

# Namespace ChessServer.Controllers

## Classes

[BaseController](#)

[ChessController](#)

[UserController](#)

# Class BaseController

Namespace: [ChessServer.Controllers](#)

Assembly: ChessServer.dll

```
public class BaseController : ControllerBase
```

## Inheritance

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

## Derived

[ChessController](#), [UserController](#)

## 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\(string, string, DateTimeOffset?\)](#)  ,  [ControllerBase.File\(string, string, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string\)](#)  ,  [ControllerBase.PhysicalFile\(string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?\)](#)  ,  [ControllerBase.PhysicalFile\(string, string, EntityTagHeaderValue\)](#)  ,

[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.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

### BaseController()

```
public BaseController()
```

## Methods

### CreateToken(User, IConfiguration)

```
[NonAction]  
public string CreateToken(User user, IConfiguration configuration)
```

Parameters

`user` [User](#)

`configuration`  [IConfiguration](#)

Returns

[string](#)

`decodeToken(string)`

`[NonAction]`

`public JwtSecurityToken decodeToken(string token)`

Parameters

`token` [string](#)

Returns

[JwtSecurityToken](#)

# Class ChessController

Namespace: [ChessServer.Controllers](#)

Assembly: ChessServer.dll

```
[ApiController]
[Route("api/chess")]
public class ChessController : BaseController
```

## Inheritance

[object](#) ← [ControllerBase](#) ← [BaseController](#) ← [ChessController](#)

## Inherited Members

[BaseController.CreateToken\(User, IConfiguration\)](#), [BaseController.decodeToken\(string\)](#),  
[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\(string, string, DateTimeOffset?\)](#)  ,  [ControllerBase.File\(string, string, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string\)](#)  ,  [ControllerBase.PhysicalFile\(string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, string, bool\)](#)  ,  
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?\)](#)  ,  [ControllerBase.PhysicalFile\(string, string, EntityTagHeaderValue\)](#)  ,

[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.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

ChessController(IChessDataService, IStockFishService, LinkGenerator)

```
public ChessController(IChessDataService chessDataService, IStockFishService  
stockFishService, LinkGenerator linkGenerator)
```

## Parameters

chessDataService [IChessDataService](#)

stockFishService [IStockFishService](#)

linkGenerator [LinkGenerator](#)

## Methods

### CreateBotGame(CreateBotChessModel)

```
[Authorize]
[HttpPost]
[Route("newbotgame")]
public Task<IActionResult> CreateBotGame(CreateBotChessModel model)
```

#### Parameters

model [CreateBotChessModel](#)

#### Returns

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

### CreateGame(CreateChessModel)

```
[HttpPost]
[Route("new")]
public Task<IActionResult> CreateGame(CreateChessModel model)
```

#### Parameters

model [CreateChessModel](#)

#### Returns

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

### Move(int, MoveModel)

```
[HttpPut]
[Route("{id}/move")]
public Task<IActionResult> Move(int id, MoveModel moveModel)
```

Parameters

`id int`

`moveModel MoveModel`

Returns

`Task<ActionResult>`

## MoveBot(int)

```
[HttpPost]  
[Route("{id}/moveBot")]  
public Task<IActionResult> MoveBot(int id)
```

Parameters

`id int`

Returns

`Task<ActionResult>`

## StartStocky()

```
[HttpGet]  
[Route("stockfish")]  
public Task<IActionResult> StartStocky()
```

Returns

`Task<ActionResult>`

# Class UserController

Namespace: [ChessServer.Controllers](#)

Assembly: ChessServer.dll

```
[ApiController]
[Route("api/user")]
public class UserController : BaseController
```

## Inheritance

[object](#) ← [ControllerBase](#) ← [BaseController](#) ← UserController

## Inherited Members

[BaseController.CreateToken\(User, IConfiguration\)](#), [BaseController.decodeToken\(string\)](#),  
[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\(string, string, DateTimeOffset?\)](#)  ,  [ControllerBase.File\(string, string, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#)  ,  
 [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.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

### UserController(IDataService, IConfiguration)

```
public UserController(IDataService dataService, IConfiguration configuration)
```

#### Parameters

`dataService` [IDataService](#)

`configuration` [IConfiguration](#)

## Methods

## ClearJwtCookie(HttpResponse)

```
[NonAction]  
public void ClearJwtCookie(HttpContext response)
```

### Parameters

response [HttpResponse](#)

## CreateUser(UserCreateModel)

```
[HttpPost]  
[Route("new")]  
public IActionResult CreateUser(UserCreateModel userModel)
```

### Parameters

userModel [UserCreateModel](#)

### Returns

[IActionResult](#)

## SetJwtCookie(HttpContext, string)

```
[NonAction]  
public void SetJwtCookie(HttpContext response, string token)
```

### Parameters

response [HttpContext](#)

token [string](#)

## SignInRequest(UserSignInModel)

```
[HttpPost]  
[Route("sign_in")]  
public Task<IActionResult> SignInRequest(UserSignInModel userModel)
```

## Parameters

userModel [UserSignInModel](#)

## Returns

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

## SignUserOut()

```
[HttpPost]  
[Authorize]  
[Route("sign_out")]  
public IActionResult SignUserOut()
```

## Returns

[IActionResult](#)

# Namespace ChessServer.Hubs

## Classes

[GameHub](#)

# Class GameHub

Namespace: [ChessServer.Hubs](#)

Assembly: ChessServer.dll

```
[Authorize]
public class GameHub : Hub<IGameHub>, IDisposable
```

## Inheritance

[object](#) ← [Hub](#) ← [Hub](#)<[IGameHub](#)> ← GameHub

## Implements

[IDisposable](#)

## Inherited Members

[Hub<IGameHub>.Clients](#) , [Hub.OnConnectedAsync\(\)](#) , [Hub.Dispose\(bool\)](#) , [Hub.Dispose\(\)](#) ,  
[Hub.Context](#) , [Hub.Groups](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Constructors

## GameHub(IGameManager, IChessDataService)

```
public GameHub(IGameManager gameManager, IChessDataService chessDataService)
```

## Parameters

gameManager [IGameManager](#)

chessDataService [IChessDataService](#)

# Methods

## JoinGame(string)

```
public Task JoinGame(string user)
```

Parameters

user [string](#)

Returns

[Task](#)

## LeaveGame(string)

```
public Task LeaveGame(string sessionId)
```

Parameters

sessionId [string](#)

Returns

[Task](#)

## MakeMove(int, string, MoveModel)

```
public Task MakeMove(int gameId, string sessionId, MoveModel move)
```

Parameters

gameId [int](#)

sessionId [string](#)

move [MoveModel](#)

Returns

[Task](#)

## OnDisconnectedAsync(Exception?)

Called when a connection with the hub is terminated.

```
public override Task OnDisconnectedAsync(Exception? ex)
```

Parameters

ex [Exception](#)

Returns

[Task](#)

A [Task](#) that represents the asynchronous disconnect.

## SendMessageToGroup(string, string)

```
public Task SendMessageToGroup(string message, string sessionId)
```

Parameters

message [string](#)

sessionId [string](#)

Returns

[Task](#)

## StopQueue()

```
public Task StopQueue()
```

Returns

## Task ↗

# Namespace DataLayer

## Classes

[ChessContext](#)

# Class ChessContext

Namespace: [DataLayer](#)

Assembly: DataLayer.dll

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

## Inheritance

[object](#) ↗ ← [DbContext](#) ↗ ← ChessContext

## Implements

[IInfrastructure](#) ↗ <[IServiceProvider](#) ↗>, [IDbContextDependencies](#) ↗, [IDbSetCache](#) ↗, [IDbContextPoolable](#) ↗, [IResettableService](#) ↗, [IDisposable](#) ↗, [IAsyncDisposable](#) ↗

## Inherited Members

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

## Constructors

### ChessContext(DbContextOptions<ChessContext>)

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

#### Parameters

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

## Properties

### ChessGames

```
public DbSet<ChessGame> ChessGames { get; set; }
```

#### Property Value

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

### Moves

```
public DbSet<Move> Moves { get; set; }
```

#### Property Value

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

### Users

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

## Property Value

[DbSet](#) <User>

## Methods

### OnModelCreating(ModelBuilder)

Override this method to further configure the model that was discovered by convention from the entity types exposed in [DbSet< TEntity >](#) properties on your derived context. The resulting model may be cached and re-used for subsequent instances of your derived context.

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The builder being used to construct the model for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure aspects of the model that are specific to a given database.

## Remarks

If a model is explicitly set on the options for this context (via [UseModel\(IModel\)](#)) then this method will not be run. However, it will still run when creating a compiled model.

See [Modeling entity types and relationships](#) for more information and examples.

# Namespace DataLayer.DataServices

## Classes

[ChessDataService](#)

[DataService](#)

[StockFishService](#)

## Interfaces

[IChessDataService](#)

[IDataService](#)

# Class ChessDataService

Namespace: [DataLayer.DataServices](#)

Assembly: DataLayer.dll

```
public class ChessDataService : IChessDataService
```

## Inheritance

[object](#) ← ChessDataService

## Implements

[IChessDataService](#)

## Inherited Members

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

## Constructors

### ChessDataService(ChessContext, IDataService)

```
public ChessDataService(ChessContext context, IDataService dataService)
```

## Parameters

context [ChessContext](#)

dataService [IDataService](#)

## Methods

### CreateBotGameAsync(string, bool)

```
public Task<(ChessGame, ChessInfo)> CreateBotGameAsync(string userName1, bool white)
```

## Parameters

`userName1` [string](#)

`white` [bool](#)

Returns

[Task](#) <([ChessGame](#), [ChessInfo](#))>

## CreateChessModel(ChessInfo, ChessGame, string)

`public` [ChessModel](#) [CreateChessModel](#)([ChessInfo](#) chessState, [ChessGame](#) game, [string](#) sessionId)

Parameters

`chessState` [ChessInfo](#)

`game` [ChessGame](#)

`sessionId` [string](#)

Returns

[ChessModel](#)

## CreateGameAsync(string, string)

`public` [Task](#)<([ChessGame](#), [ChessInfo](#))> [CreateGameAsync](#)([string](#) userName1, [string](#) userName2)

Parameters

`userName1` [string](#)

`userName2` [string](#)

Returns

[Task](#) <([ChessGame](#), [ChessInfo](#))>

## EndGame(int)

```
public ChessGame EndGame(int chessId)
```

Parameters

chessId [int](#)

Returns

[ChessGame](#)

## GetGameAsync(int)

```
public Task<ChessGame?> GetGameAsync(int chessId)
```

Parameters

chessId [int](#)

Returns

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

## GetGames()

```
public IList<ChessGame> GetGames()
```

Returns

[IList](#) <[ChessGame](#)>

## MoveAsync(int, string, string)

```
public Task<bool> MoveAsync(int chessId, string move, string FEN)
```

Parameters

**chessId** [int](#)

**move** [string](#)

**FEN** [string](#)

Returns

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

## RemoveLastMove(int)

```
public bool RemoveLastMove(int chessId)
```

Parameters

**chessId** [int](#)

Returns

[bool](#)

# Class DataService

Namespace: [DataLayer.DataServices](#)

Assembly: DataLayer.dll

```
public class DataService : IDataService
```

Inheritance

[object](#) ← DataService

Implements

[IDataService](#)

Inherited Members

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

## Constructors

### DataService(ChessContext)

```
public DataService(ChessContext context)
```

Parameters

context [ChessContext](#)

## Methods

### CreateUser(string, string)

```
public User? CreateUser(string username, string password)
```

Parameters

username [string](#)

`password` [string](#)

Returns

[User](#)

## GetUser(int)

```
public Task<User> GetUser(int id)
```

Parameters

`id` [int](#)

Returns

[Task](#) <User>

## GetUser(string)

```
public Task<User> GetUser(string username)
```

Parameters

`username` [string](#)

Returns

[Task](#) <User>

## GetUsers()

```
public IList<User> GetUsers()
```

Returns

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

## LogOut(string)

`public bool LogOut(string username)`

Parameters

`username string`

Returns

`bool`

## RemoveLastMove(int)

`public bool RemoveLastMove(int chessId)`

Parameters

`chessId int`

Returns

`bool`

## SignInUser(string, string)

`public bool SignInUser(string username, string password)`

Parameters

`username string`

`password string`

Returns

bool ↗

## StartSession()

```
public bool StartSession()
```

Returns

bool ↗

## StopSession()

```
public bool StopSession()
```

Returns

bool ↗

## UpdateCustomization()

```
public bool UpdateCustomization()
```

Returns

bool ↗

## UpdatePassword(string)

```
public bool UpdatePassword(string newPassword)
```

Parameters

`newPassword` [string](#)

Returns

[bool](#)

# Interface IChessDataService

Namespace: [DataLayer.DataServices](#)

Assembly: DataLayer.dll

```
public interface IChessDataService
```

## Methods

### CreateBotGameAsync(string, bool)

```
Task<(ChessGame, ChessInfo)> CreateBotGameAsync(string userName1, bool white)
```

Parameters

userName1 [string](#)

white [bool](#)

Returns

```
Task<(ChessGame, ChessInfo)>
```

### CreateChessModel(ChessInfo, ChessGame, string)

```
ChessModel CreateChessModel(ChessInfo chessState, ChessGame game, string sessionId)
```

Parameters

chessState [ChessInfo](#)

game [ChessGame](#)

sessionId [string](#)

Returns

[ChessModel](#)

## CreateGameAsync(string, string)

Task<(ChessGame, ChessInfo)> CreateGameAsync(**string** userName1, **string** userName2)

Parameters

userName1 [string](#)

userName2 [string](#)

Returns

[Task](#)<(ChessGame, ChessInfo)>

## EndGame(int)

ChessGame **EndGame**(**int** chessId)

Parameters

chessId [int](#)

Returns

[ChessGame](#)

## GetGameAsync(int)

Task<ChessGame?> GetGameAsync(**int** chessId)

Parameters

`chessId` [int](#)

Returns

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

## GetGames()

`IList<ChessGame> GetGames()`

Returns

[IList](#) <[ChessGame](#)>

## MoveAsync(int, string, string)

`Task<bool> MoveAsync(int chessId, string move, string FEN)`

Parameters

`chessId` [int](#)

`move` [string](#)

`FEN` [string](#)

Returns

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

## RemoveLastMove(int)

`bool RemoveLastMove(int chessId)`

Parameters

`chessId` [int](#)

Returns

[bool](#)

# Interface IDataService

Namespace: [DataLayer.DataServices](#)

Assembly: DataLayer.dll

```
public interface IDataService
```

## Methods

### CreateUser(string, string)

```
User? CreateUser(string username, string password)
```

Parameters

username [string](#)

password [string](#)

Returns

[User](#)

### GetUser(int)

```
Task<User> GetUser(int id)
```

Parameters

id [int](#)

Returns

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

## GetUser(string)

Task<User>  **GetUser**(string username)

Parameters

username [string](#)

Returns

[Task](#) <User>

## GetUsers()

IList<User>  **GetUsers**()

Returns

[IList](#) <User>

## LogOut(string)

bool  **LogOut**(string username)

Parameters

username [string](#)

Returns

[bool](#)

## SignInUser(string, string)

bool  **SignInUser**(string username, string password)

Parameters

`username string` ↗

`password string` ↗

Returns

`bool` ↗

## StartSession()

`bool StartSession()`

Returns

`bool` ↗

## StopSession()

`bool StopSession()`

Returns

`bool` ↗

## UpdateCustomization()

`bool UpdateCustomization()`

Returns

`bool` ↗

## UpdatePassword(string)

```
bool UpdatePassword(string newPassword)
```

### Parameters

newPassword [string](#)

### Returns

[bool](#)

# Class StockFishService

Namespace: [DataLayer.DataServices](#)

Assembly: DataLayer.dll

```
public class StockFishService : IStockFishService
```

## Inheritance

[object](#) ← StockFishService

## Implements

[IStockFishService](#)

## Inherited Members

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

## Constructors

StockFishService()

```
public StockFishService()
```

## Methods

MoveFrom(string)

```
public MoveModel MoveFrom(string FEN)
```

### Parameters

FEN [string](#)

### Returns

[MoveModel](#)

## StartNewStockFishGame()

```
public void StartNewStockFishGame()
```

# Namespace DataLayer.Entities.Chess

## Classes

[ChessGame](#)

[Move](#)

## Enums

[GameResult](#)

[GameType](#)

# Class ChessGame

Namespace: [DataLayer.Entities.Chess](#)

Assembly: DataLayer.dll

```
public class ChessGame
```

## Inheritance

[object](#) ← ChessGame

## Inherited Members

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

# Properties

## BlackId

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

### Property Value

[int](#)

## BlackPlayer

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

### Property Value

[User](#)

## CreatedAt

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

Property Value

[DateTime](#)

## GameType

```
public GameType GameType { get; set; }
```

Property Value

[GameType](#)

## Id

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

Property Value

[int](#)

## Moves

```
public List<Move> Moves { get; set; }
```

Property Value

[List](#)<[Move](#)>

## Result

```
public GameResult Result { get; set; }
```

Property Value

[GameResult](#)

WhiteId

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

Property Value

[int](#)

WhitePlayer

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

Property Value

[User](#)

# Enum GameResult

Namespace: [DataLayer.Entities.Chess](#)

Assembly: DataLayer.dll

```
public enum GameResult
```

## Fields

BlackWin = 1

Draw = 2

Ongoing = 3

WhiteWin = 0

# Enum GameType

Namespace: [DataLayer.Entities.Chess](#)

Assembly: DataLayer.dll

```
public enum GameType
```

## Fields

Bot = 0

Multiplayer = 1

# Class Move

Namespace: [DataLayer.Entities.Chess](#)

Assembly: DataLayer.dll

```
public class Move
```

Inheritance

[object](#) ← Move

Inherited Members

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

## Properties

ChessGame

```
public ChessGame ChessGame { get; set; }
```

Property Value

[ChessGame](#)

ChessGameId

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

Property Value

[int](#)

FEN

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

Property Value

[string](#) ↗

Id

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

Property Value

[int](#) ↗

MoveString

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

Property Value

[string](#) ↗

# Namespace DataLayer.Entities.Users

## Classes

[Customization](#)

[Session](#)

[User](#)

# Class Customization

Namespace: [DataLayer.Entities.Users](#)

Assembly: DataLayer.dll

```
public class Customization
```

## Inheritance

[object](#) ← Customization

## Inherited Members

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

# Properties

## boardPref

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

## Property Value

[string](#)

## customizationId

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

## Property Value

[int](#)

## darkMode

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

Property Value

[bool ↗](#)

piecePref

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

Property Value

[string ↗](#)

user

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

Property Value

[User](#)

userId

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

Property Value

[int ↗](#)

volume

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

Property Value

[int ↗](#)

# Class Session

Namespace: [DataLayer.Entities.Users](#)

Assembly: DataLayer.dll

```
public class Session
```

## Inheritance

[object](#) ← Session

## Inherited Members

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

# Properties

## createdAt

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

## Property Value

[DateTime](#)

## endedAt

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

## Property Value

[DateTime](#)?

## sessionId

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

Property Value

[int ↗](#)

user

```
public virtual User user { get; set; }
```

Property Value

[User](#)

userId

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

Property Value

[int ↗](#)

# Class User

Namespace: [DataLayer.Entities.Users](#)

Assembly: DataLayer.dll

```
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\(\)](#)

# Properties

## BlackGames

```
public List<ChessGame> BlackGames { get; set; }
```

## Property Value

[List](#) <[ChessGame](#)>

## Email

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

## Property Value

[string](#)

## Id

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

Property Value

[int ↗](#)

Password

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

Property Value

[string ↗](#)

Salt

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

Property Value

[string ↗](#)

Username

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

Property Value

[string ↗](#)

WhiteGames

```
public List<ChessGame> WhiteGames { get; set; }
```

Property Value

[List](#) <ChessGame>

# Namespace DataLayer.HelperMethods

## Classes

[ChessMethods](#)

[Hashing](#)

# Class ChessMethods

Namespace: [DataLayer.HelperMethods](#)

Assembly: DataLayer.dll

```
public static class ChessMethods
```

## Inheritance

[object](#) ← ChessMethods

## Inherited Members

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

# Methods

## ConvertMoveToColRow(string)

Converts a move string in the format "e2,e4" into row and column indices for from and to squares.

```
public static (int fromRow, int fromCol, int toRow, int toCol) ConvertMoveToColRow(string move)
```

### Parameters

move [string](#)

### Returns

[\(int fromRow, int fromCol, int toRow, int toCol\)](#)

## FindCheckBlockers(ChessInfo, King, Piece)

Finds squares that block a check from a sliding piece (queen, rook, bishop) to the king. Adds these blocking squares to the chessState.Blockers list.

```
public static void FindCheckBlockers(ChessInfo chessState, King king, Piece pieceChecked)
```

Parameters

chessState [ChessInfo](#)

king [King](#)

pieceChecked [Piece](#)

## GenerateFEN(ChessInfo)

Generates the FEN string representation of the current chess position from the chessState.

```
public static string GenerateFEN(ChessInfo chessState)
```

Parameters

chessState [ChessInfo](#)

Returns

[string](#) ↗

## MakeMove(ChessInfo, MoveModel)

Executes a move on the chessboard, updating the chessState accordingly. Handles promotions, en passant, castling, and updating FEN related state.

```
public static void MakeMove(ChessInfo chessState, MoveModel move)
```

Parameters

chessState [ChessInfo](#)

move [MoveModel](#)

## RankFileToRowCol(string)

Converts a chessboard coordinate in file-rank format (e.g., "e4") to row and column indexes.

```
public static (int, int) RankFileToRowCol(string fileRank)
```

Parameters

fileRank [string](#)

Returns

([int](#), [int](#))

A tuple with the corresponding row and col: (row, col).

## RowColToRankFile(int, int)

Converts row and column indexes to a chessboard coordinate in file-rank format (e.g., 0,0 -> "a1").

```
public static string RowColToRankFile(int row, int col)
```

Parameters

row [int](#)

col [int](#)

Returns

[string](#)

# Class Hashing

Namespace: [DataLayer.HelperMethods](#)

Assembly: DataLayer.dll

```
public class Hashing
```

## Inheritance

[object](#) ← Hashing

## Inherited Members

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

## Fields

### random

```
protected RandomNumberGenerator random
```

#### Field Value

[RandomNumberGenerator](#)

### saltBitSize

```
protected const int saltBitSize = 64
```

#### Field Value

[int](#)

### saltByteSize

```
protected const byte saltByteSize = 8
```

Field Value

[byte](#)

## Methods

### Hash(string)

```
public (string hash, string salt) Hash(string password)
```

Parameters

[password](#) [string](#)

Returns

[\(string](#) [hash](#), [string](#) [salt](#))

### Verify(string, string, string)

```
public bool Verify(string password, string storedHash, string storedSalt)
```

Parameters

[password](#) [string](#)

[storedHash](#) [string](#)

[storedSalt](#) [string](#)

Returns

[bool](#)

# Namespace DataLayer.HubServices

## Classes

[GameSession](#)

# Class GameSession

Namespace: [DataLayer.HubServices](#)

Assembly: DataLayer.dll

```
public class GameSession
```

## Inheritance

[object](#) ← GameSession

## Inherited Members

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

# Properties

## BlackPlayer

```
public string BlackPlayer { get; }
```

## Property Value

[string](#)

## Id

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

## Property Value

[string](#)

## IsReady

```
public bool IsReady { get; }
```

Property Value

[bool](#)

## Player1

```
public string Player1 { get; set; }
```

Property Value

[string](#)

## Player2

```
public string? Player2 { get; set; }
```

Property Value

[string](#)

## WhitePlayer

```
public string WhitePlayer { get; }
```

Property Value

[string](#)

## Methods

### Initialize()

```
public bool Initialize()
```

Returns

bool ↗

# Namespace DataLayer.IDataServices

## Interfaces

[IStockFishService](#)

# Interface IStockFishService

Namespace: [DataLayer.IDataServices](#)

Assembly: DataLayer.dll

```
public interface IStockFishService
```

## Methods

### MoveFrom(string)

MoveModel **MoveFrom**(**string** fEN)

Parameters

fEN [string](#)

Returns

[MoveModel](#)

### StartNewStockFishGame()

**void** **StartNewStockFishGame**()

# Namespace DataLayer.Migrations

## Classes

### [AddUniqueIndexToUsername](#)

A base class inherited by each EF Core migration.

### [ChessGameUpdate](#)

A base class inherited by each EF Core migration.

### [ChessGameUpdate1](#)

A base class inherited by each EF Core migration.

### [ChessgameAndMoves](#)

A base class inherited by each EF Core migration.

### [InitialCreate](#)

A base class inherited by each EF Core migration.

### [addedFENtomoves](#)

A base class inherited by each EF Core migration.

### [addedUserRelationsToChessGames](#)

A base class inherited by each EF Core migration.

### [added\\_moves](#)

A base class inherited by each EF Core migration.

### [added\\_salt](#)

A base class inherited by each EF Core migration.

### [email](#)

A base class inherited by each EF Core migration.

### [refreshdb](#)

A base class inherited by each EF Core migration.

### [removed\\_required\\_for\\_dev](#)

A base class inherited by each EF Core migration.

### [removed\\_users\\_from\\_chessgame](#)

A base class inherited by each EF Core migration.

### [trying\\_again](#)

A base class inherited by each EF Core migration.



# Class AddUniqueIndexToUsername

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250607184828_AddUniqueIndexToUsername")]
public class AddUniqueIndexToUsername : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← AddUniqueIndexToUsername

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class ChessGameUpdate

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250709100636_ChessGameUpdate")]
public class ChessGameUpdate : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← ChessGameUpdate

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class ChessGameUpdate1

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250709111712_ChessGameUpdate1")]
public class ChessGameUpdate1 : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← ChessGameUpdate1

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class ChessgameAndMoves

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250614215033_ChessgameAndMoves")]
public class ChessgameAndMoves : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← ChessgameAndMoves

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class InitialCreate

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250607180741_InitialCreate")]
public class InitialCreate : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← InitialCreate

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class addedFENtomoves

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250630160221_addedFENtomoves")]
public class addedFENtomoves : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← addedFENtomoves

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

[modelBuilder](#) [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class addedUserRelationsToChessGames

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250709113850_addedUserRelationsToChessGames")]
public class addedUserRelationsToChessGames : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← addedUserRelationsToChessGames

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class added\_moves

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250615205414_added_moves")]
public class added_moves : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← added\_moves

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class added\_salt

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250607232018_added_salt")]
public class added_salt : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← added\_salt

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

[modelBuilder](#) [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class email

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250610164552_email")]
public class email : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← email

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

[modelBuilder](#) [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class refreshdb

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250630160346_refreshdb")]
public class refreshdb : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← refreshdb

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class removed\_required\_for\_dev

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250615204429_removed_required_for_dev")]
public class removed_required_for_dev : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← removed\_required\_for\_dev

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class removed\_users\_from\_chessgame

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250615205326_removed_users_from_chessgame")]
public class removed_users_from_chessgame : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← removed\_users\_from\_chessgame

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Class trying\_again

Namespace: [DataLayer.Migrations](#)

Assembly: DataLayer.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ChessContext))]
[Migration("20250615205852_trying_again")]
public class trying_again : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← trying\_again

## Inherited Members

[Migration.InitialDatabase](#) , [Migration.TargetModel](#) , [Migration.UpOperations](#) ,  
[Migration.DownOperations](#) , [Migration.ActiveProvider](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from [Migration](#) if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See [Database migrations](#) for more information and examples.

## Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from [Migration](#).

See [Database migrations](#) for more information and examples.

# Namespace DataLayer.Models.Chess

## Classes

[ChessModel](#)

[CreateBotChessModel](#)

[CreateChessModel](#)

[MoveModel](#)

# Class ChessModel

Namespace: [DataLayer.Models.Chess](#)

Assembly: DataLayer.dll

```
public class ChessModel
```

## Inheritance

[object](#) ← ChessModel

## Inherited Members

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

# Properties

## BlockCheckPositions

```
public List<string> BlockCheckPositions { get; set; }
```

### Property Value

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

## Check

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

### Property Value

[bool](#)

## CheckMate

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

Property Value

[bool](#)

## Chessboard

```
public Piece[][] Chessboard { get; set; }
```

Property Value

[Piece](#)[][]

## CurrentPlayer

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

Property Value

[string](#)

## FEN

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

Property Value

[string](#)

## Id

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

Property Value

[int ↗](#)

IsWhite

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

Property Value

[bool ↗](#)

LastMove

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

Property Value

[string ↗](#)

Players

```
public string[] Players { get; set; }
```

Property Value

[string\[\] ↗](#)

SessionId

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

Property Value

[string](#) ↗

# Class CreateBotChessModel

Namespace: [DataLayer.Models.Chess](#)

Assembly: DataLayer.dll

```
public class CreateBotChessModel
```

## Inheritance

[object](#) ← CreateBotChessModel

## Inherited Members

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

# Properties

## PickedWhite

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

## Property Value

[bool](#)

## Player1

```
public string Player1 { get; set; }
```

## Property Value

[string](#)

# Class CreateChessModel

Namespace: [DataLayer.Models.Chess](#)

Assembly: DataLayer.dll

```
public class CreateChessModel
```

## Inheritance

[object](#) ← CreateChessModel

## Inherited Members

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

# Properties

## Player1

```
public string Player1 { get; set; }
```

### Property Value

[string](#)

## Player2

```
public string Player2 { get; set; }
```

### Property Value

[string](#)

# Class MoveModel

Namespace: [DataLayer.Models.Chess](#)

Assembly: DataLayer.dll

```
public class MoveModel
```

## Inheritance

[object](#) ← MoveModel

## Inherited Members

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

# Properties

## Move

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

### Property Value

[string](#)

## Promotion

```
public char? Promotion { get; set; }
```

### Property Value

[char](#)?

# Namespace DataLayer.Models.User Classes

[UserCreateModel](#)

[UserModel](#)

[UserSignInModel](#)

# Class UserCreateModel

Namespace: [DataLayer.Models.User](#)

Assembly: DataLayer.dll

```
public class UserCreateModel
```

## Inheritance

[object](#) ← UserCreateModel

## Inherited Members

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

# Properties

## Password

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

## Property Value

[string](#)

## Username

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

## Property Value

[string](#)

# Class UserModel

Namespace: [DataLayer.Models.User](#)

Assembly: DataLayer.dll

```
public class UserModel
```

## Inheritance

[object](#) ← UserModel

## Inherited Members

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

# Properties

## Id

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

### Property Value

[int](#)

## Username

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

### Property Value

[string](#)

# Class UserSignInModel

Namespace: [DataLayer.Models.User](#)

Assembly: DataLayer.dll

```
public class UserSignInModel
```

## Inheritance

[object](#) ← UserSignInModel

## Inherited Members

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

# Properties

## Password

[Required]  
public string Password { get; set; }

## Property Value

[string](#)

## Username

[Required]  
public string Username { get; set; }

## Property Value

[string](#)

# Namespace Testing

## Classes

[UserTests](#)

# Class UserTests

Namespace: [Testing](#)

Assembly: Testing.dll

```
public class UserTests
```

## Inheritance

[object](#) ← UserTests

## Inherited Members

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

# Methods

## TestTest()

[Fact]

```
public void TestTest()
```