Namespace opcUaWebMVC

Classes

ApplicationContext

Logger

Class ApplicationContext

Namespace: opcUaWebMVC

Assembly: opcUaWebMVC.dll

```
public sealed class ApplicationContext : DbContext, IInfrastructure<IServiceProvider>,
IDbContextDependencies, IDbSetCache, IDbContextPoolable, IResettableService,
IDisposable, IAsyncDisposable
```

Inheritance

object ← DbContext ← ApplicationContext

Implements

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

Inherited Members

DbContext.Set<TEntity>() , DbContext.Set<TEntity>(string) , 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<IQueryable<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.ReferenceEquals(object, object) , object.ToString()

Constructors

ApplicationContext()

```
public ApplicationContext()
```

Properties

Locations

```
public DbSet<Location> Locations { get; set; }
```

Property Value

DbSet<Location>

Plcs

```
public DbSet<Plc> Plcs { get; set; }
```

Property Value

DbSet<Plc>

Users

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

Property Value

DbSet <User>

Methods

OnConfiguring(DbContextOptionsBuilder)

Override this method to configure the database (and other options) to be used for this context. This method is called for each instance of the context that is created. The base implementation does nothing.

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
```

Parameters

`optionsBuilder` DbContextOptionsBuilder

A builder used to create or modify options for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure the context.

Remarks

In situations where an instance of DbContextOptions may or may not have been passed to the constructor, you can use IsConfigured to determine if the options have already been set, and skip some or all of the logic in OnConfiguring(DbContextOptionsBuilder).

See DbContext lifetime, configuration, and initialization for more information and examples.

Class Logger

Namespace: opcUaWebMVC

Assembly: opcUaWebMVC.dll

```
public static class Logger
```

Inheritance

object ← Logger

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

debug(string, string)

```
public static void debug(string tag, string value)
```

Parameters

`tag` string

`value` string

param(int, float, int)

```
public static void param(int paramId, float value, int locationId)
```

Parameters

`paramId` int

`value` float

`locationId` int

system(string, string)

```
public static void system(string tag, string value)
```

Parameters

`tag` string

`value` string

Namespace opcUaWebMVC.Controllers

Classes

AuthController

Controller-класс для обработки страницы авторизации пользователя

HomeController

LocationController

PlcController

UserController

UserSingleton

Singleton-класс для хранения текущего пользователя

Class AuthController

Namespace: opcUaWebMVC.Controllers

Assembly: opcUaWebMVC.dll

Controller-класс для обработки страницы авторизации пользователя

```
public class AuthController : Controller, IActionFilter, IAsyncActionFilter,
IFilterMetadata, IDisposable
```

Inheritance

object ← ControllerBase ← Controller ← AuthController

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.ValidationProblem(ValidationProblemDetails) ,
ControllerBase.ValidationProblem(ModelStateDictionary) , ControllerBase.ValidationProblem() ,
ControllerBase.ValidationProblem(string, string, int?, string, string, ModelStateDictionary) ,
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()

Methods

AuthPage()

```
[HttpGet]
public IActionResult AuthPage()
```

Returns

IActionResult

AuthPage(string, string)

Метод для получения данных входа пользователя. Сравнивает данные с БД

```
[HttpPost]
public IActionResult AuthPage(string login, string passwd)
```

Parameters

`login` string

Логин пользователя

`passwd` string

Пароль пользователя

Returns

IActionResult

Переадрессация на главную страницу или страницу входа

Logout()

```
public IActionResult Logout()
```

Returns

IActionResult

Class HomeController

Namespace: opcUaWebMVC.Controllers

Assembly: opcUaWebMVC.dll

```
public class HomeController : Controller, IActionFilter, IAsyncActionFilter,
IFilterMetadata, IDisposable
```

Inheritance

object ← ControllerBase ← Controller ← HomeController

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.ValidationProblem(ValidationProblemDetails) ,
ControllerBase.ValidationProblem(ModelStateDictionary) , ControllerBase.ValidationProblem() ,
ControllerBase.ValidationProblem(string, string, int?, string, string, ModelStateDictionary) ,
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

HomeController(ILogger<HomeController>)

```
public HomeController(ILogger<HomeController> logger)
```

Parameters

`logger` ILogger<HomeController>

Methods

Error()

```
[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
public IActionResult Error()
```

Returns

IActionResult

Index()

```
public IActionResult Index()
```

Returns

IActionResult

Privacy()

```
public IActionResult Privacy()
```

Returns

IActionResult

Class LocationController

Namespace: opcUaWebMVC.Controllers

Assembly: opcUaWebMVC.dll

```
public class LocationController : Controller, IActionFilter, IAsyncActionFilter,
IFilterMetadata, IDisposable
```

Inheritance

object ← ControllerBase ← Controller ← LocationController

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.ValidationProblem(ValidationProblemDetails) ,
ControllerBase.ValidationProblem(ModelStateDictionary) , ControllerBase.ValidationProblem() ,
ControllerBase.ValidationProblem(string, string, int?, string, string, ModelStateDictionary) ,
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()

Methods

AddLocationPage()

```
[HttpGet]
public IActionResult AddLocationPage()
```

Returns

IActionResult

AddLocationPage(string)

```
[HttpPost]
public IActionResult AddLocationPage(string name)
```

Parameters

`name` string

Returns

IActionResult

EditLocation(int)

```
[HttpGet]
public IActionResult EditLocation(int id)
```

Parameters

`id` int

Returns

IActionResult

EditLocation(string, int, string)

```
[HttpPost]
public IActionResult EditLocation(string name, int id, string action)
```

Parameters

`name` string

`id` int

`action` string

Returns

IActionResult

LocationList()

```
[HttpGet]
public IActionResult LocationList()
```

Returns

IActionResult

Class PlcController

Namespace: opcUaWebMVC.Controllers

Assembly: opcUaWebMVC.dll

```
public class PlcController : Controller, IActionFilter, IAsyncActionFilter,
IFilterMetadata, IDisposable
```

Inheritance

object ← ControllerBase ← Controller ← PlcController

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.ValidationProblem(ValidationProblemDetails) ,
ControllerBase.ValidationProblem(ModelStateDictionary) , ControllerBase.ValidationProblem() ,
ControllerBase.ValidationProblem(string, string, int?, string, string, ModelStateDictionary) ,
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()

Methods

AddPlcPage()

```
[HttpGet]
public IActionResult AddPlcPage()
```

Returns

IActionResult

AddPlcPage(string, string, int, string)

```
[HttpPost]
public IActionResult AddPlcPage(string ip, string port, int locationId, string name)
```

Parameters

ip string

port string

locationId int

name string

Returns

IActionResult

EditPlc(int)

```
[HttpGet]
public IActionResult EditPlc(int id)
```

Parameters

id int

Returns

IActionResult

EditPlc(string, int, string, string, int, string)

```
[HttpPost]
public IActionResult EditPlc(string name, int locationId, string ip, string port, int id,
string action)
```

Parameters

`name` string

`locationId` int

`ip` string

`port` string

`id` int

`action` string

Returns

IActionResult

PlcList()

```
public IActionResult PlcList()
```

Returns

IActionResult

Class UserController

Namespace: opcUaWebMVC.Controllers

Assembly: opcUaWebMVC.dll

```
public class UserController : Controller, IActionFilter, IAsyncActionFilter,
IFilterMetadata, IDisposable
```

Inheritance

object ← ControllerBase ← Controller ← UserController

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.ValidationProblem(ValidationProblemDetails) ,
ControllerBase.ValidationProblem(ModelStateDictionary) , ControllerBase.ValidationProblem() ,
ControllerBase.ValidationProblem(string, string, int?, string, string, ModelStateDictionary) ,
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()

Methods

AddUserPage()

```
[HttpGet]
public IActionResult AddUserPage()
```

Returns

IActionResult

AddUserPage(string, string, string, bool)

```
[HttpPost]
public IActionResult AddUserPage(string name, string login, string passwd, bool admin)
```

Parameters

`name` string

`login` string

`passwd` string

`admin` bool

Returns

IActionResult

EditUser(int)

```
[HttpGet]
public IActionResult EditUser(int id)
```

Parameters

`id` int

Returns

IActionResult

EditUser(string, string, string, bool, int, string)

```
[HttpPost]
public IActionResult EditUser(string name, string login, string passwd, bool admin, int id,
string action)
```

Parameters

`name` string

`login` string

`passwd` string

`admin` bool

`id` int

`action` string

Returns

IActionResult

UserList()

```
public IActionResult UserList()
```

Returns

IActionResult

Class UserSingleton

Namespace: opcUaWebMVC.Controllers

Assembly: opcUaWebMVC.dll

Singleton-класс для хранения текущего пользователя

```
public class UserSingleton
```

Inheritance

object ← UserSingleton

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

currentUser

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

Property Value

User

Methods

GetInstance()

```
public static UserSingleton GetInstance()
```

Returns

UserSingleton

Namespace opcUaWebMVC.Models

Classes

ErrorViewModel

User

Class ErrorViewModel

Namespace: opcUaWebMVC.Models

Assembly: opcUaWebMVC.dll

```
public class ErrorViewModel
```

Inheritance

object ← ErrorViewModel

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

RequestId

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

Property Value

string

ShowRequestId

```
public bool ShowRequestId { get; }
```

Property Value

bool

Class User

Namespace: opcUaWebMVC.Models

Assembly: opcUaWebMVC.dll

```
[Table("user")]
public class User
```

Inheritance

object ← User

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

User(string, string, string, bool)

```
public User(string name, string login, string passwd, bool isAdmin)
```

Parameters

`name` string

`login` string

`passwd` string

`isAdmin` bool

Properties

id

```
[Column("id")]
```

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

Property Value

int

isAdmin

```
[Column("is_admin")]
public bool isAdmin { get; set; }
```

Property Value

bool

login

```
[Column("login")]
public string login { get; set; }
```

Property Value

string

name

```
[Column("name")]
public string name { get; set; }
```

Property Value

string

passwd

```
[Column("password")]
public string passwd { get; set; }
```

Property Value

string

Namespace opcUaWebMVC.Models.Database

Classes

ControlParameter

Location

Plc

Class ControlParameter

Namespace: opcUaWebMVC.Models.Database

Assembly: opcUaWebMVC.dll

```
[Table("control_parameter")]
public class ControlParameter
```

Inheritance

object ← ControlParameter

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

ControlParameter(string, float, float, string, string)

```
public ControlParameter(string name, float minValue, float maxValue, string opcNameMin,
string opcNameMax)
```

Parameters

`name` string

`minValue` float

`maxValue` float

`opcNameMin` string

`opcNameMax` string

Properties

id

```
[Column("id")]
public int id { get; set; }
```

Property Value

int

maxValue

```
[Column("max_value")]
public float maxValue { get; set; }
```

Property Value

float

minValue

```
[Column("min_value")]
public float minValue { get; set; }
```

Property Value

float

name

```
[Column("name")]
public string name { get; set; }
```

Property Value

string

opcNameMax

```
[Column("opc_name_max")]
public string opcNameMax { get; set; }
```

Property Value

string

opcNameMin

```
[Column("opc_name_min")]
public string opcNameMin { get; set; }
```

Property Value

string

Class Location

Namespace: opcUaWebMVC.Models.Database

Assembly: opcUaWebMVC.dll

```
[Table("location")]
public class Location
```

Inheritance

object ← Location

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

Location()

```
public Location()
```

Location(string)

```
public Location(string name)
```

Parameters

`name` string

Properties

Plcs

```
public ICollection<Plc> Plcs { get; set; }
```

Property Value

ICollection<Plc>

id

```
[Column("id")]
public int id { get; set; }
```

Property Value

int

name

```
[Column("name")]
public string name { get; set; }
```

Property Value

string

Methods

ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

string

A string that represents the current object.

Class Plc

Namespace: opcUaWebMVC.Models.Database

Assembly: opcUaWebMVC.dll

```
[Table("plc")]
public class Plc
```

Inheritance

object ← Plc

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

Plc(string, string, int, string)

```
public Plc(string ip, string port, int locationId, string name)
```

Parameters

`ip` string

`port` string

`locationId` int

`name` string

Properties

id

```
[Column("id")]
```

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

Property Value

int

ip

```
[Column("ip")]
public string ip { get; set; }
```

Property Value

string

location

```
public Location location { get; set; }
```

Property Value

Location

locationId

```
[Column("location_id")]
public int locationId { get; set; }
```

Property Value

int

name

```
[Column("name")]
public string name { get; set; }
```

Property Value

string

port

```
[Column("port")]
public string port { get; set; }
```

Property Value

string

Methods

ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

string

A string that represents the current object.