase.NoContent\(\)](#) ,  
[ControllerBase.Ok\(\)](#) , [ControllerBase.Ok\(object\)](#) , [ControllerBase.Redirect\(string\)](#) ,  
[ControllerBase.RedirectPermanent\(string\)](#) ,  
[ControllerBase.RedirectPreserveMethod\(string\)](#) ,  
[ControllerBase.RedirectPermanentPreserveMethod\(string\)](#) ,  
[ControllerBase.LocalRedirect\(string\)](#) , [ControllerBase.LocalRedirectPermanent\(string\)](#) ,  
[ControllerBase.LocalRedirectPreserveMethod\(string\)](#) ,  
[ControllerBase.LocalRedirectPermanentPreserveMethod\(string\)](#) ,  
[ControllerBase.RedirectToAction\(\)](#) , [ControllerBase.RedirectToAction\(string\)](#) ,  
[ControllerBase.RedirectToAction\(string, object\)](#) ,  
[ControllerBase.RedirectToAction\(string, string\)](#) ,  
[ControllerBase.RedirectToAction\(string, string, object\)](#) ,  
[ControllerBase.RedirectToAction\(string, string, string\)](#) ,  
[ControllerBase.RedirectToAction\(string, string, object, string\)](#) ,  
[ControllerBase.RedirectToActionPreserveMethod\(string, string, object, string\)](#) ,  
[ControllerBase.RedirectToActionPermanent\(string\)](#) ,  
[ControllerBase.RedirectToActionPermanent\(string, object\)](#) ,  
[ControllerBase.RedirectToActionPermanent\(string, string\)](#) ,  
[ControllerBase.RedirectToActionPermanent\(string, string, string\)](#) ,  
[ControllerBase.RedirectToActionPermanent\(string, string, object\)](#) ,

[ControllerBase.RedirectToActionPermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string\)](#)  ,  [ControllerBase.RedirectToRoute\(object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanentPreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string\)](#)  ,  [ControllerBase.RedirectToPage\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.File\(byte\[\], string\)](#)  ,  [ControllerBase.File\(byte\[\], string, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string\)](#)  ,  [ControllerBase.File\(Stream, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(string, string\)](#)  ,  [ControllerBase.File\(string, string, bool\)](#)  ,

[ControllerBase.File\(string, string, string\)](#)  ,  [ControllerBase.File\(string, string, string, bool\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.Unauthorized\(\)](#)  ,  [ControllerBase.Unauthorized\(object\)](#)  ,  
 [ControllerBase.NotFound\(\)](#)  ,  [ControllerBase.NotFound\(object\)](#)  ,  
 [ControllerBase.BadRequest\(\)](#)  ,  [ControllerBase.BadRequest\(object\)](#)  ,  
 [ControllerBase.BadRequest\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.UnprocessableEntity\(\)](#)  ,  [ControllerBase.UnprocessableEntity\(object\)](#)  ,  
 [ControllerBase.UnprocessableEntity\(ModelStateDictionary\)](#)  ,  [ControllerBase.Conflict\(\)](#)  ,  
 [ControllerBase.Conflict\(object\)](#)  ,  [ControllerBase.Conflict\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.Problem\(string, string, int?, string, string\)](#)  ,  
 [ControllerBase.Problem\(string, string, int?, string, string, IDictionary<string, object>\)](#)  ,  
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#)  ,  
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.ValidationProblem\(\)](#)  ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#)  ,  
  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary, IDictionary<string, object>\)](#)  ,  
 [ControllerBase.Created\(\)](#)  ,  [ControllerBase.Created\(string, object\)](#)  ,  
 [ControllerBase.Created\(Uri, object\)](#)  ,  [ControllerBase.CreatedAtAction\(string, object\)](#)  ,  
 [ControllerBase.CreatedAtAction\(string, object, object\)](#)  ,  
 [ControllerBase.CreatedAtAction\(string, string, object, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(string, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(object, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(string, object, object\)](#)  ,  [ControllerBase.Accepted\(\)](#)  ,  
 [ControllerBase.Accepted\(object\)](#)  ,  [ControllerBase.Accepted\(Uri\)](#)  ,  
 [ControllerBase.Accepted\(string\)](#)  ,  [ControllerBase.Accepted\(string, object\)](#)  ,  
 [ControllerBase.Accepted\(Uri, object\)](#)  ,  [ControllerBase.AcceptedAtAction\(string\)](#)  ,

[ControllerBase.AcceptedAtAction\(string, string\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, string, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, object, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, string, object, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(object\)](#)  ,  [ControllerBase.AcceptedAtRoute\(string\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(string, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(object, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(string, object, object\)](#)  ,  [ControllerBase.Challenge\(\)](#)  ,  
 [ControllerBase.Challenge\(params string\[\]\)](#)  ,  
 [ControllerBase.Challenge\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.Forbid\(\)](#)  ,  [ControllerBase.Forbid\(params string\[\]\)](#)  ,  
 [ControllerBase.Forbid\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal\)](#)  ,  [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#)  ,  
 [ControllerBase.SignOut\(\)](#)  ,  [ControllerBase.SignOut\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.SignOut\(params string\[\]\)](#)  ,  
 [ControllerBase.SignOut\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\] \)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\] \)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#)  ,  
 [ControllerBase.TryValidateModel\(object\)](#)  ,  
 [ControllerBase.TryValidateModel\(object, string\)](#)  ,  [ControllerBase.HttpContext](#)  ,  
 [ControllerBase.Request](#)  ,  [ControllerBase.Response](#)  ,  [ControllerBase.RouteData](#)  ,  
 [ControllerBase.ModelState](#)  ,  [ControllerBase.ControllerContext](#)  ,  
 [ControllerBase.MetadataProvider](#)  ,  [ControllerBase.ModelBinderFactory](#)  ,

[ControllerBase.Url](#) ,  [ControllerBase.ObjectValidator](#) ,  
 [ControllerBase.ProblemDetailsFactory](#) ,  [ControllerBase.User](#) ,  [ControllerBase.Empty](#) ,  
 [object.Equals\(object\)](#) ,  [object.Equals\(object, object\)](#) ,  [object.GetHashCode\(\)](#) ,  
 [object.GetType\(\)](#) ,  [object.MemberwiseClone\(\)](#) ,  [object.ReferenceEquals\(object, object\)](#) ,  
 [object.ToString\(\)](#)

## Constructors

UploadFilesController(IUploadService, IUserRepository, IVehicleRepository)

```
public UploadFilesController(IUploadService uploadService, IUserRepository  
userRepository, IVehicleRepository vehicleRepository)
```

## Parameters

uploadService [IUploadService](#)

userRepository [IUserRepository](#)

vehicleRepository [IVehicleRepository](#)

## Methods

UploadFileAsync(UploadFilePostDTO)

Uploads a file to the server.

```
[HttpPost]  
[Route("upload")]  
[RequestSizeLimit(5242880)]  
public Task<IActionResult> UploadFileAsync(UploadFilePostDTO uploadFile)
```

## Parameters

uploadFile [UploadFilePostDTO](#)

The file to be uploaded along with its metadata.

## Returns

[Task](#) <[IActionResult](#)>

Returns an IActionResult:

- 200 OK: If the file is successfully uploaded. Returns a message and the uploaded file metadata.
- 400 Bad Request: If the request is invalid, the user or vehicle is not found, or the file upload fails. Returns an error message.
- 401 Unauthorized: If the user is not authenticated. Returns an error message.

# Class VehiclesController

Namespace: [Backend\\_Reservation\\_Service.Controllers](#)

Assembly: Backend\_Reservation\_Service.dll

Controller for managing vehicles

```
[Route("api/[controller]")]
[ApiController]
public class VehiclesController : Controller, IActionFilter, IAsyncActionFilter,
IFilterMetadata, IDisposable
```

## Inheritance

[object](#) ← [ControllerBase](#) ← [Controller](#) ← VehiclesController

## Implements

[IActionFilter](#), [IAsyncActionFilter](#), [IFilterMetadata](#), [IDisposable](#)

## Inherited Members

[Controller.View\(\)](#), [Controller.View\(string\)](#), [Controller.View\(object\)](#),  
[Controller.View\(string, object\)](#), [Controller.PartialView\(\)](#), [Controller.PartialView\(string\)](#),  
[Controller.PartialView\(object\)](#), [Controller.PartialView\(string, object\)](#),  
[Controller.ViewComponent\(string\)](#), [Controller.ViewComponent\(Type\)](#),  
[Controller.ViewComponent\(string, object\)](#), [Controller.ViewComponent\(Type, object\)](#),  
[Controller.Json\(object\)](#), [Controller.Json\(object, object\)](#),  
[Controller.OnActionExecuting\(ActionExecutingContext\)](#),  
[Controller.OnActionExecuted\(ActionExecutedContext\)](#),  
[Controller.OnActionExecutionAsync\(ActionExecutingContext, ActionExecutionDelegate\)](#),  
[Controller.Dispose\(\)](#), [Controller.Dispose\(bool\)](#), [Controller.ViewData](#),  
[Controller.TempData](#), [Controller.ViewBag](#), [ControllerBase.StatusCode\(int\)](#),  
[ControllerBase.StatusCode\(int, object\)](#), [ControllerBase.Content\(string\)](#),  
[ControllerBase.Content\(string, string\)](#), [ControllerBase.Content\(string, string, Encoding\)](#),  
[ControllerBase.Content\(string, MediaTypeHeaderValue\)](#), [ControllerBase.NoContent\(\)](#),  
[ControllerBase.Ok\(\)](#), [ControllerBase.Ok\(object\)](#), [ControllerBase.Redirect\(string\)](#),  
[ControllerBase.RedirectPermanent\(string\)](#),  
[ControllerBase.RedirectPreserveMethod\(string\)](#),  
[ControllerBase.RedirectPermanentPreserveMethod\(string\)](#),  
[ControllerBase.LocalRedirect\(string\)](#), [ControllerBase.LocalRedirectPermanent\(string\)](#),  
[ControllerBase.LocalRedirectPreserveMethod\(string\)](#),  
[ControllerBase.LocalRedirectPermanentPreserveMethod\(string\)](#),

[ControllerBase.RedirectToAction\(\)](#)  ,  [ControllerBase.RedirectToAction\(string\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, object\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToAction\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToActionPermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToActionPermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string\)](#)  ,  [ControllerBase.RedirectToRoute\(object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoute\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanent\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToRoutePermanentPreserveMethod\(string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string\)](#)  ,  [ControllerBase.RedirectToPage\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, object\)](#)  ,  
 [ControllerBase.RedirectToPage\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, object\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanent\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.RedirectToPagePermanentPreserveMethod\(string, string, object, string\)](#)  ,  
 [ControllerBase.File\(byte\[\], string\)](#)  ,  [ControllerBase.File\(byte\[\], string, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, bool\)](#)  ,

[ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string\)](#)  ,  [ControllerBase.File\(Stream, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(string, string\)](#)  ,  [ControllerBase.File\(string, string, bool\)](#)  ,  
 [ControllerBase.File\(string, string, string\)](#)  ,  [ControllerBase.File\(string, string, string, bool\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#)  ,  
 [ControllerBase.Unauthorized\(\)](#)  ,  [ControllerBase.Unauthorized\(object\)](#)  ,  
 [ControllerBase.NotFound\(\)](#)  ,  [ControllerBase.NotFound\(object\)](#)  ,  
 [ControllerBase.BadRequest\(\)](#)  ,  [ControllerBase.BadRequest\(object\)](#)  ,  
 [ControllerBase.BadRequest\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.UnprocessableEntity\(\)](#)  ,  [ControllerBase.UnprocessableEntity\(object\)](#)  ,  
 [ControllerBase.UnprocessableEntity\(ModelStateDictionary\)](#)  ,  [ControllerBase.Conflict\(\)](#)  ,  
 [ControllerBase.Conflict\(object\)](#)  ,  [ControllerBase.Conflict\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.Problem\(string, string, int?, string, string\)](#)  ,  
 [ControllerBase.Problem\(string, string, int?, string, string, IDictionary<string, object>\)](#)  ,  
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#)  ,  
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#)  ,  
 [ControllerBase.ValidationProblem\(\)](#)  ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#)  ,

[ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary, IDictionary<string, object>\)](#)  ,  
 [ControllerBase.Created\(\)](#)  ,  [ControllerBase.Created\(string, object\)](#)  ,  
 [ControllerBase.Created\(Uri, object\)](#)  ,  [ControllerBase.CreatedAtAction\(string, object\)](#)  ,  
 [ControllerBase.CreatedAtAction\(string, object, object\)](#)  ,  
 [ControllerBase.CreatedAtAction\(string, string, object, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(string, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(object, object\)](#)  ,  
 [ControllerBase.CreatedAtRoute\(string, object, object\)](#)  ,  [ControllerBase.Accepted\(\)](#)  ,  
 [ControllerBase.Accepted\(object\)](#)  ,  [ControllerBase.Accepted\(Uri\)](#)  ,  
 [ControllerBase.Accepted\(string\)](#)  ,  [ControllerBase.Accepted\(string, object\)](#)  ,  
 [ControllerBase.Accepted\(Uri, object\)](#)  ,  [ControllerBase.AcceptedAtAction\(string\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, string\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, string, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, object, object\)](#)  ,  
 [ControllerBase.AcceptedAtAction\(string, string, object, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(object\)](#)  ,  [ControllerBase.AcceptedAtRoute\(string\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(string, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(object, object\)](#)  ,  
 [ControllerBase.AcceptedAtRoute\(string, object, object\)](#)  ,  [ControllerBase.Challenge\(\)](#)  ,  
 [ControllerBase.Challenge\(params string\[\]\)](#)  ,  
 [ControllerBase.Challenge\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.Forbid\(\)](#)  ,  [ControllerBase.Forbid\(params string\[\]\)](#)  ,  
 [ControllerBase.Forbid\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal\)](#)  ,  [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#)  ,  
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#)  ,  
 [ControllerBase.SignOut\(\)](#)  ,  [ControllerBase.SignOut\(AuthenticationProperties\)](#)  ,  
 [ControllerBase.SignOut\(params string\[\]\)](#)  ,  
 [ControllerBase.SignOut\(AuthenticationProperties, params string\[\]\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\] \)](#)  ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#)  ,

[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#) ,  
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string\)](#) ,  
 [ControllerBase.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,  
 [ControllerBase.TryValidateModel\(object\)](#) ,  
 [ControllerBase.TryValidateModel\(object, string\)](#) , ControllerBase.HttpContext ,  
 ControllerBase.Request , ControllerBase.Response , ControllerBase.RouteData ,  
 ControllerBase.ModelState , ControllerBase.ControllerContext ,  
 ControllerBase.MetadataProvider , ControllerBase.ModelBinderFactory ,  
 ControllerBase.Url , ControllerBase.ObjectValidator ,  
 ControllerBase.ProblemDetailsFactory , ControllerBase.User , ControllerBase.Empty ,  
 object.Equals(object) , object.Equals(object, object) , object.GetHashCode() ,  
 object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) ,  
 object.ToString()

## Constructors

### VehiclesController(IVehicleService)

```
public VehiclesController(IVehicleService vehicleService)
```

## Parameters

vehicleService [IVehicleService](#)

## Methods

### DeleteVehicle(long)

Changes vehicle.deleted value as true.

```
[HttpDelete("deleteby/id={vehicleId}")]
[Authorize(Roles = "Admin")]
public Task<IActionResult> DeleteVehicle(long vehicleId)
```

Parameters

**vehicleId** [long](#)

Returns

[Task](#)<[IActionResult](#)>

## DeleteVehicle(string)

Changes vehicle.deleted value as true. Cannot delete generated nicknames (for now)

```
[HttpDelete("deleteby/name={vehicleName}")]
[Authorize(Roles = "Admin")]
public Task<IActionResult> DeleteVehicle(string vehicleName)
```

Parameters

**vehicleName** [string](#)

Returns

[Task](#)<[IActionResult](#)>

## GetDdMakeListAsync(string?, string?)

Get list of every unique vehicle make Can narrowdown using type and model

```
[HttpGet("ddlist_make")]
[Authorize]
public Task<ActionResult<List<string>>> GetDdMakeListAsync(string? type,
string? model)
```

Parameters

**type** [string](#)

**model** [string](#)

Returns

[Task](#) <[ActionResult](#) <[List](#) <[string](#)>>>

## GetDdModelListAsync(string?, string?)

Get list of every unique vehicle model Can narrowdown using type and make

```
[HttpGet("ddlist_model")]
[Authorize]
public Task<ActionResult<List<string>>> GetDdModelListAsync(string? type,
string? make)
```

Parameters

type [string](#)

make [string](#)

Returns

[Task](#) <[ActionResult](#) <[List](#) <[string](#)>>>

## GetDdTypeListAsync(string?, string?)

Get list of every unique vehicle type Can narrowdown using make and model

```
[HttpGet("ddlist_type")]
[Authorize]
public Task<ActionResult<List<string>>> GetDdTypeListAsync(string? make,
string? model)
```

Parameters

make [string](#)

model [string](#)

Returns

[Task](#) <[ActionResult](#)> <[List](#)<[string](#)>>>

## GetDelVehicleList(int, int?, bool?)

Get softdeleted vehicles ordered by deletion timestamp.

```
[HttpGet("dellist")]
[Authorize(Roles = "Admin")]
public Task<ActionResult<IEnumerable<Vehicle>>> GetDelVehicleList(int itemsPerPage,
int? page, bool? oldestFirst)
```

### Parameters

itemsPerPage [int](#)

page [int](#)?

oldestFirst [bool](#)?

### Returns

[Task](#) <[ActionResult](#)> <[IEnumerable](#)<[Vehicle](#)>>>

## GetVehicleById(long)

Retrieves a vehicle by its unique identifier. Cannot get vehicle marked as deleted

```
[HttpGet("by/id={id}")]
[Authorize]
public Task<ActionResult<VehicleDTO>> GetVehicleById(long id)
```

### Parameters

id [long](#)

The unique identifier of the vehicle.

### Returns

[Task](#) <[ActionResult](#)> <[VehicleDTO](#)>>

An [ActionResult](#) containing the vehicle details if found; otherwise, a NotFound result.

## GetVehicleByLicenseplate(string)

Retrieves vehicle using licenseplate Cannot get vehicle marked as deleted

```
[HttpGet("by/licenseplate={licenseplate}")]
[Authorize]
public Task<ActionResult<VehicleDTO>> GetVehicleByLicenseplate(string licenseplate)
```

Parameters

licenseplate [string](#)

Returns

[Task](#) <[ActionResult](#)> <[VehicleDTO](#)>>

## GetVehicleByNickname(string)

Retrieves vehicle using nickname Cannot get vehicle marked as deleted Cannot retrieve generated nicknames (for now)

```
[HttpGet("by/nickname={nickname}")]
[Authorize]
public Task<ActionResult<VehicleDTO>> GetVehicleByNickname(string nickname)
```

Parameters

nickname [string](#)

Returns

[Task](#) <[ActionResult](#)> <[VehicleDTO](#)>>

## GetVehicleList(string?, string?, string?, string?, DateOnly?, DateOnly?, bool?, int, int?)

About:

currently model only looks if said string is contained in model string.

```
[HttpGet("list")]
[Authorize]
public Task<ActionResult<IEnumerable<VehicleDTO>>> GetVehicleList(string? type,
string? make, string? model, string? year, DateOnly? dtmStart, DateOnly? dtmEnd,
bool? includeNotUsable, int itemsPerPage, int? page)
```

### Parameters

type [string](#)?

make [string](#)?

model [string](#)?

year [string](#)?

dtmStart [DateOnly](#)?

dtmEnd [DateOnly](#)?

includeNotUsable [bool](#)?

itemsPerPage [int](#)?

page [int](#)?

### Returns

[Task](#)<[ActionResult](#)<[IEnumerable](#)<[VehicleDTO](#)>>>

## GetVehicleListAsAdmin(bool?, bool?, bool?, string?, string?, string?, DateOnly?, DateOnly?, bool?, int, int?)

How to use Use "-" for empty and null search (applies to year, model) About:  
currently model only looks if said string is contained in model string. returns vehicle, not  
vehicleDto (Currently only difference is that there won't be pregen nickname if that value is  
null and this will include deleted field)

```
[HttpGet("List/admin")]
[Authorize(Roles = "Admin")]
public Task<ActionResult<IEnumerable<Vehicle>>> GetVehicleListAsAdmin(bool?
includeDeleted, bool? onlyEmptyDescriptions, bool? onlyEmptyNicknames, string? type,
string? make, string? model, string? year, DateOnly? dtmStart, DateOnly? dtmEnd,
bool? includeNotUsable, int itemsPerPage, int? page)
```

## Parameters

includeDeleted [bool](#)?

onlyEmptyDescriptions [bool](#)?

onlyEmptyNicknames [bool](#)?

type [string](#)

make [string](#)

model [string](#)

year [string](#)

dtmStart [DateOnly](#)?

dtmEnd [DateOnly](#)?

includeNotUsable [bool](#)?

itemsPerPage [int](#)

page [int](#)?

## Returns

[Task](#)<[ActionResult](#)<[IEnumerable](#)<[Vehicle](#)>>>

## PostVehicle(VehicleDTO)

Create new vehicle, Make and Type are required and if motorized vehicle type then unique LicensePlate is required. Nickname if given must be length of atleast 3 and cannot contain " " or "\_". Deleted is automatically filled NotUsable is automatically filled if left null. Motorized types ["car","van","truck","motorcycle","moped","atv","snowmobile"]

```
[HttpPost]  
[Authorize(Roles = "Admin")]  
public Task<ActionResult<VehicleDTO>> PostVehicle(VehicleDTO vehicle)
```

Parameters

vehicle [VehicleDTO](#)

Returns

[Task](#) <[ActionResult](#) <[VehicleDTO](#)>>

## RestoreSoftDelVehicle(int)

Restores softdeleted vehicle

```
[HttpPut("restoreId={id}")]
[Authorize(Roles = "Admin")]
public Task<ActionResult<VehicleDTO>> RestoreSoftDelVehicle(int id)
```

Parameters

id [int](#)

Returns

[Task](#) <[ActionResult](#) <[VehicleDTO](#)>>

## UpdateVehicle(VehicleDTO)

Update vehicle. Specify vehicle using vehicle.Id. Cannot change following values (id, vehicleType, make) Cannot update vehicle marked as deleted

```
[HttpPost]  
[Authorize(Roles = "Admin")]  
public Task<IActionResult> UpdateVehicle(VehicleDTO vehicle)
```

Parameters

vehicle [VehicleDTO](#)

Returns

[Task](#) <[IActionResult](#)>

# Namespace Backend\_Reservation\_Service.

## Data

### Classes

[ReservationServiceDbContext](#)

# Class ReservationServiceDbContext

Namespace: [Backend\\_Reservation\\_Service.Data](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class ReservationServiceDbContext : IdentityDbContext<User>,
IInfrastructure<IServiceProvider>, IDbContextDependencies, IDbSetCache,
IDbContextPoolable, IResettableService, IDisposable, IAsyncDisposable
```

## Inheritance

```
object ↳ DbContext ↳
IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>,
IdentityUserToken<string>> ↳
IdentityDbContext<User, IdentityRole, string, IdentityUserClaim<string>, IdentityUserRole<string>,
IdentityUserLogin<string>, IdentityRoleClaim<string>, IdentityUserToken<string>> ↳
IdentityDbContext<User, IdentityRole, string> ↳ IdentityDbContext<User> ↳
ReservationServiceDbContext
```

## Implements

```
IInfrastructure<IServiceProvider>, IDbContextDependencies, IDbSetCache,
IDbContextPoolable, IResettableService, IDisposable, IAsyncDisposable
```

## Inherited Members

```
IdentityDbContext<User, IdentityRole, string, IdentityUserClaim<string>,
IdentityUserRole<string>, IdentityUserLogin<string>, IdentityRoleClaim<string>,
IdentityUserToken<string>>.OnModelCreating(ModelBuilder) ,
IdentityDbContext<User, IdentityRole, string, IdentityUserClaim<string>,
IdentityUserRole<string>, IdentityUserLogin<string>, IdentityRoleClaim<string>,
IdentityUserToken<string>>.UserRoles ,
IdentityDbContext<User, IdentityRole, string, IdentityUserClaim<string>,
IdentityUserRole<string>, IdentityUserLogin<string>, IdentityRoleClaim<string>,
IdentityUserToken<string>>.Roles ,
IdentityDbContext<User, IdentityRole, string, IdentityUserClaim<string>,
IdentityUserRole<string>, IdentityUserLogin<string>, IdentityRoleClaim<string>,
IdentityUserToken<string>>.RoleClaims ,
IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>,
IdentityUserToken<string>>.OnModelCreating(ModelBuilder) ,
```

[IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>, IdentityUserToken<string>>.Users](#) ,  
[IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>, IdentityUserToken<string>>.UserClaims](#) ,  
[IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>, IdentityUserToken<string>>.UserLogins](#) ,  
[IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>, IdentityUserToken<string>>.UserTokens](#) ,  
[IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>, IdentityUserToken<string>>.SchemaVersion](#) ,  
[DbContext.Set< TEntity >\(\)](#) , [DbContext.Set< TEntity >\(string\)](#) ,  
[DbContext.OnConfiguring\(DbContextOptionsBuilder\)](#) ,  
[DbContext.ConfigureConventions\(ModelConfigurationBuilder\)](#) ,  
[DbContext.SaveChanges\(\)](#) , [DbContext.SaveChanges\(bool\)](#) ,  
[DbContext.SaveChangesAsync\(CancellationToken\)](#) ,  
[DbContext.SaveChangesAsync\(bool, CancellationToken\)](#) , [DbContext.Dispose\(\)](#) ,  
[DbContext.DisposeAsync\(\)](#) , [DbContext.Entry< TEntity >\(TEntity\)](#) ,  
[DbContext.Entry\(object\)](#) , [DbContext.Add< TEntity >\(TEntity\)](#) ,  
[DbContext.AddAsync< TEntity >\(TEntity, CancellationToken\)](#) ,  
[DbContext.Attach< TEntity >\(TEntity\)](#) , [DbContext.Update< TEntity >\(TEntity\)](#) ,  
[DbContext.Remove< TEntity >\(TEntity\)](#) , [DbContext.Add\(object\)](#) ,  
[DbContext.AddAsync\(object, CancellationToken\)](#) , [DbContext.Attach\(object\)](#) ,  
[DbContext.Update\(object\)](#) , [DbContext.Remove\(object\)](#) ,  
[DbContext.AddRange\(params object\[\]\)](#) , [DbContext.AddRangeAsync\(params object\[\]\)](#) ,  
[DbContext.AttachRange\(params object\[\]\)](#) , [DbContext.UpdateRange\(params object\[\]\)](#) ,  
[DbContext.RemoveRange\(params object\[\]\)](#) ,  
[DbContext.AddRange\(IEnumerable<object>\)](#) ,  
[DbContext.AddRangeAsync\(IEnumerable<object>, CancellationToken\)](#) ,  
[DbContext.AttachRange\(IEnumerable<object>\)](#) ,  
[DbContext.UpdateRange\(IEnumerable<object>\)](#) ,  
[DbContext.RemoveRange\(IEnumerable<object>\)](#) ,  
[DbContext.Find\(Type, params object\[\]\)](#) , [DbContext.FindAsync\(Type, params object\[\]\)](#) ,  
[DbContext.FindAsync\(Type, object\[\], CancellationToken\)](#) ,  
[DbContext.Find< TEntity >\(params object\[\]\)](#) ,  
[DbContext.FindAsync< TEntity >\(params object\[\]\)](#) ,  
[DbContext.FindAsync< TEntity >\(object\[\], CancellationToken\)](#) ,  
[DbContext.FromExpression< TResult >\(Expression< Func< IQueryble< TResult >> \)](#) ,  
[DbContext.Database](#) , [DbContext.ChangeTracker](#) , [DbContext.Model](#) ,  
[DbContext.ContextId](#) , [DbContext.SavingChanges](#) , [DbContext.SavedChanges](#) ,  
[DbContext.SaveChangesFailed](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,

[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### ReservationServiceDbContext(DbContextOptions<ReservationServiceDbContext>)

```
public ReservationServiceDbContext(DbContextOptions<ReservationServiceDbContext>  
options)
```

#### Parameters

options [DbContextOptions](#)<[ReservationServiceDbContext](#)>

## Properties

### News

```
public DbSet<News> News { get; set; }
```

#### Property Value

[DbSet](#)<[News](#)>

### Reservations

```
public DbSet<Reservation> Reservations { get; set; }
```

#### Property Value

[DbSet](#)<[Reservation](#)>

# Uploads

```
public DbSet<UploadFile> Uploads { get; set; }
```

Property Value

[DbSet](#)<[UploadFile](#)>

# Users

```
public DbSet<User> Users { get; set; }
```

Property Value

[DbSet](#)<[User](#)>

# Vehicles

```
public DbSet<Vehicle> Vehicles { get; set; }
```

Property Value

[DbSet](#)<[Vehicle](#)>

# Methods

## OnModelCreating(ModelBuilder)

Configures the schema needed for the identity framework.

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

Parameters

`modelBuilder` [ModelBuilder](#)



# Namespace Backend\_Reservation\_Service. Data.Models

## Classes

[CreateReservationDTO](#)

[FullReservationDTO](#)

[FullUserDTO](#)

Data Transfer Object for full user details.

[LogOut](#)

[Login](#)

[News](#)

[NewsDTO](#)

[PaginationDTO](#)

Represents pagination details for a collection of items.

[Register](#)

Represents a registration model with user details.

[Reservation](#)

[ReservationDTO](#)

[Settings](#)

Represents the settings for various constraints in the application.

[UpdateUserDTO](#)

Data Transfer Object for updating user details.

[UpdateUserPwdDTO](#)

[UploadFile](#)

Represents a file uploaded by a user.

[UploadFileDialog](#)

[UploadFilePostDTO](#)

[User](#)

Represents a user in the system.

## UserDTO

Data Transfer Object for User entity.

## UserRole

## Vehicle

## VehicleDTO

# Class CreateReservationDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class CreateReservationDTO
```

## Inheritance

[object](#) ← CreateReservationDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### ReservationEnd

```
[Required(ErrorMessage = "{0}: required / {0}: vaaditaan")]
public DateOnly ReservationEnd { get; set; }
```

Property Value

[DateOnly](#)

### ReservationStart

```
[Required(ErrorMessage = "{0}: required / {0}: vaaditaan")]
public DateOnly ReservationStart { get; set; }
```

Property Value

[DateOnly](#)

## VehicleId

```
[Required(ErrorMessage = "{0}: required / {0}: vaaditaan")]
public long? VehicleId { get; set; }
```

Property Value

[long](#)?

# Class FullReservationDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class FullReservationDTO
```

## Inheritance

[object](#) ← FullReservationDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Created

```
public DateTime Created { get; set; }
```

Property Value

[DateTime](#)

### DeletionDate

```
public DateTime? DeletionDate { get; set; }
```

Property Value

[DateTime](#)?

### Id

```
public long Id { get; set; }
```

Property Value

[long](#) ↗

## IsDeleted

```
public bool IsDeleted { get; set; }
```

Property Value

[bool](#) ↗

## ReservationEnd

```
public DateOnly ReservationEnd { get; set; }
```

Property Value

[DateOnly](#) ↗

## ReservationStart

```
public DateOnly ReservationStart { get; set; }
```

Property Value

[DateOnly](#) ↗

## Username

```
public string? Username { get; set; }
```

Property Value

[string](#) ↗

## VehicleId

```
public string? VehicleId { get; set; }
```

Property Value

[string](#) ↗

# Class FullUserDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

Data Transfer Object for full user details.

```
public class FullUserDTO
```

## Inheritance

[object](#) ← FullUserDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### DeletionDate

Gets or sets the date when the user was marked deleted.

```
public DateTime? DeletionDate { get; set; }
```

### Property Value

[DateTime](#)?

The date when the user was deleted, system sets this.

### Email

Gets or sets the email of the user.

```
public string Email { get; set; }
```

Property Value

[string](#)

## FirstName

Gets or sets the first name of the user.

```
public string FirstName { get; set; }
```

Property Value

[string](#)

## Id

Gets or sets the unique identifier for the user.

```
public string Id { get; set; }
```

Property Value

[string](#)

## IsDeleted

Gets or sets a value indicating whether the user is deleted.

```
public bool IsDeleted { get; set; }
```

Property Value

[bool](#)

**true** if the user is deleted; otherwise, **false**, system sets this.

## JoinDate

Gets or sets the date when the user joined.

```
public DateTime JoinDate { get; set; }
```

Property Value

[DateTime](#)

## LastLogin

Gets or sets the date when the user last logged in.

```
public DateTime? LastLogin { get; set; }
```

Property Value

[DateTime?](#)

## LastName

Gets or sets the last name of the user.

```
public string LastName { get; set; }
```

Property Value

[string](#)

## Roles

Gets the role of the user.

```
public string? Roles { get; set; }
```

## PropertyValue

[string](#) ↗

The role of the user from rolemanager.

## Updated

Gets or sets the date when the user was last updated.

```
public DateTime? Updated { get; set; }
```

## PropertyValue

[DateTime](#) ↗?

## Username

Gets or sets the username of the user.

```
public string Username { get; set; }
```

## PropertyValue

[string](#) ↗

# Class LogOut

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class LogOut
```

## Inheritance

[object](#) ← LogOut

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Properties

## Username

```
public string Username { get; set; }
```

## Property Value

[string](#)

# Class Login

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class Login
```

## Inheritance

[object](#) ← Login

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Password

```
public string Password { get; set; }
```

#### Property Value

[string](#)

### Username

```
public string Username { get; set; }
```

#### Property Value

[string](#)

# Class News

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class News
```

## Inheritance

[object](#) ← News

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### CreatedDate

```
public DateTime CreatedDate { get; set; }
```

Property Value

[DateTime](#)

### Description

```
public string? Description { get; set; }
```

Property Value

[string](#)

### Id

```
public long Id { get; set; }
```

Property Value

[long](#) ↗

## ImageUrl

```
public string? ImageUrl { get; set; }
```

Property Value

[string](#) ↗

## Title

```
public string Title { get; set; }
```

Property Value

[string](#) ↗

## UpdatedDate

```
public DateTime? UpdatedDate { get; set; }
```

Property Value

[DateTime](#) ↗?

# Class NewsDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class NewsDTO
```

## Inheritance

[object](#) ← NewsDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### CreatedDate

```
public DateTime? CreatedDate { get; set; }
```

Property Value

[DateTime](#)?

### Description

```
public string? Description { get; set; }
```

Property Value

[string](#)

### Id

```
public long Id { get; set; }
```

Property Value

[long](#) ↗

## ImageUrl

```
public string? ImageUrl { get; set; }
```

Property Value

[string](#) ↗

## Title

```
public string? Title { get; set; }
```

Property Value

[string](#) ↗

## UpdatedDate

```
public DateTime? UpdatedDate { get; set; }
```

Property Value

[DateTime](#) ↗?

# Class PaginationDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

Represents pagination details for a collection of items.

```
public class PaginationDTO
```

## Inheritance

[object](#) ← PaginationDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### ItemsPerPage

Gets or sets the number of items per page.

```
public int ItemsPerPage { get; set; }
```

Property Value

[int](#)

### OrderDesc

Gets or sets a value indicating whether the order is descending.

```
public bool OrderDesc { get; set; }
```

Property Value

[bool](#)

## PageNumber

Gets or sets the current page number.

```
public int PageNumber { get; set; }
```

Property Value

[int](#)

## TotalItems

Gets or sets the total number of items.

```
public int TotalItems { get; set; }
```

Property Value

[int](#)

## TotalPages

Gets or sets the total number of pages.

```
public int TotalPages { get; set; }
```

Property Value

[int](#)

# Class Register

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

Represents a registration model with user details.

```
public class Register
```

## Inheritance

[object](#) ← Register

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Properties

## Email

Gets or sets the email address.

```
[Required]
[EmailAddress(ErrorMessage = "Invalid {0}")]
[Display(Name = "Email")]
[RegularExpression("^[_]*([a-zA-Z0-9]+(\\". |_* )?)+@[a-zA-Z][a-zA-Z0-9-]+(\\". |-
*\\". )+[a-zA-Z]{2,6}$", ErrorMessage = "{0}: regex-validation error -> input a valid
email like 'user@domain.xxx'")]
public string Email { get; set; }
```

## Property Value

[string](#)

The email address must be valid and follow the specified regular expression pattern.

## FirstName

Gets or sets the first name.

```
[StringLength(50, MinimumLength = 2, ErrorMessage = "{0}: length must be between {2} and {1}")]
[RegularExpression("^[abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-]*$",
ErrorMessage = "{0}: Prohibited character used, only alphanumericals and characters '-' and '_' are allowed")]
[Display(Name = "Firstname")]
public string? FirstName { get; set; }
```

### Property Value

[string](#)

## LastName

Gets or sets the last name.

```
[StringLength(70, MinimumLength = 2, ErrorMessage = "{0}: length must be between {2} and {1}")]
[RegularExpression("^[abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-]*$",
ErrorMessage = "{0}: Prohibited character used, only alphanumericals and characters '-' and '_' are allowed")]
[Display(Name = "Lastname")]
public string? LastName { get; set; }
```

### Property Value

[string](#)

## Password

Gets or sets the password.

```
[Required(ErrorMessage = "{0} required")]
[MinLength(6, ErrorMessage = "{0} too short, minimum of {1} characters")]
```

```
[Display(Name = "Password")]
public string Password { get; set; }
```

## Property Value

[string](#)

The password must be at least 6 characters long.

## UserName

Gets or sets the username.

```
[Required(ErrorMessage = "{0}: required")]
[StringLength(25, MinimumLength = 2, ErrorMessage = "{0}: length must be between {2} and {1}")]
[RegularExpression(@"^[\w\-\_]*$", ErrorMessage = "{0}: Prohibited character used, only alphanumericals and characters '-' and '_' are allowed")]
[Display(Name = "Username")]
public string UserName { get; set; }
```

## Property Value

[string](#)

The username must be between 2 and 25 characters long and can only contain alphanumeric characters, hyphens, and underscores.

# Class Reservation

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class Reservation
```

## Inheritance

[object](#) ← Reservation

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Properties

## Created

```
public DateTime Created { get; set; }
```

Property Value

[DateTime](#)

## DeletionDate

```
public DateTime? DeletionDate { get; set; }
```

Property Value

[DateTime](#)?

## Id

```
public long Id { get; set; }
```

Property Value

[long](#) ↗

## IsDeleted

```
public bool IsDeleted { get; set; }
```

Property Value

[bool](#) ↗

## ReservationEnd

```
public DateOnly ReservationEnd { get; set; }
```

Property Value

[DateOnly](#) ↗

## ReservationStart

```
public DateOnly ReservationStart { get; set; }
```

Property Value

[DateOnly](#) ↗

## User

```
public User? User { get; set; }
```

Property Value

[User](#)

## Vehicle

```
public Vehicle? Vehicle { get; set; }
```

Property Value

[Vehicle](#)

# Class ReservationDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class ReservationDTO
```

## Inheritance

[object](#) ← ReservationDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Created

```
public DateTime Created { get; set; }
```

Property Value

[DateTime](#)

### Id

```
public long Id { get; set; }
```

Property Value

[long](#)

### ReservationEnd

```
public DateOnly ReservationEnd { get; set; }
```

Property Value

[DateOnly](#) ↗

## ReservationStart

```
public DateOnly ReservationStart { get; set; }
```

Property Value

[DateOnly](#) ↗

## Username

```
public string? Username { get; set; }
```

Property Value

[string](#) ↗

## VehicleId

```
public long VehicleId { get; set; }
```

Property Value

[long](#) ↗

# Class Settings

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

Represents the settings for various constraints in the application.

```
public class Settings
```

## Inheritance

[object](#) ← Settings

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Fields

### corsSites

Frontend CORS

```
public static string[] corsSites
```

### Field Value

[string](#)[]

### firstNameMaxLength

Maximum length for the first name.

```
public const int firstNameMaxLength = 50
```

### Field Value

[int](#)

## firstNameMinLength

Minimum length for the first name.

```
public const int firstNameMinLength = 2
```

Field Value

[int](#)

## lastNameMaxLength

Maximum length for the last name.

```
public const int lastNameMaxLength = 70
```

Field Value

[int](#)

## lastNameMinLength

Minimum length for the last name.

```
public const int lastNameMinLength = 2
```

Field Value

[int](#)

## maxFileSize

Maximum file size for uploaded files.

```
public const long maxFileSize = 5242880
```

Field Value

[long](#)

## motorizedTypes

Indicates what vehicles require lisenceplate

```
public static string[] motorizedTypes
```

Field Value

[string](#)[]

## numberOfNewsRetrieved

Indicates how many news is retrieved by GetNewsList. Value of 0 will retrieve all. Positive values will limit number of hits.

```
public const int numberOfNewsRetrieved = 3
```

Field Value

[int](#)

## pwdMaxLength

Maximum length for the password.

```
public const int pwdMaxLength = 100
```

Field Value

[int](#)

## pwdMinLength

Minimum length for the password.

```
public const int pwdMinLength = 6
```

### Field Value

[int](#)

## reservationSoftDelete

Indicates whether or not reservations are soft deleted.

```
public const bool reservationSoftDelete = true
```

### Field Value

[bool](#)

## useMariaDB

Indicates whether to use MariaDB as the database provider.

```
public const bool useMariaDB = true
```

### Field Value

[bool](#)

## userSoftDelete

Indicates whether or not user accounts are soft deleted.

```
public const bool userSoftDelete = true
```

Field Value

[bool](#)

## usernameMaxLength

Maximum length for the username.

```
public const int usernameMaxLength = 25
```

Field Value

[int](#)

## usernameMinLength

Minimum length for the username.

```
public const int usernameMinLength = 2
```

Field Value

[int](#)

## vehicleSoftDelete

Indicates whether or not vehicles are soft deleted.

```
public const bool vehicleSoftDelete = true
```

Field Value

[bool](#)

# Properties

## uploadServiceImgUrlBaseUrl

Base URL for the upload service image.

```
public static string uploadServiceImgUrlBaseUrl { get; set; }
```

Property Value

[string](#)

# Class UpdateUserDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

Data Transfer Object for updating user details.

```
public class UpdateUserDTO
```

## Inheritance

[object](#) ← UpdateUserDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Email

Gets or sets the email of the user.

```
[EmailAddress(ErrorMessage = "Invalid {0}")]
[RegularExpression("^[_]*([a-zA-Z0-9]+(\\".|\_*)?)+@[a-zA-Z][a-zA-Z0-9-]+(\\".|-\\".))+[a-zA-Z]{2,6}$", ErrorMessage = "{0}: regex-validation error -> input a valid
email like 'user@domain.xxx'")]
[Display(Name = "Email")]
public string? Email { get; set; }
```

### Property Value

[string](#)

### FirstName

Gets or sets the first name of the user.

```
[StringLength(50, MinimumLength = 2, ErrorMessage = "{0}: length must be between {2} and {1}")]
[RegularExpression("^[abcdefghijklmnopqrstuvwxyzÄÖÅ_]*$",
ErrorMessage = "{0}: Prohibited character used, only alphanumericals and characters '-' and '_' are allowed")]
[Display(Name = "Firstname")]
public string? FirstName { get; set; }
```

Property Value

[string](#)

## Lastname

Gets or sets the last name of the user.

```
[StringLength(70, MinimumLength = 2, ErrorMessage = "{0}: length must be between {2} and {1}")]
[RegularExpression("^[abcdefghijklmnopqrstuvwxyzÄÖÅ_]*$",
ErrorMessage = "{0}: Prohibited character used, only alphanumericals and characters '-' and '_' are allowed")]
[Display(Name = "Lastname")]
public string? LastName { get; set; }
```

Property Value

[string](#)

# Class UpdateUserPwdDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class UpdateUserPwdDTO
```

## Inheritance

[object](#) ← UpdateUserPwdDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### NewPwd

```
[Required(ErrorMessage = "{0} required")]
[MinLength(6, ErrorMessage = "{0} too short, minimum of {1} characters")]
[Display(Name = "Password")]
public string NewPwd { get; set; }
```

### Property Value

[string](#)

### NewPwd2

```
[Required(ErrorMessage = "{0} required")]
[MinLength(6, ErrorMessage = "{0} too short, minimum of {1} characters")]
[Display(Name = "Password")]
public string NewPwd2 { get; set; }
```

### Property Value

[string](#) ↗

## OldPwd

```
public string OldPwd { get; set; }
```

Property Value

[string](#) ↗

# Class UploadFile

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

Represents a file uploaded by a user.

```
public class UploadFile
```

## Inheritance

[object](#) ← UploadFile

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### DeletionDate

```
public DateTime? DeletionDate { get; set; }
```

Property Value

[DateTime](#)?

### Description

```
public string? Description { get; set; }
```

Property Value

[string](#)

## FilePath

Gets or sets the file path where the uploaded file is stored.

[Required]

```
public string FilePath { get; set; }
```

### Property Value

[string](#)

The file path where the uploaded file is stored.

## FileUrl

Gets or sets the URL of the uploaded file.

[Required]

[Url]

```
public string FileUrl { get; set; }
```

### Property Value

[string](#)

The URL of the uploaded file.

## Id

Gets or sets the unique identifier for the uploaded file.

```
public int Id { get; set; }
```

### Property Value

[int](#)

The unique identifier for the uploaded file.

## IsDeleted

```
public bool IsDeleted { get; set; }
```

Property Value

[bool](#)

## RelatedVehicle

Gets or sets the vehicle associated with the uploaded file.

[Required]

```
public Vehicle RelatedVehicle { get; set; }
```

Property Value

[Vehicle](#)

The vehicle associated with the uploaded file.

## UniqueFileName

Gets or sets the name of the uploaded file.

[Required]

```
public string UniqueFileName { get; set; }
```

Property Value

[string](#)

The name of the uploaded file.

## Uploaded

```
public DateTime Uploaded { get; set; }
```

Property Value

[DateTime](#) ↗

## Uploader

Gets or sets the user who uploaded the file.

[Required]

```
public User Uploader { get; set; }
```

Property Value

[User](#)

The user who uploaded the file.

# Class UploadFileDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class UploadFileDTO
```

## Inheritance

[object](#) ← UploadFileDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Description

```
public string? Description { get; set; }
```

#### Property Value

[string](#)

### FilePath

```
public string FilePath { get; set; }
```

#### Property Value

[string](#)

### FileUrl

```
public string FileUrl { get; set; }
```

Property Value

[string](#)

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

RelatedVehicleId

```
public long RelatedVehicleId { get; set; }
```

Property Value

[long](#)

UniqueFileName

```
public string UniqueFileName { get; set; }
```

Property Value

[string](#)

Uploaded

```
public DateTime Uploaded { get; set; }
```

Property Value

[DateTime](#) ↗

## UploaderId

```
public string UploaderId { get; set; }
```

Property Value

[string](#) ↗

# Class UploadFilePostDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class UploadFilePostDTO
```

## Inheritance

[object](#) ← UploadFilePostDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Description

```
public string? Description { get; set; }
```

#### Property Value

[string](#)

### File

```
public IFormFile File { get; set; }
```

#### Property Value

[IFormFile](#)

### RelatedVehicleId

```
public int RelatedVehicleId { get; set; }
```

Property Value

[int](#) ↗

# Class User

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

Represents a user in the system.

```
public class User : IdentityUser
```

## Inheritance

[object](#) ← [IdentityUser<string>](#) ← [IdentityUser](#) ← [User](#)

## Inherited Members

[IdentityUser<string>.ToString\(\)](#) , [IdentityUser<string>.Id](#) ,  
[IdentityUser<string>.UserName](#) , [IdentityUser<string>.NormalizedUserName](#) ,  
[IdentityUser<string>.Email](#) , [IdentityUser<string>.NormalizedEmail](#) ,  
[IdentityUser<string>.EmailConfirmed](#) , [IdentityUser<string>.PasswordHash](#) ,  
[IdentityUser<string>.SecurityStamp](#) , [IdentityUser<string>.ConcurrencyStamp](#) ,  
[IdentityUser<string>.PhoneNumber](#) , [IdentityUser<string>.PhoneNumberConfirmed](#) ,  
[IdentityUser<string>.TwoFactorEnabled](#) , [IdentityUser<string>.LockoutEnd](#) ,  
[IdentityUser<string>.LockoutEnabled](#) , [IdentityUser<string>.AccessFailedCount](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

## Properties

### DeletionDate

Gets or sets the date when the user was marked deleted.

```
public DateTime? DeletionDate { get; set; }
```

### Property Value

[DateTime](#)?

The date when the user was deleted, system sets this.

## FirstName

Gets or sets the first name of the user.

```
public string? FirstName { get; set; }
```

### Property Value

[string](#)

The first name of the user.

## IsDeleted

Gets or sets a value indicating whether the user is deleted.

```
public bool IsDeleted { get; set; }
```

### Property Value

[bool](#)

`true` if the user is deleted; otherwise, `false`, system sets this.

## JoinDate

Gets or sets the date when the user joined.

```
public DateTime JoinDate { get; set; }
```

### Property Value

[DateTime](#)

The date when the user joined, system sets this.

## LastLogin

Gets or sets the date when the user last logged in.

```
public DateTime? LastLogin { get; set; }
```

## Property Value

[DateTime](#)?

The date when the user last logged in, system sets this.

## LastName

Gets or sets the last name of the user.

```
public string? LastName { get; set; }
```

## Property Value

[string](#)?

The last name of the user.

## Updated

Gets or sets the date when the user was last updated.

```
public DateTime? Updated { get; set; }
```

## Property Value

[DateTime](#)?

The date when the user was last updated, system sets this.

# Class UserDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

Data Transfer Object for User entity.

```
public class UserDTO
```

## Inheritance

[object](#) ← UserDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Id

Gets or sets the unique identifier for the user.

```
public string Id { get; set; }
```

### Property Value

[string](#)

### JoinDate

Gets or sets the date when the user joined.

```
public DateTime JoinDate { get; set; }
```

### Property Value

## Username

Gets or sets the username of the user.

```
public string Username { get; set; }
```

Property Value

[string](#)

# Class UserRole

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class UserRole
```

## Inheritance

[object](#) ← UserRole

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Role

```
public string Role { get; set; }
```

Property Value

[string](#)

### Username

```
public string Username { get; set; }
```

Property Value

[string](#)

# Class Vehicle

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class Vehicle
```

## Inheritance

[object](#) ← Vehicle

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Deleted

```
public bool Deleted { get; set; }
```

Property Value

[bool](#)

### DeletionDate

```
public DateTime? DeletionDate { get; set; }
```

Property Value

[DateTime](#)?

## Description

```
public string? Description { get; set; }
```

Property Value

[string](#)

Id

```
public long Id { get; set; }
```

Property Value

[long](#)

ImageUrl

[Url]

```
public string? ImageUrl { get; set; }
```

Property Value

[string](#)

LicensePlate

```
public string? LicensePlate { get; set; }
```

Property Value

[string](#)

Make

```
[Required]  
public string Make { get; set; }
```

Property Value

[string](#) ↗

## Model

```
public string? Model { get; set; }
```

Property Value

[string](#) ↗

## ModelYear

```
public int? ModelYear { get; set; }
```

Property Value

[int](#) ↗?

## Nickname

```
[MinLength(3)]  
public string? Nickname { get; set; }
```

Property Value

[string](#) ↗

## NotUsable

```
public bool? NotUsable { get; set; }
```

Property Value

bool?

## VehicleType

[Required]

```
public string VehicleType { get; set; }
```

Property Value

string?

# Class VehicleDTO

Namespace: [Backend\\_Reservation\\_Service.Data.Models](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class VehicleDTO
```

## Inheritance

[object](#) ← VehicleDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Description

```
public string? Description { get; set; }
```

#### Property Value

[string](#)

### Id

```
public long Id { get; set; }
```

#### Property Value

[long](#)

### ImageUrl

```
public string? ImageUrl { get; set; }
```

Property Value

[string](#) ↗

## LicensePlate

```
public string? LicensePlate { get; set; }
```

Property Value

[string](#) ↗

## Make

```
public string? Make { get; set; }
```

Property Value

[string](#) ↗

## Model

```
public string? Model { get; set; }
```

Property Value

[string](#) ↗

## ModelYear

```
public int? ModelYear { get; set; }
```

Property Value

[int](#)?

## Nickname

```
public string? Nickname { get; set; }
```

Property Value

[string](#)?

## NotUsable

```
public bool? NotUsable { get; set; }
```

Property Value

[bool](#)?

## VehicleType

```
public string? VehicleType { get; set; }
```

Property Value

[string](#)?

# Namespace Backend\_Reservation\_Service. Data.Repositories

## Classes

[NewsRepository](#)

[ReservationRepository](#)

[RoleRepository](#)

[UploadFileRepository](#)

[UserRepository](#)

[VehicleRepository](#)

## Interfaces

[INewsRepository](#)

[IReservationRepository](#)

[IRoleRepository](#)

Interface for role repository to manage roles.

[IUploadFileRepository](#)

[IUserRepository](#)

[IVehicleRepository](#)

# Interface INewsRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public interface INewsRepository
```

## Methods

### DeleteNewsAsync(News)

Task<bool> **DeleteNewsAsync**(News news)

#### Parameters

news [News](#)

#### Returns

[Task](#)<bool>

### GetNewsListAsync(bool)

Task<IEnumerable<News>> **GetNewsListAsync**(bool all)

#### Parameters

all [bool](#)

#### Returns

[Task](#)<IEnumerable<News>>

## GetSingleNewsAsync(long)

Task<News?> GetSingleNewsAsync([long](#) id)

Parameters

[id](#) [long](#)

Returns

[Task](#)<[News](#)>

## PostNewNewsAsync(News)

Task<News?> PostNewNewsAsync(News newNews)

Parameters

[newNews](#) [News](#)

Returns

[Task](#)<[News](#)>

## UpdateNewsAsync(News)

Task<News?> UpdateNewsAsync(News news)

Parameters

[news](#) [News](#)

Returns

[Task](#)<[News](#)>

# Interface IReservationRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public interface IReservationRepository
```

## Methods

### CreateReservationAsync(Reservation)

```
Task<(Reservation reservation, string statusMessage)>
CreateReservationAsync(Reservation reservation)
```

#### Parameters

reservation [Reservation](#)

#### Returns

[Task](#)<(Reservation reservation, string statusMessage)>

### DeleteReservationAsync(Reservation)

```
Task<(bool success, string statusMessage)> DeleteReservationAsync(Reservation
reservation)
```

#### Parameters

reservation [Reservation](#)

#### Returns

[Task](#)<(bool success, string statusMessage)>

## GetActiveReservationsByUserAsync(string, int, int, bool)

```
Task<(IEnumerable<Reservation>, PaginationDTO pagination)>
GetActiveReservationsByUserAsync(string userName, int itemsPerPage, int pageNumber,
bool orderDesc)
```

### Parameters

userName [string](#)

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

### Returns

[Task](#)<(IEnumerable<Reservation>, PaginationDTO pagination)>

## GetAllReservationsAsync(int, int, bool, int)

```
Task<(IEnumerable<Reservation>, PaginationDTO pagination)>
GetAllReservationsAsync(int itemsPerPage, int pageNumber, bool orderDesc,
int deletedIncludeMode)
```

### Parameters

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

deletedIncludeMode [int](#)

### Returns

[Task](#)<(IEnumerable<Reservation>, PaginationDTO pagination)>

## GetPastReservationsByUserAsync(string, int, int, bool)

```
Task<(IEnumerable<Reservation>, PaginationDTO pagination)>
GetPastReservationsByUserAsync(string userName, int itemsPerPage, int pageNumber,
bool orderDesc)
```

### Parameters

userName [string](#)

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

### Returns

[Task](#)<(IEnumerable<Reservation>, PaginationDTO pagination)>

## GetReservationByIdAsync(long)

```
Task<Reservation> GetReservationByIdAsync(long reservationId)
```

### Parameters

reservationId [long](#)

### Returns

[Task](#)<Reservation>

## GetReservationsByUserAsync(string, int, int, bool)

```
Task<(IEnumerable<Reservation>, PaginationDTO pagination)>
GetReservationsByUserAsync(string userName, int itemsPerPage, int pageNumber,
bool orderDesc)
```

## Parameters

userName [string](#)

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

## Returns

[Task](#)<([IEnumerable](#)<[Reservation](#)>, [PaginationDTO](#) pagination)>

## GetReservationsByVehicleAsync(long)

[Task](#)<[IEnumerable](#)<[Reservation](#)>> GetReservationsByVehicleAsync([long](#) vehicleId)

## Parameters

vehicleId [long](#)

## Returns

[Task](#)<[IEnumerable](#)<[Reservation](#)>>

## GetReservationsByVehicleAsync(long, int, int, bool)

[Task](#)<([IEnumerable](#)<[Reservation](#)>, [PaginationDTO](#) pagination)>

GetReservationsByVehicleAsync([long](#) vehicleId, [int](#) itemsPerPage, [int](#) pageNumber, [bool](#) orderDesc)

## Parameters

vehicleId [long](#)

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

Returns

[Task](#)<([IEnumerable](#)<[Reservation](#)>, [PaginationDTO](#) [pagination](#))>

## IsVehicleFree(long, DateOnly?, DateOnly?)

Task<bool> [IsVehicleFree](#)([long](#) vehicleId, [DateOnly?](#) startDtm, [DateOnly?](#) endDtm)

Parameters

vehicleId [long](#)

startDtm [DateOnly](#)?

endDtm [DateOnly](#)?

Returns

[Task](#)<[bool](#)>

## ReservationExistsAsync(long)

Task<bool> [ReservationExistsAsync](#)([long](#) reservationId)

Parameters

reservationId [long](#)

Returns

[Task](#)<[bool](#)>

# Interface IRoleRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

Interface for role repository to manage roles.

```
public interface IRoleRepository
```

## Methods

### CreateRoleAsync(string)

Creates a new role asynchronously.

```
Task<bool> CreateRoleAsync(string role)
```

#### Parameters

**role** [string](#)

The name of the role to create.

#### Returns

[Task](#)<[bool](#)>

A task that represents the asynchronous operation. The task result contains a boolean indicating whether the role was created successfully.

### DeleteRoleAsync(string)

Deletes an existing role asynchronously.

```
Task<bool> DeleteRoleAsync(string role)
```

## Parameters

`role` [string](#)

The name of the role to delete.

## Returns

[Task](#)<[bool](#)>

A task that represents the asynchronous operation. The task result contains a boolean indicating whether the role was deleted successfully.

# Interface IUploadFileRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public interface IUploadFileRepository
```

## Methods

### FindFileMetaDataByVehcileDbAsync(long, string)

```
Task<(bool Success, string Message, UploadFile UploadFile)>
FindFileMetaDataByVehcileDbAsync(long relatedVehicleId, string uniquefilename)
```

#### Parameters

relatedVehicleId [long](#)

uniquefilename [string](#)

#### Returns

[Task](#)<(bool Success, string Message, UploadFile UploadFile)>

### SaveFileMetaDataToDbAsync(UploadFile)

```
Task<(bool Success, string Message)> SaveFileMetaDataToDbAsync(UploadFile
dbUploadFile)
```

#### Parameters

dbUploadFile [UploadFile](#)

#### Returns

Task<(bool success, string statusMessage)>

# Interface IUserRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public interface IUserRepository
```

## Methods

### AssignRoleAsync(User, string)

```
Task<IdentityResult> AssignRoleAsync(User user, string role)
```

#### Parameters

user [User](#)

role [string](#)

#### Returns

[Task](#)<[IdentityResult](#)>

### ChangePwdAsync(User, string, string)

```
Task<IdentityResult> ChangePwdAsync(User user, string oldPassword,  
string newPassword)
```

#### Parameters

user [User](#)

oldPassword [string](#)

newPassword [string](#)

Returns

[Task](#)<[IdentityResult](#)>

## CheckPasswordAsync(User, string)

Task<bool> [CheckPasswordAsync](#)(User user, string password)

Parameters

user [User](#)

password [string](#)

Returns

[Task](#)<bool>

## CreateUserAsync(User, string)

Task<IdentityResult> [CreateUserAsync](#)(User user, string password)

Parameters

user [User](#)

password [string](#)

Returns

[Task](#)<[IdentityResult](#)>

## DeleteUserAsync(User)

Task<(bool Success, string Message)> [DeleteUserAsync](#)(User user)

Parameters

user [User](#)

Returns

[Task](#)<([bool](#) [success](#), [string](#) [statusMessage](#))>

## FindByEmailAsyncWithDeleted(string?)

`Task<User?> FindByEmailAsyncWithDeleted(string? email)`

Parameters

email [string](#)

Returns

[Task](#)<[User](#)>

## FindByIdAsync(string?)

`Task<User?> FindByIdAsync(string? userId)`

Parameters

userId [string](#)

Returns

[Task](#)<[User](#)>

## FindByUsernameAsync(string?)

`Task<User?> FindByUsernameAsync(string? username)`

Parameters

username [string](#)

Returns

[Task](#)<[User](#)>

## FindByUsernameWithDeletedAsync(string?)

`Task<User?> FindByUsernameWithDeletedAsync(string? username)`

Parameters

username [string](#)

Returns

[Task](#)<[User](#)>

## GetAllUsersAsync(int, int, bool, bool)

`Task<(IEnumerable<User>, PaginationDTO pagination)> GetAllUsersAsync(int itemsPerPage, int pageNumber, bool orderDesc, bool onlyDeleted)`

Parameters

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

onlyDeleted [bool](#)

Returns

[Task](#)<(IEnumerable<User>, PaginationDTO pagination)>

## GetRolesAsync(User)

Task<IList<[string](#)>> GetRolesAsync(User user)

Parameters

user [User](#)

Returns

[Task](#)<[IList](#)<[string](#)>>

## RestoreSoftDeletedUserAsync(User)

Task<(bool succeeded, string message)> RestoreSoftDeletedUserAsync(User user)

Parameters

user [User](#)

Returns

[Task](#)<(bool success, string statusMessage)>

## UpdateUserAsync(User)

Task<bool> UpdateUserAsync(User user)

Parameters

user [User](#)

Returns

[Task](#)<bool>

# Interface IVehicleRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public interface IVehicleRepository
```

## Methods

### DeleteVehicleAsync(Vehicle)

```
Task<bool> DeleteVehicleAsync(Vehicle vehicle)
```

#### Parameters

**vehicle** [Vehicle](#)

#### Returns

[Task](#)<[bool](#)>

### GetDdMakeListAsync(string?, string?)

```
Task<IEnumerable<Vehicle>> GetDdMakeListAsync(string? type, string? model)
```

#### Parameters

**type** [string](#)

**model** [string](#)

#### Returns

[Task](#)<[IEnumerable](#)<[Vehicle](#)>>

## GetDdModelListAsync(string?, string?)

Task<IEnumerable<Vehicle>> GetDdModelListAsync(string? type, string? make)

### Parameters

type [string](#)

make [string](#)

### Returns

[Task](#)<[IEnumerable](#)<Vehicle>>

## GetDdTTypeListAsync(string?, string?)

Task<IEnumerable<Vehicle>> GetDdTTypeListAsync(string? make, string? model)

### Parameters

make [string](#)

model [string](#)

### Returns

[Task](#)<[IEnumerable](#)<Vehicle>>

## GetDelVehicleListAsync(int, int, bool?)

Task<(IEnumerable<Vehicle>, PaginationDTO)> GetDelVehicleListAsync(int itemsPerPage, int page, bool? oldestFirst)

### Parameters

itemsPerPage [int](#)

page [int](#)

oldestFirst [bool](#)?

Returns

[Task](#)<([IEnumerable](#)<[Vehicle](#)>, [PaginationDTO](#))>

## GetVehicleByIdAsync(long?)

[Task](#)<[Vehicle](#)?> GetVehicleByIdAsync([long](#)? vehicleId)

Parameters

vehicleId [long](#)?

Returns

[Task](#)<[Vehicle](#)>

## GetVehicleByLicenseplateAsync(string)

[Task](#)<[Vehicle](#)?> GetVehicleByLicenseplateAsync([string](#) licenseplate)

Parameters

licenseplate [string](#)

Returns

[Task](#)<[Vehicle](#)>

## GetVehicleByNicknameAsync(string)

[Task](#)<[Vehicle](#)?> GetVehicleByNicknameAsync([string](#) name)

## Parameters

name [string](#)?

## Returns

[Task](#)<[Vehicle](#)>

**GetVehiclesAsAdminAsync(bool?, bool?, bool?, string?, string?, string?, DateOnly?, DateOnly?, bool?, int, int)**

```
Task<(IEnumerable<Vehicle>, PaginationDTO)> GetVehiclesAsAdminAsync(bool?  
includeDeleted, bool? onlyEmptyDescriptions, bool? onlyEmptyNicknames, string? type,  
string? make, string? model, string? year, DateOnly? dtmStart, DateOnly? dtmEnd,  
bool? includeNotUsable, int itemsPerPage, int page)
```

## Parameters

includeDeleted [bool](#)?

onlyEmptyDescriptions [bool](#)?

onlyEmptyNicknames [bool](#)?

type [string](#)?

make [string](#)?

model [string](#)?

year [string](#)?

dtmStart [DateOnly](#)?

dtmEnd [DateOnly](#)?

includeNotUsable [bool](#)?

itemsPerPage [int](#)?

page [int](#)?

Returns

[Task](#)<(IEnumarable<Vehicle>, PaginationDTO)>

## GetVehiclesAsync(string?, string?, string?, string?, DateOnly?, DateOnly?, bool?, int, int)

```
Task<(IEnumarable<Vehicle>, PaginationDTO)> GetVehiclesAsync(string? type, string?  
make, string? model, string? year, DateOnly? dtmStart, DateOnly? dtmEnd, bool?  
includeNotUsable, int itemsPerPage, int page)
```

Parameters

type [string](#)

make [string](#)

model [string](#)

year [string](#)

dtmStart [DateOnly](#)?

dtmEnd [DateOnly](#)?

includeNotUsable [bool](#)?

itemsPerPage [int](#)

page [int](#)

Returns

[Task](#)<(IEnumarable<Vehicle>, PaginationDTO)>

## NewVehicleAsync(Vehicle)

```
Task<Vehicle?> NewVehicleAsync(Vehicle vehicle)
```

Parameters

[vehicle](#) [Vehicle](#)

Returns

[Task](#) <[Vehicle](#)>

## RestoreSoftDelVehicleAsync(Vehicle)

`Task<bool> RestoreSoftDelVehicleAsync(Vehicle vehicle)`

Parameters

[vehicle](#) [Vehicle](#)

Returns

[Task](#) <[bool](#)>

## UpdateVehicleAsync(Vehicle)

`Task<bool> UpdateVehicleAsync(Vehicle vehicle)`

Parameters

[vehicle](#) [Vehicle](#)

Returns

[Task](#) <[bool](#)>

# Class NewsRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class NewsRepository : INewsRepository
```

## Inheritance

[object](#) ← NewsRepository

## Implements

[INewsRepository](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

NewsRepository(ReservationServiceDbContext, IReservationRepository)

```
public NewsRepository(ReservationServiceDbContext context,  
IReservationRepository reservationRepository)
```

## Parameters

context [ReservationServiceDbContext](#)

reservationRepository [IReservationRepository](#)

## Methods

DeleteNewsAsync(News)

```
public Task<bool> DeleteNewsAsync(News news)
```

Parameters

news [News](#)

Returns

[Task](#)<bool>

## GetNewsListAsync(bool)

```
public Task<IEnumerable<News>> GetNewsListAsync(bool all)
```

Parameters

all [bool](#)

Returns

[Task](#)<[IEnumerable](#)<News>>

## GetSingleNewsAsync(long)

```
public Task<News?> GetSingleNewsAsync(long id)
```

Parameters

id [long](#)

Returns

[Task](#)<News>

## PostNewNewsAsync(News)

```
public Task<News?> PostNewNewsAsync(News newNews)
```

Parameters

newNews [News](#)

Returns

[Task](#) <[News](#)>

## UpdateNewsAsync(News)

```
public Task<News?> UpdateNewsAsync(News news)
```

Parameters

news [News](#)

Returns

[Task](#) <[News](#)>

# Class ReservationRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class ReservationRepository : IReservationRepository
```

## Inheritance

[object](#) ← ReservationRepository

## Implements

[IReservationRepository](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### ReservationRepository(ReservationServiceDbContext)

```
public ReservationRepository(ReservationServiceDbContext context)
```

## Parameters

context [ReservationServiceDbContext](#)

## Methods

### CreateReservationAsync(Reservation)

```
public Task<(Reservation reservation, string statusMessage)>  
CreateReservationAsync(Reservation reservation)
```

Parameters

reservation [Reservation](#)

Returns

[Task](#)<([Reservation](#) reservation, [string](#) statusMessage)>

## DeleteReservationAsync(Reservation)

```
public Task<(bool success, string statusMessage)> DeleteReservationAsync(Reservation reservation)
```

Parameters

reservation [Reservation](#)

Returns

[Task](#)<([bool](#) success, [string](#) statusMessage)>

## GetActiveReservationsByUserAsync(string, int, int, bool)

```
public Task<(IEnumerable<Reservation>, PaginationDTO pagination)>
GetActiveReservationsByUserAsync(string userName, int itemsPerPage, int pageNumber,
bool orderDesc)
```

Parameters

userName [string](#)

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

Returns

[Task](#)<(IEnumarable<Reservation>, PaginationDTO pagination)>

## GetAllReservationsAsync(int, int, bool, int)

```
public Task<(IEnumarable<Reservation>, PaginationDTO pagination)>
GetAllReservationsAsync(int itemsPerPage, int pageNumber, bool orderDesc,
int deletedIncludeMode)
```

### Parameters

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

deletedIncludeMode [int](#)

### Returns

[Task](#)<(IEnumarable<Reservation>, PaginationDTO pagination)>

## GetPastReservationsByUserAsync(string, int, int, bool)

```
public Task<(IEnumarable<Reservation>, PaginationDTO pagination)>
GetPastReservationsByUserAsync(string userName, int itemsPerPage, int pageNumber,
bool orderDesc)
```

### Parameters

userName [string](#)

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

### Returns

[Task](#)<(IEnumarable<Reservation>, PaginationDTO pagination)>

## GetReservationByIdAsync(long)

public Task<Reservation> GetReservationByIdAsync(long reservationId)

Parameters

reservationId [long](#)

Returns

[Task](#)<Reservation>

## GetReservationsByUserAsync(string, int, int, bool)

public Task<(IEnumarable<Reservation>, PaginationDTO pagination)> GetReservationsByUserAsync(string userName, int itemsPerPage, int pageNumber, bool orderDesc)

Parameters

userName [string](#)

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

Returns

[Task](#)<(IEnumarable<Reservation>, PaginationDTO pagination)>

## GetReservationsByVehicleAsync(long)

Retrieves a list of reservations for a specific vehicle. Created for a service layer function that checks overlapping reservations.

```
public Task<IEnumerable<Reservation>> GetReservationsByVehicleAsync(long vehicleId)
```

## Parameters

**vehicleId** [long](#)

The ID of the vehicle.

## Returns

[Task](#)<[IEnumerable](#)<[Reservation](#)>>

A list of reservations for the specified vehicle.

## Exceptions

[Exception](#)

Thrown when there is an error retrieving the reservations.

# GetReservationsByVehicleAsync([long](#), [int](#), [int](#), [bool](#))

Retrieves a paginated list of reservations for a specific vehicle. For controller use.

```
public Task<(IEnumerable<Reservation>, PaginationDTO pagination)>
GetReservationsByVehicleAsync(long vehicleId, int itemsPerPage, int pageNumber,
bool orderDesc)
```

## Parameters

**vehicleId** [long](#)

The ID of the vehicle.

**itemsPerPage** [int](#)

The number of reservations to return per page.

`pageNumber` [int](#)

The page number to retrieve.

`orderDesc` [bool](#)

If set to true, the results will be ordered in descending order; otherwise, they will be ordered in ascending order.

Returns

[Task](#)<(I[Enumerable](#)<[Reservation](#)>, [PaginationDTO](#) [pagination](#))>

A list of reservations for the specified vehicle.

Exceptions

[Exception](#)

Thrown when there is an error retrieving the reservations.

## IsVehicleFree(long, DateOnly?, DateOnly?)

```
public Task<bool> IsVehicleFree(long vehicleId, DateOnly? startDtm,  
DateOnly? endDtm)
```

Parameters

`vehicleId` [long](#)

`startDtm` [DateOnly](#)?

`endDtm` [DateOnly](#)?

Returns

[Task](#)<[bool](#)>

## ReservationExistsAsync(long)

```
public Task<bool> ReservationExistsAsync(long reservationId)
```

Parameters

reservationId [long](#)

Returns

[Task](#) <bool>

# Class RoleRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class RoleRepository : IRoleRepository
```

## Inheritance

[object](#) ← RoleRepository

## Implements

[IRoleRepository](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### RoleRepository(RoleManager<IdentityRole>)

```
public RoleRepository(RoleManager<IdentityRole> roleManager)
```

## Parameters

roleManager [RoleManager](#)<[IdentityRole](#)>

## Methods

### CreateRoleAsync(string)

Creates a new role asynchronously.

```
public Task<bool> CreateRoleAsync(string role)
```

## Parameters

`role string`

The name of the role to create.

## Returns

`Task<bool>`

A task that represents the asynchronous operation. The task result contains a boolean indicating whether the role was created successfully.

## DeleteRoleAsync(string)

Deletes an existing role asynchronously.

```
public Task<bool> DeleteRoleAsync(string role)
```

## Parameters

`role string`

The name of the role to delete.

## Returns

`Task<bool>`

A task that represents the asynchronous operation. The task result contains a boolean indicating whether the role was deleted successfully.

# Class UploadFileRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class UploadFileRepository : IUploadFileRepository
```

## Inheritance

[object](#) ← UploadFileRepository

## Implements

[IUploadFileRepository](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### UploadFileRepository(ReservationServiceDbContext)

```
public UploadFileRepository(ReservationServiceDbContext context)
```

## Parameters

context [ReservationServiceDbContext](#)

## Methods

### FindFileMetaDataByVehcileDbAsync(long, string)

```
public Task<(bool Success, string Message, UploadFile UploadFile)>
FindFileMetaDataByVehcileDbAsync(long relatedVehicleId, string uniquefilename)
```

## Parameters

relatedVehicleId [long](#)

uniqueFilename [string](#)

## Returns

[Task](#)<([bool](#) [Success](#), [string](#) [Message](#), [UploadFile](#) [UploadFile](#))>

## SaveFileMetaDataToDbAsync(UploadFile)

```
public Task<(bool Success, string Message)> SaveFileMetaDataToDbAsync(UploadFile dbUploadFile)
```

## Parameters

dbUploadFile [UploadFile](#)

## Returns

[Task](#)<([bool](#) [success](#), [string](#) [statusMessage](#))>

# Class UserRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class UserRepository : IUserRepository
```

## Inheritance

[object](#) ← UserRepository

## Implements

[IUserRepository](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

UserRepository(UserManager<User>,  
RoleManager<IdentityRole>,  
ReservationServiceDbContext)

```
public UserRepository(UserManager<User> userManager, RoleManager<IdentityRole>  
roleManager, ReservationServiceDbContext context)
```

## Parameters

userManager [UserManager](#)<User>

roleManager [RoleManager](#)<IdentityRole>

context [ReservationServiceDbContext](#)

## Methods

## AssignRoleAsync(User, string)

```
public Task<IdentityResult> AssignRoleAsync(User user, string role)
```

Parameters

user [User](#)

role [string](#)

Returns

[Task](#)<[IdentityResult](#)>

## ChangePwdAsync(User, string, string)

```
public Task<IdentityResult> ChangePwdAsync(User user, string oldPwd, string newPwd)
```

Parameters

user [User](#)

oldPwd [string](#)

newPwd [string](#)

Returns

[Task](#)<[IdentityResult](#)>

## CheckPasswordAsync(User, string)

```
public Task<bool> CheckPasswordAsync(User user, string password)
```

Parameters

user [User](#)

password [string](#)

Returns

[Task](#)<[bool](#)>

## CreateUserAsync(User, string)

```
public Task<IdentityResult> CreateUserAsync(User user, string password)
```

Parameters

user [User](#)

password [string](#)

Returns

[Task](#)<[IdentityResult](#)>

## DeleteUserAsync(User)

```
public Task<(bool Success, string Message)> DeleteUserAsync(User user)
```

Parameters

user [User](#)

Returns

[Task](#)<([bool](#) [success](#), [string](#) [statusMessage](#))>

## FindByEmailAsyncWithDeleted(string?)

```
public Task<User?> FindByEmailAsyncWithDeleted(string? email)
```

Parameters

`email` [string](#)

Returns

[Task](#)<[User](#)>

## FindByIdAsync(string?)

```
public Task<User?> FindByIdAsync(string? userId)
```

Parameters

`userId` [string](#)

Returns

[Task](#)<[User](#)>

## FindByUsernameAsync(string?)

```
public Task<User?> FindByUsernameAsync(string? username)
```

Parameters

`username` [string](#)

Returns

[Task](#)<[User](#)>

## FindByUsernameWithDeletedAsync(string?)

```
public Task<User?> FindByUsernameWithDeletedAsync(string? username)
```

Parameters

username [string](#)

Returns

[Task](#)<[User](#)>

## GetAllUsersAsync(int, int, bool, bool)

```
public Task<(IEnumerable<User>, PaginationDTO pagination)> GetAllUsersAsync(int itemsPerPage, int pageNumber, bool orderDesc, bool onlyDeleted)
```

Parameters

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

onlyDeleted [bool](#)

Returns

[Task](#)<(IEnumerable<User>, PaginationDTO pagination)>

## GetRolesAsync(User)

```
public Task<IList<string>> GetRolesAsync(User user)
```

Parameters

user [User](#)

Returns

[Task](#)<IList<string>>

## RestoreSoftDeletedUserAsync(User)

```
public Task<(bool succeeded, string message)> RestoreSoftDeletedUserAsync(User user)
```

Parameters

user [User](#)

Returns

[Task](#)<(bool success, string statusMessage)>

## UpdateUserAsync(User)

```
public Task<bool> UpdateUserAsync(User user)
```

Parameters

user [User](#)

Returns

[Task](#)<bool>

# Class VehicleRepository

Namespace: [Backend\\_Reservation\\_Service.Data.Repositories](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class VehicleRepository : IVehicleRepository
```

## Inheritance

[object](#) ← VehicleRepository

## Implements

[IVehicleRepository](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

VehicleRepository(ReservationServiceDbContext,  
IReservationRepository)

```
public VehicleRepository(ReservationServiceDbContext context,  
IReservationRepository reservationRepository)
```

## Parameters

context [ReservationServiceDbContext](#)

reservationRepository [IReservationRepository](#)

## Methods

DeleteVehicleAsync(Vehicle)

```
public Task<bool> DeleteVehicleAsync(Vehicle vehicle)
```

Parameters

vehicle [Vehicle](#)

Returns

[Task](#)<bool>

## GetDdMakeListAsync(string?, string?)

```
public Task<IEnumerable<Vehicle>> GetDdMakeListAsync(string? type, string? model)
```

Parameters

type [string](#)

model [string](#)

Returns

[Task](#)<IEnumerable<Vehicle>>

## GetDdModelListAsync(string?, string?)

```
public Task<IEnumerable<Vehicle>> GetDdModelListAsync(string? type, string? make)
```

Parameters

type [string](#)

make [string](#)

Returns

[Task](#)<IEnumerable<Vehicle>>

## GetDdTypeListAsync(string?, string?)

```
public Task<IEnumerable<Vehicle>> GetDdTypeListAsync(string? make, string? model)
```

### Parameters

make [string](#)

model [string](#)

### Returns

[Task](#)<[IEnumerable](#)<Vehicle>>

## GetDelVehicleListAsync(int, int, bool?)

```
public Task<(IEnumerable<Vehicle>, PaginationDTO)> GetDelVehicleListAsync(int itemsPerPage, int page, bool? oldestFirst)
```

### Parameters

itemsPerPage [int](#)

page [int](#)

oldestFirst [bool](#)?

### Returns

[Task](#)<([IEnumerable](#)<Vehicle>, [PaginationDTO](#))>

## GetVehicleByIdAsync(long?)

Retrieves a vehicle by its unique identifier.

```
public Task<Vehicle?> GetVehicleByIdAsync(long? vehicleId)
```

## Parameters

**vehicleId** [long](#)?

The unique identifier of the vehicle.

## Returns

[Task](#)<[Vehicle](#)>

A task that represents the asynchronous operation. The task result contains the vehicle if found; otherwise, null.

## GetVehicleByLicenseplateAsync(string)

```
public Task<Vehicle?> GetVehicleByLicenseplateAsync(string licenseplate)
```

## Parameters

**licenseplate** [string](#)

## Returns

[Task](#)<[Vehicle](#)>

## GetVehicleByNicknameAsync(string)

```
public Task<Vehicle?> GetVehicleByNicknameAsync(string name)
```

## Parameters

**name** [string](#)

## Returns

[Task](#)<[Vehicle](#)>

GetVehiclesAsAdminAsync(bool?, bool?, bool?, string?,  
string?, string?, string?, DateOnly?, DateOnly?, bool?,  
int, int)

```
public Task<(IEnumerable<Vehicle>, PaginationDTO)> GetVehiclesAsAdminAsync(bool?  
includeDeleted, bool? onlyEmptyDescriptions, bool? onlyEmptyNicknames, string? type,  
string? make, string? model, string? year, DateOnly? dtmStart, DateOnly? dtmEnd,  
bool? includeNotUsable, int itemsPerPage, int page)
```

Parameters

includeDeleted [bool](#)?

onlyEmptyDescriptions [bool](#)?

onlyEmptyNicknames [bool](#)?

type [string](#)

make [string](#)

model [string](#)

year [string](#)

dtmStart [DateOnly](#)?

dtmEnd [DateOnly](#)?

includeNotUsable [bool](#)?

itemsPerPage [int](#)

page [int](#)

Returns

[Task](#)<(IEnumerable<[Vehicle](#)>, [PaginationDTO](#))>

GetVehiclesAsync(string?, string?, string?, string?,  
DateOnly?, DateOnly?, bool?, int, int)

```
public Task<(IEnumerable<Vehicle>, PaginationDTO)> GetVehiclesAsync(string? type, string? make, string? model, string? year, DateOnly? dtmStart, DateOnly? dtmEnd, bool? includeNotUsable, int itemsPerPage, int page)
```

## Parameters

type [string](#)

make [string](#)

model [string](#)

year [string](#)

dtmStart [DateOnly](#)?

dtmEnd [DateOnly](#)?

includeNotUsable [bool](#)?

itemsPerPage [int](#)

page [int](#)

## Returns

[Task](#)<(IEnumerable<Vehicle>, PaginationDTO)>

## NewVehicleAsync(Vehicle)

```
public Task<Vehicle?> NewVehicleAsync(Vehicle vehicle)
```

## Parameters

vehicle [Vehicle](#)

## Returns

[Task](#)<[Vehicle](#)>

## RestoreSoftDelVehicleAsync(Vehicle)

```
public Task<bool> RestoreSoftDelVehicleAsync(Vehicle vehicle)
```

Parameters

**vehicle** [Vehicle](#)

Returns

[Task](#)<[bool](#)>

## UpdateVehicleAsync(Vehicle)

```
public Task<bool> UpdateVehicleAsync(Vehicle vehicle)
```

Parameters

**vehicle** [Vehicle](#)

Returns

[Task](#)<[bool](#)>

# Namespace Backend\_Reservation\_Service. Helpers

## Classes

[HelperMethods](#)

# Class HelperMethods

Namespace: [Backend\\_Reservation\\_Service.Helpers](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class HelperMethods
```

## Inheritance

[object](#) ← HelperMethods

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### ToSkipCalculator(int, int, int)

Luokan sisäinen Laskurifunktio, palauttaa tietokantahaulle tarvittavan 'skipattavien' määrään, jota tarvitaan kun määritellään millä 'sivulla' listauksen näkymä on front-endissä.

```
public static int ToSkipCalculator(int pageNumber, int itemsPerPage,  
int totalObjects)
```

#### Parameters

pageNumber [int](#)

itemsPerPage [int](#)

totalObjects [int](#)

#### Returns

[int](#)

# Namespace Backend\_Reservation\_Service. Middleware

## Classes

[ApiKeyMiddleware](#)

# Class ApiKeyMiddleware

Namespace: [Backend\\_Reservation\\_Service.Middleware](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class ApiKeyMiddleware
```

## Inheritance

[object](#) ← ApiKeyMiddleware

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### ApiKeyMiddleware(RequestDelegate)

```
public ApiKeyMiddleware(RequestDelegate next)
```

## Parameters

next [RequestDelegate](#)

## Methods

### InvokeAsync(HttpContext)

```
public Task InvokeAsync(HttpContext context)
```

## Parameters

context [HttpContext](#)

# Returns

[Task ↗](#)

# Namespace Backend\_Reservation\_Service.Services

## Classes

[AccountService](#)

[AuthenticationService](#)

[AuthorizationService](#)

[NewsService](#)

[ReservationService](#)

[TokenService](#)

[UploadFileService](#)

[VehicleService](#)

## Interfaces

[IAccountService](#)

Interface for account-related operations.

[IAuthorizationService](#)

[INewsService](#)

[IReservationService](#)

[IUploadService](#)

[IVehicleService](#)

# Class AccountService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class AccountService : IAccountService
```

## Inheritance

[object](#) ← AccountService

## Implements

[IAccountService](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### AccountService(IUserRepository)

```
public AccountService(IUserRepository @object)
```

#### Parameters

object [IUserRepository](#)

### AccountService(IUserRepository, IConfiguration, IRoleRepository)

```
public AccountService(IUserRepository repository, IConfiguration configuration,  
IRoleRepository roleRepository)
```

#### Parameters

repository [IUserRepository](#)

configuration  [IConfiguration](#)

roleRepository [IRoleRepository](#)

## Methods

### AssignRoleAsync(string, string)

Assigns a role to a user asynchronously.

```
public Task<IdentityResult> AssignRoleAsync(string username, string role)
```

#### Parameters

username [string](#)

The username of the user.

role [string](#)

The name of the role to assign.

#### Returns

[Task](#)<[IdentityResult](#)>

An IdentityResult indicating the result of the operation.

### ChangePasswordAsync(string, string, string)

```
public Task<(bool Success, string Message)> ChangePasswordAsync(string username, string oldPwd, string newPwd)
```

#### Parameters

username [string](#)

oldPwd [string](#)

newPwd [string](#)

Returns

[Task](#)<([bool](#) [success](#), [string](#) [statusMessage](#))>

## CreateRoleAsync(string)

Creates a new role asynchronously.

```
public Task<(bool Success, string Message)> CreateRoleAsync(string role)
```

Parameters

role [string](#)

The name of the role to create.

Returns

[Task](#)<([bool](#) [success](#), [string](#) [statusMessage](#))>

A tuple indicating success and a message.

## CreateUserAsync(Register)

Creates a new user asynchronously.

```
public Task<(bool Success, string Message, FullUserDTO? FullUserDTO)>  
CreateUserAsync(Register model)
```

Parameters

model [Register](#)

The registration model containing user details.

Returns

[Task](#)<(bool Success, string Message, FullUserDTO FullUserDTO)>

A tuple indicating success and a message.

## DeleteUserAsync(string)

Deletes a user asynchronously.

```
public Task<(bool Success, string Message)> DeleteUserAsync(string username)
```

Parameters

username [string](#)

The username of the user to delete.

Returns

[Task](#)<(bool success, string statusMessage)>

A tuple indicating success and a message.

## GetAllUsersAsync(int, int, bool, bool)

Retrieves all users asynchronously with pagination and ordering options.

```
public Task<(IEnumerable<FullUserDTO>, PaginationDTO pagination)>
GetAllUsersAsync(int itemsPerPage, int pageNumber, bool orderDesc, bool onlyDeleted)
```

Parameters

itemsPerPage [int](#)

The number of items per page.

pageNumber [int](#)

The page number to retrieve.

`orderDesc` [bool](#)

A boolean indicating whether the order should be descending.

`onlyDeleted` [bool](#)

Returns

[Task](#)<([IEnumerable](#)<[FullUserDTO](#)>, [PaginationDTO](#) [pagination](#))>

A tuple containing an enumerable collection of FullUserDTO objects and a PaginationDTO object.

## GetFullUserAsync(string)

Retrieves full user details asynchronously by username.

```
public Task<FullUserDTO?> GetFullUserAsync(string username)
```

Parameters

`username` [string](#)

The username of the user.

Returns

[Task](#)<[FullUserDTO](#)>

A FullUserDTO object containing full user details.

## GetRolesAsync(string)

Retrieves roles for a user asynchronously.

```
public Task<(bool Success, string Message)> GetRolesAsync(string username)
```

Parameters

`username` [string](#)

The username of the user.

Returns

[Task](#)<([bool](#) `success`, [string](#) `statusMessage`)>

A tuple indicating success and a message.

## GetUserAsync(string)

Retrieves a user asynchronously by username.

```
public Task<UserDTO> GetUserAsync(string username)
```

Parameters

`username` [string](#)

The username of the user.

Returns

[Task](#)<[UserDTO](#)>

A UserDTO object containing user details.

## RestoreSoftDelUserAsync(string?)

```
public Task<(bool succeeded, FullUserDTO? userDTO, string message)>
RestoreSoftDelUserAsync(string? username)
```

Parameters

`username` [string](#)

Returns

[Task](#)<([bool](#) succeeded, [FullUserDTO](#) userDTO, [string](#) message)>

## UpdateUserAsync(string, User)

Updates a user asynchronously.

```
public Task<(bool Success, string Message, FullUserDTO? FullUserDTO)>
UpdateUserAsync(string username, User model)
```

### Parameters

username [string](#)

The username of the user to update.

model [User](#)

### Returns

[Task](#)<([bool](#) Success, [string](#) Message, [FullUserDTO](#) FullUserDTO)>

A tuple indicating success and a message.

# Class AuthenticationService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class AuthenticationService
```

## Inheritance

[object](#) ← AuthenticationService

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

AuthenticationService(IUserRepository, TokenService)

```
public AuthenticationService(IUserRepository userRepository,  
TokenService tokenService)
```

## Parameters

userRepository [IUserRepository](#)

tokenService [TokenService](#)

## Methods

AuthenticateAsync(string, string)

```
public Task<string> AuthenticateAsync(string username, string password)
```

## Parameters

username [string](#)

password [string](#)

Returns

[Task](#) <[string](#)>

# Class AuthorizationService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class AuthorizationService : IAuthorizationService
```

## Inheritance

[object](#) ← AuthorizationService

## Implements

[IAuthorizationService](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

AuthorizationService(IUserRepository,  
IReservationRepository)

```
public AuthorizationService(IUserRepository userRepository,  
IReservationRepository reservationService)
```

## Parameters

userRepository [IUserRepository](#)

reservationService [IReservationRepository](#)

## Methods

IsAllowedToManipulateReservationAsync(string, long)

Determines whether the specified user is allowed to manipulate the reservation with the given ID.

```
public Task<bool> IsAllowedToManipulateReservationAsync(string username, long id)
```

## Parameters

**username** [string](#)

The username of the user attempting to manipulate the reservation.

**id** [long](#)

The ID of the reservation to be manipulated.

## Returns

[Task](#)<bool>

A task that represents the asynchronous operation. The task result contains a boolean value indicating whether the user is allowed to manipulate the reservation.

## Remarks

This method first checks if the user exists. If the user does not exist, it returns false. It then retrieves the roles of the user and checks if the user has the "Admin" role. If the user is an admin, it returns true. If the user is not an admin, it retrieves the reservation by its ID. If the reservation does not exist or is deleted, it returns false. Finally, it checks if the reservation is not deleted and if the reservation belongs to the user. If both conditions are met, it returns true.

# Interface IAccountService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

Interface for account-related operations.

```
public interface IAccountService
```

## Methods

### AssignRoleAsync(string, string)

Assigns a role to a user asynchronously.

```
Task<IdentityResult> AssignRoleAsync(string username, string role)
```

#### Parameters

**username** [string](#)

The username of the user.

**role** [string](#)

The name of the role to assign.

#### Returns

[Task](#)<[IdentityResult](#)>

An IdentityResult indicating the result of the operation.

### ChangePasswordAsync(string, string, string)

```
Task<(bool Success, string Message)> ChangePasswordAsync(string username, string oldPwd, string newPwd)
```

## Parameters

username [string](#)

oldPwd [string](#)

newPwd [string](#)

## Returns

[Task](#)<([bool](#) [success](#), [string](#) [statusMessage](#))>

## CreateRoleAsync(string)

Creates a new role asynchronously.

```
Task<(bool Success, string Message)> CreateRoleAsync(string role)
```

## Parameters

role [string](#)

The name of the role to create.

## Returns

[Task](#)<([bool](#) [success](#), [string](#) [statusMessage](#))>

A tuple indicating success and a message.

## CreateUserAsync(Register)

Creates a new user asynchronously.

```
Task<(bool Success, string Message, FullUserDTO? FullUserDTO)>
CreateUserAsync(Register model)
```

## Parameters

## model [Register](#)

The registration model containing user details.

Returns

[Task](#)<([bool](#) [Success](#), [string](#) [Message](#), [FullUserDTO](#) [FullUserDTO](#))>

A tuple indicating success and a message.

## DeleteUserAsync(string)

Deletes a user asynchronously.

[Task](#)<([bool](#) Success, [string](#) Message)> DeleteUserAsync([string](#) username)

Parameters

username [string](#)

The username of the user to delete.

Returns

[Task](#)<([bool](#) [success](#), [string](#) [statusMessage](#))>

A tuple indicating success and a message.

## GetAllUsersAsync(int, int, bool, bool)

Retrieves all users asynchronously.

[Task](#)<([IEnumerable](#)<[FullUserDTO](#)>, [PaginationDTO](#) pagination)> GetAllUsersAsync([int](#) itemsPerPage, [int](#) pageNumber, [bool](#) orderDesc, [bool](#) onlyDeleted)

Parameters

itemsPerPage [int](#)

`pageNumber` [int](#)

`orderDesc` [bool](#)

`onlyDeleted` [bool](#)

Returns

[Task](#)<([IEnumerable](#)<[FullUserDTO](#)>, [PaginationDTO](#) [pagination](#))>

An enumerable collection of UserDTO objects.

## GetFullUserAsync(string)

Retrieves full user details asynchronously by username.

`Task<FullUserDTO?> GetFullUserAsync(string username)`

Parameters

`username` [string](#)

The username of the user.

Returns

[Task](#)<[FullUserDTO](#)>

A FullUserDTO object containing full user details.

## GetRolesAsync(string)

Retrieves roles for a user asynchronously.

`Task<(bool Success, string Message)> GetRolesAsync(string username)`

Parameters

`username` [string](#)

The username of the user.

Returns

[Task](#)<([bool](#) [success](#), [string](#) [statusMessage](#))>

A tuple indicating success and a message.

## GetUserAsync(string)

Retrieves a user asynchronously by username.

[Task](#)<[UserDTO](#)> [GetUserAsync](#)([string](#) [username](#))

Parameters

[username](#) [string](#)

The username of the user.

Returns

[Task](#)<[UserDTO](#)>

A UserDTO object containing user details.

## RestoreSoftDelUserAsync(string?)

[Task](#)<([bool](#) [succeeded](#), [FullUserDTO](#)? [userDTO](#), [string](#) [message](#))>  
[RestoreSoftDelUserAsync](#)([string](#)? [username](#))

Parameters

[username](#) [string](#)

Returns

[Task](#)<([bool](#) [succeeded](#), [FullUserDTO](#) [userDTO](#), [string](#) [message](#))>

# UpdateUserAsync(string, User)

Updates a user asynchronously.

```
Task<(bool Success, string Message, FullUserDTO? FullUserDTO)>
UpdateUserAsync(string username, User user)
```

## Parameters

**username** [string](#)

The username of the user to update.

**user** [User](#)

The user object containing updated details.

## Returns

[Task](#)<(bool Success, string Message, FullUserDTO FullUserDTO)>

A tuple indicating success and a message.

# Interface IAuthorizationService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public interface IAuthorizationService
```

## Methods

### IsAllowedToManipulateReservationAsync(string, long)

```
Task<bool> IsAllowedToManipulateReservationAsync(string username, long id)
```

#### Parameters

username [string](#)

id [long](#)

#### Returns

[Task](#)<bool>

# Interface INewsService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public interface INewsService
```

## Methods

### DeleteNewsAsync(long)

```
Task<(string, bool)> DeleteNewsAsync(long id)
```

#### Parameters

**id** [long](#)

#### Returns

[Task](#)<(string, bool)>

### GetNewsListAsync(bool)

```
Task<IEnumerable<NewsDTO>> GetNewsListAsync(bool all)
```

#### Parameters

**all** [bool](#)

#### Returns

[Task](#)<[IEnumerable](#)<NewsDTO>>

## NewNewsAsync(NewsDTO)

Task<(NewsDTO?, string)> NewNewsAsync(NewsDTO news)

Parameters

news [NewsDTO](#)

Returns

[Task](#)<(NewsDTO, string)>

## UpdateNewsAsync(NewsDTO)

Task<(NewsDTO?, string)> UpdateNewsAsync(NewsDTO news)

Parameters

news [NewsDTO](#)

Returns

[Task](#)<(NewsDTO, string)>

# Interface IReservationService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public interface IReservationService
```

## Methods

### CreateReservationAsync(CreateReservationDTO, string)

```
Task<(ReservationDTO reservation, string statusMessage)>
CreateReservationAsync(CreateReservationDTO reservation, string username)
```

#### Parameters

reservation [CreateReservationDTO](#)

username [string](#)

#### Returns

[Task](#)<(ReservationDTO reservation, string statusMessage)>

### DeleteReservationAsync(long, string)

```
Task<(bool success, string statusMessage)> DeleteReservationAsync(long
reservationId, string username)
```

#### Parameters

reservationId [long](#)

username [string](#)

Returns

[Task](#)<(bool success, string statusMessage)>

## GetAllReservationsAsync(int, int, bool, int)

```
Task<(IEnumerable<FullReservationDTO>, PaginationDTO pagination)>
GetAllReservationsAsync(int itemsPerPage, int pageNumber, bool orderDesc,
int deletedIncludeMode)
```

Parameters

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

deletedIncludeMode [int](#)

Returns

[Task](#)<(IEnumerable<FullReservationDTO>, PaginationDTO pagination)>

## GetReservationByIdAsync(long)

```
Task<Reservation> GetReservationByIdAsync(long reservationId)
```

Parameters

reservationId [long](#)

Returns

[Task](#)<[Reservation](#)>

## GetReservationDTOByIdAsync(long)

```
Task<ReservationDTO> GetReservationDTOByIdAsync(long reservationId)
```

## Parameters

reservationId [long](#)

## Returns

[Task](#)<[ReservationDTO](#)>

## GetReservationsByUserAsync(string, int, int, bool, bool, bool)

```
Task<(IEnumerable<ReservationDTO>, string statusMessage, PaginationDTO pagination)>
GetReservationsByUserAsync(string userName, int itemsPerPage, int pageNumber, bool
orderDesc, bool onlyActive, bool onlyPast)
```

## Parameters

userName [string](#)

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

onlyActive [bool](#)

onlyPast [bool](#)

## Returns

[Task](#)<(IEnumerable<[ReservationDTO](#)>, [string](#) [statusMessage](#), [PaginationDTO](#) [pagination](#))>

## GetReservationsByVehicleAsync(long?, int, int, bool)

```
Task<(IEnumerable<ReservationDTO>, string statusMessage, PaginationDTO? pagination)>
GetReservationsByVehicleAsync(long? vehicleId, int itemsPerPage, int pageNumber,
bool orderDesc)
```

## Parameters

vehicleId [long](#)?

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

## Returns

[Task](#)<(IEnumerable<[ReservationDTO](#)>, [string](#) [statusMessage](#), [PaginationDTO](#) [pagination](#))>

# Interface IUploadService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public interface IUploadService
```

## Methods

### GetUniqueFileName(string)

```
string GetUniqueFileName(string fileName)
```

#### Parameters

fileName [string](#)

#### Returns

[string](#)

### SaveFileAsync(UploadFilePostDTO)

```
Task<(bool Success, string Message, UploadFileDTO UploadFileDTO)>
SaveFileAsync(UploadFilePostDTO uploadFile)
```

#### Parameters

uploadFile [UploadFilePostDTO](#)

#### Returns

[Task](#)<(bool Success, string Message, UploadFileDTO UploadFileDTO)>

## UploadFileAsync(UploadFileDTO)

```
Task<(bool Success, string Message, UploadFileDTO UploadFileDTO)>
UploadFileAsync(UploadFileDTO uploadfileDto)
```

### Parameters

uploadfileDto [UploadFileDTO](#)

### Returns

[Task](#)<(bool Success, string Message, [UploadFileDTO](#) UploadFileDTO)>

# Interface IVehicleService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public interface IVehicleService
```

## Methods

### DeleteVehicleAsync(long)

```
Task<bool> DeleteVehicleAsync(long vehicleId)
```

#### Parameters

vehicleId [long](#)

#### Returns

[Task](#)<[bool](#)>

### DeleteVehicleAsync(string)

```
Task<bool> DeleteVehicleAsync(string vehicleName)
```

#### Parameters

vehicleName [string](#)

#### Returns

[Task](#)<[bool](#)>

## GetDdMakeListAsync(string?, string?)

Task<List<string>> GetDdMakeListAsync(string? type, string? model)

Parameters

type [string](#)

model [string](#)

Returns

[Task](#)<[List](#)<[string](#)>>

## GetDdModelListAsync(string?, string?)

Task<List<string>> GetDdModelListAsync(string? type, string? make)

Parameters

type [string](#)

make [string](#)

Returns

[Task](#)<[List](#)<[string](#)>>

## GetDdTypeListAsync(string?, string?)

Task<List<string>> GetDdTypeListAsync(string? make, string? model)

Parameters

make [string](#)

model [string](#)

Returns

[Task](#)<[List](#)<[string](#)>>

## GetDelVehicleListAsync(int, int, bool?)

Task<(IEnumerable<Vehicle>, PaginationDTO)> GetDelVehicleListAsync([int](#) itemsPerPage, [int](#) page, [bool](#)? oldestFirst)

Parameters

[itemsPerPage](#) [int](#)

[page](#) [int](#)

[oldestFirst](#) [bool](#)?

Returns

[Task](#)<(IEnumerable<Vehicle>, PaginationDTO)>

## GetVehicleByIdAsync(long)

Task<(VehicleDTO?, [string](#))> GetVehicleByIdAsync([long](#) id)

Parameters

[id](#) [long](#)

Returns

[Task](#)<(VehicleDTO, [string](#))>

## GetVehicleByLicenseplateAsync(string)

Task<VehicleDTO?> GetVehicleByLicenseplateAsync([string](#) licenseplate)

Parameters

licenseplate [string](#)?

Returns

[Task](#)<[VehicleDTO](#)>

## GetVehicleByNicknameAsync(string)

`Task<VehicleDTO?> GetVehicleByNicknameAsync(string name)`

Parameters

name [string](#)?

Returns

[Task](#)<[VehicleDTO](#)>

## GetVehiclesAsAdminAsync(bool?, bool?, bool?, string?, string?, string?, string?, DateOnly?, DateOnly?, bool?, int, int)

`Task<(IEnumerable<Vehicle>, PaginationDTO)> GetVehiclesAsAdminAsync(bool?  
includeDeleted, bool? onlyEmptyDescriptions, bool? onlyEmptyNicknames, string? type,  
string? make, string? model, string? year, DateOnly? dtmStart, DateOnly? dtmEnd,  
bool? includeNotUsable, int itemsPerPage, int page)`

Parameters

includeDeleted [bool](#)?

onlyEmptyDescriptions [bool](#)?

onlyEmptyNicknames [bool](#)?

type [string](#)?

```
make string
model string
year string
dtmStart DateOnly?
dtmEnd DateOnly?
includeNotUsable bool?
itemsPerPage int
page int
```

Returns

```
Task<(IEnumerable<Vehicle>, PaginationDTO)>
```

## GetVehiclesAsync(string?, string?, string?, string?, DateOnly?, DateOnly?, bool?, int, int)

```
Task<(IEnumerable<VehicleDTO>, PaginationDTO)> GetVehiclesAsync(string? type, string? make, string? model, string? year, DateOnly? dtmStart, DateOnly? dtmEnd, bool? includeNotUsable, int itemsPerPage, int page)
```

Parameters

```
type string
make string
model string
year string
dtmStart DateOnly?
dtmEnd DateOnly?
includeNotUsable bool?
```

`itemsPerPage` [int](#)

`page` [int](#)

Returns

[Task](#)<([IEnumerable](#)<[VehicleDTO](#)>, [PaginationDTO](#))>

## NewVehicleAsync(VehicleDTO)

`Task<(VehicleDTO?, string)> NewVehicleAsync(VehicleDTO vehicle)`

Parameters

`vehicle` [VehicleDTO](#)

Returns

[Task](#)<([VehicleDTO](#), [string](#))>

## RestoreSoftDelVehicleAsync(int)

`Task<(VehicleDTO?, string)> RestoreSoftDelVehicleAsync(int id)`

Parameters

`id` [int](#)

Returns

[Task](#)<([VehicleDTO](#), [string](#))>

## UpdateVehicleAsync(VehicleDTO)

`Task<(bool, string)> UpdateVehicleAsync(VehicleDTO vehicle)`

## Parameters

`vehicle` [VehicleDTO](#)

## Returns

[Task](#)<(`bool` [success](#), `string` [statusMessage](#))>

# Class NewsService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class NewsService : INewsService
```

## Inheritance

[object](#) ← NewsService

## Implements

[INewsService](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### NewsService(INewsRepository)

```
public NewsService(INewsRepository newsRepository)
```

## Parameters

newsRepository [INewsRepository](#)

## Methods

### DeleteNewsAsync(long)

```
public Task<(string, bool)> DeleteNewsAsync(long id)
```

## Parameters

`id` [long](#)

Returns

[Task](#)<([string](#), [bool](#))>

## GetNewsListAsync(bool)

```
public Task<IEnumerable<NewsDTO>> GetNewsListAsync(bool all)
```

Parameters

`all` [bool](#)

Returns

[Task](#)<[IEnumerable](#)<NewsDTO>>

## NewNewsAsync(NewsDTO)

```
public Task<(NewsDTO?, string)> NewNewsAsync(NewsDTO news)
```

Parameters

`news` [NewsDTO](#)

Returns

[Task](#)<([NewsDTO](#), [string](#))>

## UpdateNewsAsync(NewsDTO)

```
public Task<(NewsDTO?, string)> UpdateNewsAsync(NewsDTO news)
```

Parameters

news [NewsDTO](#)

Returns

[Task](#)<([NewsDTO](#), [string](#))>

# Class ReservationService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class ReservationService : IReservationService
```

## Inheritance

[object](#) ← ReservationService

## Implements

[IReservationService](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

ReservationService(IReservationRepository,  
IUserRepository, IVehicleRepository,  
IAuthorizationService)

```
public ReservationService(IReservationRepository reservationRepository,  
IUserRepository userRepository, IVehicleRepository vehicleRepository,  
IAuthorizationService authorizationService)
```

## Parameters

reservationRepository [IReservationRepository](#)

userRepository [IUserRepository](#)

vehicleRepository [IVehicleRepository](#)

authorizationService [IAuthorizationService](#)

# Methods

## CreateReservationAsync(CreateReservationDTO, string)

```
public Task<(ReservationDTO? reservation, string statusMessage)>
CreateReservationAsync(CreateReservationDTO newReservation, string username)
```

### Parameters

newReservation [CreateReservationDTO](#)

username [string](#)

### Returns

[Task](#)<(ReservationDTO reservation, string statusMessage)>

## DeleteReservationAsync(long, string)

```
public Task<(bool success, string statusMessage)> DeleteReservationAsync(long
reservationId, string username)
```

### Parameters

reservationId [long](#)

username [string](#)

### Returns

[Task](#)<(bool success, string statusMessage)>

## GetAllReservationsAsync(int, int, bool, int)

```
public Task<(IEnumerable<FullReservationDTO>, PaginationDTO pagination)>
GetAllReservationsAsync(int itemsPerPage, int pageNumber, bool orderDesc,
int deletedIncludeMode)
```

Parameters

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

deletedIncludeMode [int](#)

Returns

[Task](#)<([IEnumerable](#)<[FullReservationDTO](#)>, [PaginationDTO](#) [pagination](#))>

## GetReservationByIdAsync(long)

`public Task<Reservation> GetReservationByIdAsync(long reservationId)`

Parameters

reservationId [long](#)

Returns

[Task](#)<[Reservation](#)>

## GetReservationDTOByIdAsync(long)

`public Task<ReservationDTO> GetReservationDTOByIdAsync(long reservationId)`

Parameters

reservationId [long](#)

Returns

[Task](#)<[ReservationDTO](#)>

## GetReservationsByUserAsync(string, int, int, bool, bool)

```
public Task<(IEnumerable<ReservationDTO>, string statusMessage, PaginationDTO pagination)> GetReservationsByUserAsync(string userName, int itemsPerPage, int pageNumber, bool orderDesc, bool onlyActive, bool onlyPast)
```

### Parameters

userName [string](#)

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

onlyActive [bool](#)

onlyPast [bool](#)

### Returns

[Task](#)<(IEnumerable<[ReservationDTO](#)>, [string](#) [statusMessage](#), [PaginationDTO](#) [pagination](#))>

## GetReservationsByVehicleAsync(long?, int, int, bool)

```
public Task<(IEnumerable<ReservationDTO>, string statusMessage, PaginationDTO pagination)> GetReservationsByVehicleAsync(long? vehicleId, int itemsPerPage, int pageNumber, bool orderDesc)
```

### Parameters

vehicleId [long](#)?

itemsPerPage [int](#)

pageNumber [int](#)

orderDesc [bool](#)

## Returns

[Task](#)<([IEnumerable](#)<[ReservationDTO](#)>, [string](#) [statusMessage](#), [PaginationDTO](#)  
[pagination](#))>

# Class TokenService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class TokenService
```

## Inheritance

[object](#) ← TokenService

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### TokenService(IConfiguration)

```
public TokenService(IConfiguration configuration)
```

## Parameters

configuration [IConfiguration](#)

## Methods

### GenerateJwtToken(string, IEnumerable<string>)

```
public string GenerateJwtToken(string username, IEnumerable<string> roles)
```

## Parameters

username [string](#)

roles [IEnumerable](#)<[string](#)>

Returns

[string](#)

# Class UploadFileService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class UploadFileService : IUploadService
```

## Inheritance

[object](#) ↗ ← UploadFileService

## Implements

[IUploadService](#)

## Inherited Members

[object.Equals\(object\)](#) ↗ , [object.Equals\(object, object\)](#) ↗ , [object.GetHashCode\(\)](#) ↗ ,  
[object.GetType\(\)](#) ↗ , [object.MemberwiseClone\(\)](#) ↗ , [object.ReferenceEquals\(object, object\)](#) ↗ ,  
[object.ToString\(\)](#) ↗

## Constructors

UploadFileService(IWebHostEnvironment,  
 IConfiguration, IUserRepository, IVehicleRepository,  
 IUploadFileRepository)

```
public UploadFileService(IWebHostEnvironment webHostEnvironment, IConfiguration  
 configuration, IUserRepository userRepository, IVehicleRepository vehicleRepository,  
 IUploadFileRepository uploadFileRepository)
```

## Parameters

webHostEnvironment [IWebHostEnvironment](#) ↗

configuration  [IConfiguration](#) ↗

userRepository [IUserRepository](#)

vehicleRepository [IVehicleRepository](#)

`uploadFileRepository` [IUploadFileRepository](#)

## Methods

### GetUniqueFileName(string)

`public string GetUniqueFileName(string fileName)`

#### Parameters

`fileName` [string](#)

#### Returns

[string](#)

### SaveFileAsync(UploadFilePostDTO)

Saves the uploaded file asynchronously.

`public Task<(bool Success, string Message, UploadFileDTO UploadFileDTO)> SaveFileAsync(UploadFilePostDTO uploadFile)`

#### Parameters

`uploadFile` [UploadFilePostDTO](#)

The DTO containing the file to be uploaded and related information.

#### Returns

[Task](#)<([bool](#) [Success](#), [string](#) [Message](#), [UploadFileDTO](#) [UploadFileDTO](#))>

A tuple containing a boolean indicating success, and a message.

#### Exceptions

[Exception](#)

Throws an exception if there is an error during file saving.

## UploadFileAsync(UploadFileDTO)

```
public Task<(bool Success, string Message, UploadFileDTO UploadFileDTO)>
UploadFileAsync(UploadFileDTO uploadfileDto)
```

### Parameters

uploadfileDto [UploadFileDTO](#)

### Returns

[Task](#)<(bool Success, string Message, [UploadFileDTO](#) UploadFileDTO)>

# Class VehicleService

Namespace: [Backend\\_Reservation\\_Service.Services](#)

Assembly: Backend\_Reservation\_Service.dll

```
public class VehicleService : IVehicleService
```

## Inheritance

[object](#) ← VehicleService

## Implements

[IVehicleService](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Constructors

### VehicleService(IVehicleRepository)

```
public VehicleService(IVehicleRepository vehicleRepository)
```

## Parameters

vehicleRepository [IVehicleRepository](#)

## Methods

### DeleteVehicleAsync(long)

```
public Task<bool> DeleteVehicleAsync(long vehicleId)
```

## Parameters

`vehicleId` [long](#)

Returns

[Task](#)<[bool](#)>

## DeleteVehicleAsync(string)

```
public Task<bool> DeleteVehicleAsync(string vehicleName)
```

Parameters

`vehicleName` [string](#)

Returns

[Task](#)<[bool](#)>

## GetDdMakeListAsync(string?, string?)

```
public Task<List<string>> GetDdMakeListAsync(string? type, string? model)
```

Parameters

`type` [string](#)

`model` [string](#)

Returns

[Task](#)<[List](#)<[string](#)>>

## GetDdModelListAsync(string?, string?)

```
public Task<List<string>> GetDdModelListAsync(string? type, string? make)
```

Parameters

type [string](#)

make [string](#)

Returns

[Task](#)<[List](#)<[string](#)>>

## GetDdTypeListAsync(string?, string?)

```
public Task<List<string>> GetDdTypeListAsync(string? make, string? model)
```

Parameters

make [string](#)

model [string](#)

Returns

[Task](#)<[List](#)<[string](#)>>

## GetDelVehicleListAsync(int, int, bool?)

```
public Task<(IEnumerable<Vehicle>, PaginationDTO)> GetDelVehicleListAsync(int itemsPerPage, int page, bool? oldestFirst)
```

Parameters

itemsPerPage [int](#)

page [int](#)

oldestFirst [bool](#)?

Returns

[Task](#)<([IEnumerable](#)<[Vehicle](#)>, [PaginationDTO](#))>

## GetVehicleByIdAsync(long)

`public Task<(VehicleDTO?, string)> GetVehicleByIdAsync(long id)`

### Parameters

`id` [long](#)

### Returns

[Task](#)<([VehicleDTO](#), [string](#))>

## GetVehicleByLicenseplateAsync(string)

`public Task<VehicleDTO?> GetVehicleByLicenseplateAsync(string licenseplate)`

### Parameters

`licenseplate` [string](#)

### Returns

[Task](#)<[VehicleDTO](#)>

## GetVehicleByNicknameAsync(string)

`public Task<VehicleDTO?> GetVehicleByNicknameAsync(string nickname)`

### Parameters

`nickname` [string](#)

### Returns

[Task](#) <[VehicleDTO](#)>

**GetVehiclesAsAdminAsync(bool?, bool?, bool?, string?, string?, string?, DateOnly?, DateOnly?, bool?, int, int)**

```
public Task<(IEnumerable<Vehicle>, PaginationDTO)> GetVehiclesAsAdminAsync(bool?  
includeDeleted, bool? onlyEmptyDescriptions, bool? onlyEmptyNicknames, string? type,  
string? make, string? model, string? year, DateOnly? dtmStart, DateOnly? dtmEnd,  
bool? includeNotUsable, int itemsPerPage, int page)
```

Parameters

includeDeleted [bool](#)?

onlyEmptyDescriptions [bool](#)?

onlyEmptyNicknames [bool](#)?

type [string](#)

make [string](#)

model [string](#)

year [string](#)

dtmStart [DateOnly](#)?

dtmEnd [DateOnly](#)?

includeNotUsable [bool](#)?

itemsPerPage [int](#)

page [int](#)

Returns

[Task](#) <([IEnumerable](#) <[Vehicle](#)>, [PaginationDTO](#))>

## GetVehiclesAsync(string?, string?, string?, string?, DateOnly?, DateOnly?, bool?, int, int)

```
public Task<(IEnumerable<VehicleDTO>, PaginationDTO)> GetVehiclesAsync(string? type,  
string? make, string? model, string? year, DateOnly? dtmStart, DateOnly? dtmEnd,  
bool? includeNotUsable, int itemsPerPage, int page)
```

### Parameters

type [string](#)?

make [string](#)?

model [string](#)?

year [string](#)?

dtmStart [DateOnly](#)?

dtmEnd [DateOnly](#)?

includeNotUsable [bool](#)?

itemsPerPage [int](#)

page [int](#)

### Returns

[Task](#)<(IEnumerable<[VehicleDTO](#)>, [PaginationDTO](#))>

## NewVehicleAsync(VehicleDTO)

```
public Task<(VehicleDTO?, string)> NewVehicleAsync(VehicleDTO vehicle)
```

### Parameters

vehicle [VehicleDTO](#)

### Returns

[Task](#)<([VehicleDTO](#), [string](#))>

## RestoreSoftDelVehicleAsync(int)

`public Task<(VehicleDTO?, string)> RestoreSoftDelVehicleAsync(int id)`

Parameters

`id int`

Returns

[Task](#)<([VehicleDTO](#), [string](#))>

## UpdateVehicleAsync(VehicleDTO)

`public Task<(bool, string)> UpdateVehicleAsync(VehicleDTO vehicle)`

Parameters

`vehicle VehicleDTO`

Returns

[Task](#)<([bool](#) [success](#), [string](#) [statusMessage](#))>