Namespace Instaclause.Accountant.Api.Controllers

Classes

AnswersController

ContractsController

PartiesController

Class AnswersController

Namespace: Instaclause.Accountant.Api.Controllers

Assembly: Instaclause.Accountant.Api.dll

```
[ApiController]
[Route("[controller]/{contractId:int}")]
public class AnswersController : ControllerBase
```

Inheritance

object ← ControllerBase ← AnswersController

Inherited Members

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

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

ControllerBase.PhysicalFile(string, string, string) ,
ControllerBase.PhysicalFile(string, string, string, bool) ,
ControllerBase.PhysicalFile(string, string, DateTimeOffset?, EntityTagHeaderValue) ,
ControllerBase.PhysicalFile(string, string, DateTimeOffset?, EntityTagHeaderValue, bool) ,
ControllerBase.PhysicalFile(string, string, string, DateTimeOffset?, EntityTagHeaderValue) ,
ControllerBase.PhysicalFile(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool) ,
ControllerBase.Unauthorized() , ControllerBase.Unauthorized(object) ,
ControllerBase.NotFound() , ControllerBase.NotFound(object) , ControllerBase.BadRequest() ,
ControllerBase.BadRequest(object) , ControllerBase.BadRequest(ModelStateDictionary) ,
ControllerBase.UnprocessableEntity() , ControllerBase.UnprocessableEntity(object) ,
ControllerBase.UnprocessableEntity(ModelStateDictionary) , ControllerBase.Conflict() ,
ControllerBase.Conflict(object) , ControllerBase.Conflict(ModelStateDictionary) ,
ControllerBase.Problem(string, string, int?, string, string) ,
ControllerBase.ValidationProblem(ValidationProblemDetails) ,
ControllerBase.ValidationProblem(ModelStateDictionary) , ControllerBase.ValidationProblem() ,
ControllerBase.ValidationProblem(string, string, int?, string, string, ModelStateDictionary) ,
ControllerBase.Created() , ControllerBase.Created(string, object) ,
ControllerBase.Created(Uri, object) , ControllerBase.CreatedAtAction(string, object) ,
ControllerBase.CreatedAtAction(string, object, object) ,
ControllerBase.CreatedAtAction(string, string, object, object) ,
ControllerBase.CreatedAtRoute(string, object) , ControllerBase.CreatedAtRoute(object, object) ,
ControllerBase.CreatedAtRoute(string, object, object) , ControllerBase.Accepted() ,
ControllerBase.Accepted(object) , ControllerBase.Accepted(Uri) ,
ControllerBase.Accepted(string) , ControllerBase.Accepted(string, object) ,
ControllerBase.Accepted(Uri, object) , ControllerBase.AcceptedAtAction(string) ,
ControllerBase.AcceptedAtAction(string, string) ,
ControllerBase.AcceptedAtAction(string, object) ,
ControllerBase.AcceptedAtAction(string, string, object) ,
ControllerBase.AcceptedAtAction(string, object, object) ,
ControllerBase.AcceptedAtAction(string, string, object, object) ,
ControllerBase.AcceptedAtRoute(object) , ControllerBase.AcceptedAtRoute(string) ,
ControllerBase.AcceptedAtRoute(string, object) ,
ControllerBase.AcceptedAtRoute(object, object) ,
ControllerBase.AcceptedAtRoute(string, object, object) , ControllerBase.Challenge() ,
ControllerBase.Challenge(params string[]) ,
ControllerBase.Challenge(AuthenticationProperties) ,
ControllerBase.Challenge(AuthenticationProperties, params string[]) , ControllerBase.Forbid() ,
ControllerBase.Forbid(params string[]) , ControllerBase.Forbid(AuthenticationProperties) ,
ControllerBase.Forbid(AuthenticationProperties, params string[]) ,
ControllerBase.SignIn(ClaimsPrincipal) , ControllerBase.SignIn(ClaimsPrincipal, string) ,

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

Constructors

AnswersController(IInstaclauseContext)

```
public AnswersController(IInstaclauseContext context)
```

Parameters

`context` IInstaclauseContext

Methods

AnswerClauseField(int, int, string)

```
[HttpPost("field/{clauseFieldId:int}", Name = "AnswerClauseField")]
public Task<ActionResult> AnswerClauseField(int contractId, int clauseFieldId,
string value)
```

Parameters

`contractId` int

`clauseFieldId` int

`value` string

Returns

Task<ActionResult>

AnswerQuestion(int, int, string)

```
[HttpPost("question/{questionId:int}", Name = "AnswerQuestion")]
public Task<ActionResult> AnswerQuestion(int contractId, int questionId,
string answer)
```

Parameters

`contractId` int

`questionId` int

`answer` string

Returns

Task<ActionResult>

AssignCustomerParty(int, int, int)

```
[HttpPost("party/{modelPartyId:int}", Name = "AssignCustomerParty")]
public Task<ActionResult> AssignCustomerParty(int contractId, int modelPartyId,
int customerPartyId)
```

Parameters

`contractId` int

`modelPartyId` int

`customerPartyId` int

Returns

Task<ActionResult>

GetMissingAnswers(int)

```
[HttpGet("missing", Name = "GetMissingAnswers")]
public Task<ActionResult<MissingAnswersResponse>> GetMissingAnswers(int contractId)
```

Parameters

`contractId` int

Returns

Task<ActionResult<MissingAnswersResponse>>

Class ContractsController

Namespace: Instaclause.Accountant.Api.Controllers

Assembly: Instaclause.Accountant.Api.dll

```
[ApiController]
[Route("[controller]")]
public class ContractsController : ControllerBase
```

Inheritance

object ← ControllerBase ← ContractsController

Inherited Members

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

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

ControllerBase.PhysicalFile(string, string, string) ,
ControllerBase.PhysicalFile(string, string, string, bool) ,
ControllerBase.PhysicalFile(string, string, DateTimeOffset?, EntityTagHeaderValue) ,
ControllerBase.PhysicalFile(string, string, DateTimeOffset?, EntityTagHeaderValue, bool) ,
ControllerBase.PhysicalFile(string, string, string, DateTimeOffset?, EntityTagHeaderValue) ,
ControllerBase.PhysicalFile(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool) ,
ControllerBase.Unauthorized() , ControllerBase.Unauthorized(object) ,
ControllerBase.NotFound() , ControllerBase.NotFound(object) , ControllerBase.BadRequest() ,
ControllerBase.BadRequest(object) , ControllerBase.BadRequest(ModelStateDictionary) ,
ControllerBase.UnprocessableEntity() , ControllerBase.UnprocessableEntity(object) ,
ControllerBase.UnprocessableEntity(ModelStateDictionary) , ControllerBase.Conflict() ,
ControllerBase.Conflict(object) , ControllerBase.Conflict(ModelStateDictionary) ,
ControllerBase.Problem(string, string, int?, string, string) ,
ControllerBase.ValidationProblem(ValidationProblemDetails) ,
ControllerBase.ValidationProblem(ModelStateDictionary) , ControllerBase.ValidationProblem() ,
ControllerBase.ValidationProblem(string, string, int?, string, string, ModelStateDictionary) ,
ControllerBase.Created() , ControllerBase.Created(string, object) ,
ControllerBase.Created(Uri, object) , ControllerBase.CreatedAtAction(string, object) ,
ControllerBase.CreatedAtAction(string, object, object) ,
ControllerBase.CreatedAtAction(string, string, object, object) ,
ControllerBase.CreatedAtRoute(string, object) , ControllerBase.CreatedAtRoute(object, object) ,
ControllerBase.CreatedAtRoute(string, object, object) , ControllerBase.Accepted() ,
ControllerBase.Accepted(object) , ControllerBase.Accepted(Uri) ,
ControllerBase.Accepted(string) , ControllerBase.Accepted(string, object) ,
ControllerBase.Accepted(Uri, object) , ControllerBase.AcceptedAtAction(string) ,
ControllerBase.AcceptedAtAction(string, string) ,
ControllerBase.AcceptedAtAction(string, object) ,
ControllerBase.AcceptedAtAction(string, string, object) ,
ControllerBase.AcceptedAtAction(string, object, object) ,
ControllerBase.AcceptedAtAction(string, string, object, object) ,
ControllerBase.AcceptedAtRoute(object) , ControllerBase.AcceptedAtRoute(string) ,
ControllerBase.AcceptedAtRoute(string, object) ,
ControllerBase.AcceptedAtRoute(object, object) ,
ControllerBase.AcceptedAtRoute(string, object, object) , ControllerBase.Challenge() ,
ControllerBase.Challenge(params string[]) ,
ControllerBase.Challenge(AuthenticationProperties) ,
ControllerBase.Challenge(AuthenticationProperties, params string[]) , ControllerBase.Forbid() ,
ControllerBase.Forbid(params string[]) , ControllerBase.Forbid(AuthenticationProperties) ,
ControllerBase.Forbid(AuthenticationProperties, params string[]) ,
ControllerBase.SignIn(ClaimsPrincipal) , ControllerBase.SignIn(ClaimsPrincipal, string) ,

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

Constructors

ContractsController(IInstaclauseContext, ITemplateService)

```
public ContractsController(IInstaclauseContext context, ITemplateService
templateService)
```

Parameters

`context` IInstaclauseContext

`templateService` ITemplateService

Methods

CreateContract(CreateContractRequest)

```
[HttpPost(Name = "CreateContract")]
public Task<ActionResult<int>> CreateContract(CreateContractRequest request)
```

Parameters

`request` CreateContractRequest

Returns

Task<ActionResult<int>>

RenderContract(int, int)

```
[HttpGet("{contractId:int}/render/{documentId:int}", Name = "RenderContract")]
public Task<ActionResult<string>> RenderContract(int contractId, int documentId)
```

Parameters

`contractId` int

`documentId` int

Returns

Task<ActionResult<string>>

Class PartiesController

Namespace: Instaclause.Accountant.Api.Controllers

Assembly: Instaclause.Accountant.Api.dll

```
[ApiController]
[Route("[controller]")]
public class PartiesController : ControllerBase
```

Inheritance

object ← ControllerBase ← PartiesController

Inherited Members

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

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

ControllerBase.PhysicalFile(string, string, string) ,
ControllerBase.PhysicalFile(string, string, string, bool) ,
ControllerBase.PhysicalFile(string, string, DateTimeOffset?, EntityTagHeaderValue) ,
ControllerBase.PhysicalFile(string, string, DateTimeOffset?, EntityTagHeaderValue, bool) ,
ControllerBase.PhysicalFile(string, string, string, DateTimeOffset?, EntityTagHeaderValue) ,
ControllerBase.PhysicalFile(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool) ,
ControllerBase.Unauthorized() , ControllerBase.Unauthorized(object) ,
ControllerBase.NotFound() , ControllerBase.NotFound(object) , ControllerBase.BadRequest() ,
ControllerBase.BadRequest(object) , ControllerBase.BadRequest(ModelStateDictionary) ,
ControllerBase.UnprocessableEntity() , ControllerBase.UnprocessableEntity(object) ,
ControllerBase.UnprocessableEntity(ModelStateDictionary) , ControllerBase.Conflict() ,
ControllerBase.Conflict(object) , ControllerBase.Conflict(ModelStateDictionary) ,
ControllerBase.Problem(string, string, int?, string, string) ,
ControllerBase.ValidationProblem(ValidationProblemDetails) ,
ControllerBase.ValidationProblem(ModelStateDictionary) , ControllerBase.ValidationProblem() ,
ControllerBase.ValidationProblem(string, string, int?, string, string, ModelStateDictionary) ,
ControllerBase.Created() , ControllerBase.Created(string, object) ,
ControllerBase.Created(Uri, object) , ControllerBase.CreatedAtAction(string, object) ,
ControllerBase.CreatedAtAction(string, object, object) ,
ControllerBase.CreatedAtAction(string, string, object, object) ,
ControllerBase.CreatedAtRoute(string, object) , ControllerBase.CreatedAtRoute(object, object) ,
ControllerBase.CreatedAtRoute(string, object, object) , ControllerBase.Accepted() ,
ControllerBase.Accepted(object) , ControllerBase.Accepted(Uri) ,
ControllerBase.Accepted(string) , ControllerBase.Accepted(string, object) ,
ControllerBase.Accepted(Uri, object) , ControllerBase.AcceptedAtAction(string) ,
ControllerBase.AcceptedAtAction(string, string) ,
ControllerBase.AcceptedAtAction(string, object) ,
ControllerBase.AcceptedAtAction(string, string, object) ,
ControllerBase.AcceptedAtAction(string, object, object) ,
ControllerBase.AcceptedAtAction(string, string, object, object) ,
ControllerBase.AcceptedAtRoute(object) , ControllerBase.AcceptedAtRoute(string) ,
ControllerBase.AcceptedAtRoute(string, object) ,
ControllerBase.AcceptedAtRoute(object, object) ,
ControllerBase.AcceptedAtRoute(string, object, object) , ControllerBase.Challenge() ,
ControllerBase.Challenge(params string[]) ,
ControllerBase.Challenge(AuthenticationProperties) ,
ControllerBase.Challenge(AuthenticationProperties, params string[]) , ControllerBase.Forbid() ,
ControllerBase.Forbid(params string[]) , ControllerBase.Forbid(AuthenticationProperties) ,
ControllerBase.Forbid(AuthenticationProperties, params string[]) ,
ControllerBase.SignIn(ClaimsPrincipal) , ControllerBase.SignIn(ClaimsPrincipal, string) ,

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

Constructors

PartiesController(IInstaclauseContext)

```
public PartiesController(IInstaclauseContext context)
```

Parameters

`context` IInstaclauseContext

Methods

CreateParty(CreatePartyRequest)

```
[HttpPost(Name = "CreateParty")]
public Task<ActionResult<int>> CreateParty(CreatePartyRequest request)
```

Parameters

`request` CreatePartyRequest

Returns

Task<ActionResult<int>>

CreatePartyField(int, CreatePartyFieldRequest)

```
[HttpPost("{customerPartyId:int}/fields", Name = "CreatePartyField")]
public Task<ActionResult<int>> CreatePartyField(int customerPartyId,
CreatePartyFieldRequest request)
```

Parameters

`customerPartyId` int

`request` CreatePartyFieldRequest

Returns

Task<ActionResult<int>>

Namespace Instaclause.Accountant.Api.Models.Contracts

Classes

CreateContractRequest

CreatePartyFieldRequest

CreatePartyRequest

MissingAnswersResponse

Class CreateContractRequest

Namespace: Instaclause.Accountant.Api.Models.Contracts

Assembly: Instaclause.Accountant.Api.dll

```
public record CreateContractRequest : IEquatable<CreateContractRequest>
```

Inheritance

object ← CreateContractRequest

Implements

IEquatable<CreateContractRequest>

Inherited Members

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

Constructors

CreateContractRequest(string, int)

```
public CreateContractRequest(string Name, int CustomerId)
```

Parameters

`Name` string

`CustomerId` int

Properties

CustomerId

```
public int CustomerId { get; init; }
```

Property Value

int

Name

```
public string Name { get; init; }
```

Property Value

string

Class CreatePartyFieldRequest

Namespace: Instaclause.Accountant.Api.Models.Contracts

Assembly: Instaclause.Accountant.Api.dll

```
public class CreatePartyFieldRequest
```

Inheritance

object ← CreatePartyFieldRequest

Inherited Members

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

Properties

FieldId

```
[Required]
public required int FieldId { get; set; }
```

Property Value

int

Value

```
[Required]
public required string Value { get; set; }
```

Property Value

string

Class CreatePartyRequest

Namespace: Instaclause.Accountant.Api.Models.Contracts

Assembly: Instaclause.Accountant.Api.dll

```
public class CreatePartyRequest
```

Inheritance

object ← CreatePartyRequest

Inherited Members

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

Properties

CustomerId

```
public required int CustomerId { get; set; }
```

Property Value

int

Fields

```
public List<CreatePartyFieldRequest>? Fields { get; set; }
```

Property Value

List<CreatePartyFieldRequest>

Name

```
public required string Name { get; set; }
```

Property Value

string

TypeId

```
public required int TypeId { get; set; }
```

Property Value

int

Class MissingAnswersResponse

Namespace: Instaclause.Accountant.Api.Models.Contracts

Assembly: Instaclause.Accountant.Api.dll

```
public class MissingAnswersResponse
```

Inheritance

object ← MissingAnswersResponse

Inherited Members

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

Properties

ClauseFields

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

Property Value

List<string>

PartyAssignments

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

Property Value

List<string>

PartyFields

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

Property Value

List<string>

Questions

```
public List<int> Questions { get; set; }
```

Property Value

List<int>

Namespace Instaclause.Accountant.Api.Tests.Controllers

Classes

ContractsControllerTest

Class ContractsControllerTest

Namespace: Instaclause.Accountant.Api.Tests.Controllers

Assembly: Instaclause.Accountant.Api.Tests.dll

```
[TestClass]
[TestSubject(typeof(ContractsController))]
public class ContractsControllerTest
```

Inheritance

object ← ContractsControllerTest

Inherited Members

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

Methods

ContractRenderTest()

```
[TestMethod]
public Task ContractRenderTest()
```

Returns

Task

Namespace Instaclause.Domain

Classes

StampedEntity

Serves as a base class for entities within the system, providing common properties for tracking the creation, update, and deletion timestamps of records, enhancing data management and auditing capabilities.

TranslatableEntity<T>

Serves as a base class for entities that support translation, providing properties and relationships necessary for managing translations of content within the system, enhancing multilingual support and localization.

Class StampedEntity

Namespace: Instaclause.Domain

Assembly: Instaclause.Domain.dll

Serves as a base class for entities within the system, providing common properties for tracking the creation, update, and deletion timestamps of records, enhancing data management and auditing capabilities.

```
public abstract class StampedEntity
```

Inheritance

object ← StampedEntity

Derived

ContractClauseFieldValue, ContractQuestionValue

Inherited Members

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

Properties

CreatedAt

Gets the date and time when the entity was created, providing a timestamp for record creation.

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

Property Value

DateTime

DeletedAt

Gets the date and time when the entity was marked as deleted, allowing for soft deletion mechanisms where records are retained but flagged as inactive.

```
public DateTime DeletedAt { get; init; }
```

Property Value

DateTime

UpdatedAt

Gets the date and time when the entity was last updated, providing a timestamp for the most recent modification.

```
public DateTime UpdatedAt { get; init; }
```

Property Value

DateTime

Class TranslatableEntity<T>

Namespace: Instaclause.Domain

Assembly: Instaclause.Domain.dll

Serves as a base class for entities that support translation, providing properties and relationships necessary for managing translations of content within the system, enhancing multilingual support and localization.

```
public abstract class TranslatableEntity<T>
```

Type Parameters

T

The type of the translatable entity.

Inheritance

object ← TranslatableEntity<T>

Derived

Clause, ClauseGroup, Snippet, ContractModel, Document, Field, FieldGroup, FieldOption, FieldType, FieldTypeUnit, PartyType, Question

Inherited Members

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

Properties

BaseLanguageEntity

Gets the base language entity, linking to the original entity from which this translation was derived.

```
public T? BaseLanguageEntity { get; init; }
```

Property Value

T

BaseLanguageEntityId

Gets the identifier of the base language entity, used when this instance is a translation of another entity.

```
public int? BaseLanguageEntityId { get; init; }
```

Property Value

int?

Language

Gets the language of this entity, providing details on the language used for this instance.

```
public Language? Language { get; init; }
```

Property Value

Language

LanguageId

Gets the language identifier, indicating the language of this entity or translation.

```
public int LanguageId { get; init; }
```

Property Value

int

TranslationStatus

Gets the translation status, linking to the status details to indicate the review and approval state.

```
public TranslationStatus? TranslationStatus { get; init; }
```

Property Value

TranslationStatus

TranslationStatusId

Gets the translation status identifier, reflecting the current status of the translation in terms of approval and review.

```
public int TranslationStatusId { get; init; }
```

Property Value

int

Translations

Gets the collection of translations of this entity, facilitating management of different language versions of the content.

```
public List<T>? Translations { get; init; }
```

Property Value

List<T>

Namespace Instaclause.Domain.Clauses

Classes

Clause

Represents a reusable segment of text or conditions that form part of a document. The textual data of a clause is stored as a template, which can be rendered using the Scriban syntax. The template data can include variables, conditions, and loops to generate dynamic content. The variables within a clause are represented by clause fields.

ClauseField

Represents a specific field within a clause, detailing the data requirements that part of the clause. A field can have a condition under which it is applicable.

ClauseGroup

Represents a group of clauses, allowing for hierarchical organization of clauses within the system. Used by the front-end application to organize clauses into a library.

Snippet

Represents a reusable snippet of template or code that can be used within clauses. Used for fields that are common across multiple clauses or for defining reusable templates.

Class Clause

Namespace: Instaclause.Domain.Clauses

Assembly: Instaclause.Domain.dll

Represents a reusable segment of text or conditions that form part of a document. The textual data of a clause is stored as a template, which can be rendered using the Scriban syntax. The template data can include variables, conditions, and loops to generate dynamic content. The variables within a clause are represented by clause fields.

```
public class Clause : TranslatableEntity<Clause>
```

Inheritance

object ← TranslatableEntity<Clause> ← Clause

Inherited Members

TranslatableEntity<Clause>.BaseLanguageEntityId , TranslatableEntity<Clause>.BaseLanguageEntity , TranslatableEntity<Clause>.LanguageId , TranslatableEntity<Clause>.TranslationStatusId , TranslatableEntity<Clause>.Language , TranslatableEntity<Clause>.TranslationStatus , TranslatableEntity<Clause>.Translations , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Documents

Gets the collection of document clauses that are associated with this clause. This represents the many-to-many relationship between documents and clauses.

```
public List<DocumentClause>? Documents { get; init; }
```

Property Value

List<DocumentClause>

Fields

Gets the collection of clause fields associated with this clause. Clause fields represent the data elements or variables within a clause.

```
public List<ClauseField>? Fields { get; init; }
```

Property Value

List<ClauseField>

Group

Gets the group associated with this clause. This relationship allows clauses to be organized into hierarchical groups.

```
public ClauseGroup? Group { get; init; }
```

Property Value

ClauseGroup

GroupId

Gets the identifier of the group to which this clause belongs. Only used for clauses that would be part of clauses library.

```
public int? GroupId { get; init; }
```

Property Value

int?

Id

Gets the identifier for the clause.

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

Property Value

int

Name

Gets the name of the clause.

```
public string? Name { get; init; }
```

Property Value

string

ReferenceClause

Gets the reference clause entity. This is used for clauses that are based on or linked to another clause.

```
public Clause? ReferenceClause { get; init; }
```

Property Value

Clause

ReferenceClauseId

Gets the identifier of a reference clause. This property can be used to link clauses together.

```
public int? ReferenceClauseId { get; init; }
```

Property Value

int?

ReferencedClauses

Gets the collection of clauses that reference this clause. This allows for the establishment of relationships between clauses.

```
public List<Clause>? ReferencedClauses { get; init; }
```

Property Value

List<Clause>

TemplateData

Gets the template data for the clause. The template data uses the Scriban syntax.

```
public required string TemplateData { get; init; }
```

Property Value

string

Class ClauseField

Namespace: Instaclause.Domain.Clauses

Assembly: Instaclause.Domain.dll

Represents a specific field within a clause, detailing the data requirements that part of the clause. A field can have a condition under which it is applicable.

```
public class ClauseField
```

Inheritance

object ← ClauseField

Inherited Members

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

Properties

Clause

Gets the clause associated with this field.

```
public Clause? Clause { get; init; }
```

Property Value

Clause

ClauseId

Gets the identifier of the clause to which this field belongs.

```
public int ClauseId { get; init; }
```

Property Value

int

Condition

Gets the condition under which this field is applicable or required.

```
public string? Condition { get; init; }
```

Property Value

string

Field

Gets the field associated with this clause field.

```
public Field? Field { get; init; }
```

Property Value

Field

FieldId

Gets the field identifier, linking to the field definition.

```
public int FieldId { get; init; }
```

Property Value

int

FieldTypeUnit

Gets the field type unit associated with this clause field.

```
public FieldTypeUnit? FieldTypeUnit { get; init; }
```

Property Value

FieldTypeUnit

FieldTypeUnitId

Gets the field type unit identifier, specifying the unit or type of data expected for this field.

```
public int? FieldTypeUnitId { get; init; }
```

Property Value

int?

Id

Gets the identifier for the clause field.

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

Property Value

int

Identifier

Gets the identifier used to uniquely identify the field within the contract model.

```
public required string Identifier { get; init; }
```

Property Value

string

Required

Gets a value indicating whether this field is required within the clause.

```
public bool? Required { get; init; }
```

Property Value

bool?

Class ClauseGroup

Namespace: Instaclause.Domain.Clauses

Assembly: Instaclause.Domain.dll

Represents a group of clauses, allowing for hierarchical organization of clauses within the system. Used by the front-end application to organize clauses into a library.

```
public class ClauseGroup : TranslatableEntity<ClauseGroup>
```

Inheritance

object ← TranslatableEntity<ClauseGroup> ← ClauseGroup

Inherited Members

TranslatableEntity<ClauseGroup>.BaseLanguageEntityId ,
TranslatableEntity<ClauseGroup>.BaseLanguageEntity ,
TranslatableEntity<ClauseGroup>.LanguageId ,
TranslatableEntity<ClauseGroup>.TranslationStatusId ,
TranslatableEntity<ClauseGroup>.Language , TranslatableEntity<ClauseGroup>.TranslationStatus ,
TranslatableEntity<ClauseGroup>.Translations , object.Equals(object) ,
object.Equals(object, object) , object.GetHashCode() , object.GetType() ,
object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

ChildGroups

Gets the collection of child groups under this clause group.

```
public List<ClauseGroup>? ChildGroups { get; init; }
```

Property Value

List<ClauseGroup>

Clauses

Gets the collection of clauses within this group.

```
public List<Clause>? Clauses { get; init; }
```

Property Value

List<Clause>

Id

Gets the identifier for the clause group.

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

Property Value

int

Name

Gets the name of the clause group.

```
public required string Name { get; init; }
```

Property Value

string

Parent

Gets the parent group of this clause group.

```
public ClauseGroup? Parent { get; init; }
```

Property Value

ClauseGroup

ParentId

Gets the identifier of the parent group, if this is a subgroup.

```
public int? ParentId { get; init; }
```

Property Value

int?

Class Snippet

Namespace: Instaclause.Domain.Clauses

Assembly: Instaclause.Domain.dll

Represents a reusable snippet of template or code that can be used within clauses. Used for fields that are common across multiple clauses or for defining reusable templates.

```
public class Snippet : TranslatableEntity<Snippet>
```

Inheritance

object ← TranslatableEntity<Snippet> ← Snippet

Inherited Members

TranslatableEntity<Snippet>.BaseLanguageEntityId , TranslatableEntity<Snippet>.BaseLanguageEntity , TranslatableEntity<Snippet>.LanguageId , TranslatableEntity<Snippet>.TranslationStatusId , TranslatableEntity<Snippet>.Language , TranslatableEntity<Snippet>.TranslationStatus , TranslatableEntity<Snippet>.Translations , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Data

Gets the data of the snippet, which may include text, code, or template markup.

```
public required string Data { get; init; }
```

Property Value

string

Description

Gets the description of the snippet, explaining its purpose or use case.

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

Property Value

string

Id

Gets the identifier for the snippet.

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

Property Value

int

Identifier

Gets the unique identifier used to reference the snippet within templates.

```
public required string Identifier { get; init; }
```

Property Value

string

Namespace Instaclause.Domain.Contract Models

Classes

ContractModel

Represents the model for a contract, defining the structure and components that make up a contract type. Is what is versioned. It would be equivalent of the categories + versions of the old system.

ContractModelGroup

Represents a group of contract models, allowing for organizational structuring and categorization.

ContractModelParty

Represents a party within a contract model, defining the roles and parties involved in the contract. Is possible to define a template for each type of party and what types are allowed. The template is used when the meta field _template is used in a clause template data. If the template is not set the default template, defined in PartyType, is used.

ContractModelPartyAdditionalField

Represents additional fields associated with a party in a contract model, allowing for the customization and extension of party data requirements.

Class ContractModel

Namespace: Instaclause.Domain.ContractModels

Assembly: Instaclause.Domain.dll

Represents the model for a contract, defining the structure and components that make up a contract type. Is what is versioned. It would be equivalent of the categories + versions of the old system.

```
public class ContractModel : TranslatableEntity<ContractModel>
```

Inheritance

object ← TranslatableEntity<ContractModel> ← ContractModel

Inherited Members

TranslatableEntity<ContractModel>.BaseLanguageEntityId ,
TranslatableEntity<ContractModel>.BaseLanguageEntity ,
TranslatableEntity<ContractModel>.LanguageId ,
TranslatableEntity<ContractModel>.TranslationStatusId ,
TranslatableEntity<ContractModel>.Language ,
TranslatableEntity<ContractModel>.TranslationStatus ,
TranslatableEntity<ContractModel>.Translations , object.Equals(object) ,
object.Equals(object, object) , object.GetHashCode() , object.GetType() ,
object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Contracts

Gets the collection of contracts instantiated from this model.

```
public List<Contract>? Contracts { get; init; }
```

Property Value

List<Contract>

Documents

Gets the collection of documents associated with this contract model.

```
public List<Document>? Documents { get; init; }
```

Property Value

List<Document>

Group

Gets the group associated with this contract model.

```
public ContractModelGroup? Group { get; init; }
```

Property Value

ContractModelGroup

GroupId

Gets the group identifier to which this contract model belongs.

```
public int? GroupId { get; init; }
```

Property Value

int?

Id

Gets the identifier for the contract model.

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

Property Value

int

Jurisdiction

Gets the jurisdiction under which this contract model is applicable.

```
public Jurisdiction? Jurisdiction { get; init; }
```

Property Value

Jurisdiction

JurisdictionId

Gets the jurisdiction identifier, indicating the legal jurisdiction under which the contract model is applicable.

```
public required int JurisdictionId { get; init; }
```

Property Value

int

Name

Gets the name of the contract model.

```
public required string Name { get; init; }
```

Property Value

string

Parties

Gets the collection of parties associated with this contract model.

```
public List<ContractModelParty>? Parties { get; init; }
```

Property Value

List<ContractModelParty>

Questions

Gets the collection of questions associated with this contract model.

```
public List<Question>? Questions { get; init; }
```

Property Value

List<Question>

Version

Gets the version number of the contract model.

```
public int Version { get; init; }
```

Property Value

int

Class ContractModelGroup

Namespace: Instaclause.Domain.ContractModels

Assembly: Instaclause.Domain.dll

Represents a group of contract models, allowing for organizational structuring and categorization.

```
public class ContractModelGroup
```

Inheritance

object ← ContractModelGroup

Inherited Members

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

Properties

ChildGroups

Gets the collection of child groups under this contract model group.

```
public List<ContractModelGroup>? ChildGroups { get; init; }
```

Property Value

List<ContractModelGroup>

Id

Gets the identifier for the contract model group.

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

Property Value

int

Name

Gets the name of the contract model group.

```
public required string Name { get; init; }
```

Property Value

string

Parent

Gets the parent group of this contract model group.

```
public ContractModelGroup? Parent { get; init; }
```

Property Value

ContractModelGroup

ParentId

Gets the parent group identifier, allowing for hierarchical structuring.

```
public int? ParentId { get; init; }
```

Property Value

int?

Class ContractModelParty

Namespace: Instaclause.Domain.ContractModels

Assembly: Instaclause.Domain.dll

Represents a party within a contract model, defining the roles and parties involved in the contract. Is possible to define a template for each type of party and what types are allowed. The template is used when the meta field _template is used in a clause template data. If the template is not set the default template, defined in PartyType, is used.

```
public class ContractModelParty
```

Inheritance

object ← ContractModelParty

Inherited Members

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

Properties

AdditionalFields

Gets the collection of additional fields associated with this party.

```
public List<ContractModelPartyAdditionalField>? AdditionalFields { get; init; }
```

Property Value

List<ContractModelPartyAdditionalField>

AllowedPartyTypeIds

Gets the allowed party type identifiers, defining the types of entities that can fulfill the role of this party.

```
public required int[] AllowedPartyTypeIds { get; init; }
```

Property Value

int[]

Children

Gets the collection of child parties under this contract model party.

```
public List<ContractModelParty>? Children { get; init; }
```

Property Value

List<ContractModelParty>

CompanyTemplateData

Gets the template data for a company associated with this party.

```
public string? CompanyTemplateData { get; init; }
```

Property Value

string

Condition

Gets the condition under which this party is applicable or required.

```
public string? Condition { get; init; }
```

Property Value

string

ContractModel

Gets the contract model to which this party belongs.

```
public ContractModel? ContractModel { get; init; }
```

Property Value

ContractModel

ContractModelId

Gets the identifier of the contract model to which this party belongs.

```
public int ContractModelId { get; init; }
```

Property Value

int

Id

Gets the identifier for the contract model party.

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

Property Value

int

Identifier

Gets the identifier used to uniquely identify the party within a contract model.

```
public required string Identifier { get; init; }
```

Property Value

string

Optional

Gets a value indicating whether this party is optional within the contract model.

```
public bool Optional { get; init; }
```

Property Value

bool

ParentParty

Gets the parent party of this contract model party.

```
public ContractModelParty? ParentParty { get; init; }
```

Property Value

ContractModelParty

ParentPartyId

Gets the identifier for the parent party, allowing for hierarchical structuring of party relationships.

```
public int? ParentPartyId { get; init; }
```

Property Value

int?

PersonTemplateData

Gets the template data for a person associated with this party.

```
public string? PersonTemplateData { get; init; }
```

Property Value

string

Title

Gets the title of the party, such as "Seller" or "Buyer".

```
public required string Title { get; init; }
```

Property Value

string

Class ContractModelPartyAdditionalField

Namespace: Instaclause.Domain.ContractModels

Assembly: Instaclause.Domain.dll

Represents additional fields associated with a party in a contract model, allowing for the customization and extension of party data requirements.

```
public class ContractModelPartyAdditionalField
```

Inheritance

object ← ContractModelPartyAdditionalField

Inherited Members

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

Properties

Field

Gets the field definition associated with this additional field.

```
public Field? Field { get; init; }
```

Property Value

Field

FieldId

Gets the field identifier, linking to the field definition.

```
public int FieldId { get; init; }
```

Property Value

int

FieldTypeUnit

Gets the field type unit associated with this additional field.

```
public FieldTypeUnit? FieldTypeUnit { get; init; }
```

Property Value

FieldTypeUnit

FieldTypeUnitId

Gets the field type unit identifier, specifying the unit or type of data expected for this field.

```
public int? FieldTypeUnitId { get; init; }
```

Property Value

int?

Id

Gets the identifier for the additional field.

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

Property Value

int

ModelPartyId

Gets the identifier of the model party to which this additional field belongs.

```
public int ModelPartyId { get; init; }
```

Property Value

int

Party

Gets the party associated with this additional field.

```
public ContractModelParty? Party { get; init; }
```

Property Value

ContractModelParty

PartyType

Gets the party type associated with this field.

```
public PartyType? PartyType { get; init; }
```

Property Value

PartyType

PartyTypeId

Gets the party type identifier, indicating the type of party this field is relevant to.

```
public int PartyTypeId { get; init; }
```

Property Value

int

Required

Gets a value indicating whether this field is required for the associated party.

```
public bool Required { get; init; }
```

Property Value

bool

ValidationRegex

Gets the regular expression used for validating the field value.

```
public string? ValidationRegex { get; init; }
```

Property Value

string

Namespace Instaclause.Domain.Contracts

Classes

Contract

Represents a specific instance of a contract. Is created from a contract model, and contains all the information and values specific to an individual contract.

ContractArtifact

Represents an artifact related to a contract, such as a generated document or file, providing a way to store and reference these items within the system.

ContractClauseFieldValue

Represents the value assigned to a specific field within a clause for a given contract, capturing the data input or selection made during contract customization.

ContractDocumentRender

Represents a rendered version of a document associated with a contract, which might include the final formatted document ready for review or signatures.

ContractPartyValue

Represents the actual assignment of a customer's party to a party required by a contract model.

ContractQuestionValue

Represents the answer or value provided for a specific question within a contract, capturing user input or selections related to the contract's configuration.

Class Contract

Namespace: Instaclause.Domain.Contracts

Assembly: Instaclause.Domain.dll

Represents a specific instance of a contract. Is created from a contract model, and contains all the information and values specific to an individual contract.

```
public class Contract
```

Inheritance

object ← Contract

Inherited Members

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

Properties

Artifacts

Gets the collection of artifacts associated with this contract, such as generated documents or files.

```
public List<ContractArtifact>? Artifacts { get; init; }
```

Property Value

List<ContractArtifact>

ClausesFieldValues

Gets the collection of clause field values associated with this contract, representing the data input for fields defined in the contract's clauses.

```
public List<ContractClauseFieldValue>? ClausesFieldValues { get; init; }
```

Property Value

List<ContractClauseFieldValue>

ContractModel

Gets the contract model from which this contract is instantiated.

```
public ContractModel? ContractModel { get; init; }
```

Property Value

ContractModel

ContractModelId

Gets the identifier of the contract model from which this contract is instantiated.

```
public int ContractModelId { get; init; }
```

Property Value

int

Customer

Gets the customer associated with this contract.

```
public Customer? Customer { get; init; }
```

Property Value

Customer

CustomerId

Gets the identifier of the customer associated with this contract.

```
public int CustomerId { get; init; }
```

Property Value

int

Id

Gets the identifier for the contract.

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

Property Value

int

Name

Gets the name of the contract, often used for display or identification purposes.

```
public required string Name { get; init; }
```

Property Value

string

Parties

Gets the collection of party values associated with this contract, representing the parties involved in the contract as defined by the contract model.

```
public List<ContractPartyValue>? Parties { get; init; }
```

Property Value

List<ContractPartyValue>

QuestionsValues

Gets the collection of question values associated with this contract, representing the answers to questions defined in the contract model.

```
public List<ContractQuestionValue>? QuestionsValues { get; init; }
```

Property Value

List<ContractQuestionValue>

Renders

Gets the collection of document renders associated with this contract, representing rendered versions of the contract's documents.

```
public List<ContractDocumentRender>? Renders { get; init; }
```

Property Value

List<ContractDocumentRender>

Class ContractArtifact

Namespace: Instaclause.Domain.Contracts

Assembly: Instaclause.Domain.dll

Represents an artifact related to a contract, such as a generated document or file, providing a way to store and reference these items within the system.

```
public class ContractArtifact
```

Inheritance

object ← ContractArtifact

Inherited Members

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

Properties

Contract

Gets the contract to which this artifact is associated.

```
public Contract? Contract { get; init; }
```

Property Value

Contract

ContractId

Gets the identifier of the contract to which this artifact is associated.

```
public int ContractId { get; init; }
```

Property Value

int

Document

Gets the document associated with this artifact, if any.

```
public Document? Document { get; init; }
```

Property Value

Document

DocumentId

Gets the identifier of the document associated with this artifact, if applicable.

```
public int DocumentId { get; init; }
```

Property Value

int

Id

Gets the identifier for the contract artifact.

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

Property Value

int

Language

Gets the language in which this artifact is presented.

```
public Language? Language { get; init; }
```

Property Value

Language

LanguageId

Gets the language identifier for this artifact, indicating the language in which the artifact is presented.

```
public int LanguageId { get; init; }
```

Property Value

int

Type

Gets the type of the artifact, which could describe its format or purpose (e.g., PDF document, signature file).

```
public required string Type { get; init; }
```

Property Value

string

Url

Gets the URL where the artifact is stored or can be accessed.

```
public required string Url { get; init; }
```

Property Value

string

Class ContractClauseFieldValue

Namespace: Instaclause.Domain.Contracts

Assembly: Instaclause.Domain.dll

Represents the value assigned to a specific field within a clause for a given contract, capturing the data input or selection made during contract customization.

```
public class ContractClauseFieldValue : StampedEntity
```

Inheritance

object ← StampedEntity ← ContractClauseFieldValue

Inherited Members

StampedEntity.CreatedAt , StampedEntity.UpdatedAt , StampedEntity.DeletedAt , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

ClauseField

Gets the clause field for which this value is assigned.

```
public ClauseField? ClauseField { get; init; }
```

Property Value

ClauseField

ClauseFieldId

Gets the identifier of the clause field for which this value is assigned.

```
public int ClauseFieldId { get; init; }
```

Property Value

int

Contract

Gets the contract to which this clause field value belongs.

```
public Contract? Contract { get; init; }
```

Property Value

Contract

ContractId

Gets the identifier of the contract to which this field value belongs.

```
public int ContractId { get; init; }
```

Property Value

int

Value

Gets the value for the clause field, which may be text, a selection identifier, or other data as appropriate.

```
public required string Value { get; init; }
```

Property Value

string

Class ContractDocumentRender

Namespace: Instaclause.Domain.Contracts

Assembly: Instaclause.Domain.dll

Represents a rendered version of a document associated with a contract, which might include the final formatted document ready for review or signatures.

```
public class ContractDocumentRender
```

Inheritance

object ← ContractDocumentRender

Inherited Members

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

Properties

Contract

Gets the contract to which this document render is associated.

```
public Contract? Contract { get; init; }
```

Property Value

Contract

ContractId

Gets the identifier of the contract to which this document render belongs.

```
public int ContractId { get; init; }
```

Property Value

int

Data

Gets data rendered by the Scriban template engine, with all the variables replaced by their values.

```
public required string Data { get; init; }
```

Property Value

string

Document

Gets the original document from which this render was created.

```
public Document? Document { get; init; }
```

Property Value

Document

DocumentId

Gets the identifier of the document that has been rendered.

```
public int DocumentId { get; init; }
```

Property Value

int

Class ContractPartyValue

Namespace: Instaclause.Domain.Contracts

Assembly: Instaclause.Domain.dll

Represents the actual assignment of a customer's party to a party required by a contract model.

```
public class ContractPartyValue
```

Inheritance

object ← ContractPartyValue

Inherited Members

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

Properties

Contract

Gets the contract to which this party value is associated.

```
public Contract? Contract { get; init; }
```

Property Value

Contract

ContractId

Gets the identifier of the contract to which this party value belongs.

```
public int ContractId { get; init; }
```

Property Value

int

ContractModelParty

Gets the contract model party definition that this value corresponds to.

```
public ContractModelParty? ContractModelParty { get; init; }
```

Property Value

ContractModelParty

ContractModelPartyId

Gets the identifier of the contract model party definition that this value corresponds to.

```
public int ContractModelPartyId { get; init; }
```

Property Value

int

CustomerParty

Gets the customer party, indicating the actual party involved in the contract.

```
public CustomerParty? CustomerParty { get; init; }
```

Property Value

CustomerParty

CustomerPartyId

Gets the identifier of the customer party, indicating the actual party involved in the contract as specified by the customer.

```
public int CustomerPartyId { get; init; }
```

Property Value

int

Id

Gets the identifier for this contract party value.

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

Property Value

int

Class ContractQuestionValue

Namespace: Instaclause.Domain.Contracts

Assembly: Instaclause.Domain.dll

Represents the answer or value provided for a specific question within a contract, capturing user input or selections related to the contract's configuration.

```
public class ContractQuestionValue : StampedEntity
```

Inheritance

object ← StampedEntity ← ContractQuestionValue

Inherited Members

StampedEntity.CreatedAt , StampedEntity.UpdatedAt , StampedEntity.DeletedAt , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Contract

Gets the contract to which this question value is associated.

```
public Contract? Contract { get; init; }
```

Property Value

Contract

ContractId

Gets the identifier of the contract to which this question value belongs.

```
public int ContractId { get; init; }
```

Property Value

int

Option

Gets the selected option for the question, if applicable.

```
public FieldOption? Option { get; init; }
```

Property Value

FieldOption

Question

Gets the question for which this value is provided.

```
public Question? Question { get; init; }
```

Property Value

Question

QuestionId

Gets the identifier of the question for which this value is provided.

```
public int QuestionId { get; init; }
```

Property Value

int

Value

Gets the value provided for the question, which may be text, a selection identifier, or other data as appropriate.

```
public required string Value { get; init; }
```

Property Value

string

Namespace Instaclause.Domain.Customers

Classes

Customer

Represents a customer of an accountant. Is the owner of contracts.

CustomerParty

Represents a party related to a customer, detailing information about an individual or organization's role or identity in the context of contracts.

CustomerPartyFieldValue

Represents a specific value or attribute associated with a customer party, detailing information such as contact details, identifiers, or preferences.

User

Represents a user in the system, typically an individual who interacts with the application, such as a customer or an administrator.

Class Customer

Namespace: Instaclause.Domain.Customers

Assembly: Instaclause.Domain.dll

Represents a customer of an accountant. Is the owner of contracts.

```
public class Customer
```

Inheritance

object ← Customer

Inherited Members

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

Properties

Contracts

Gets the collection of contracts associated with this customer.

```
public List<Contract>? Contracts { get; init; }
```

Property Value

List<Contract>

Id

Gets the identifier for the customer.

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

Property Value

int

Name

Gets the name of the customer, which could be an individual's name or an organization's name.

```
public required string Name { get; init; }
```

Property Value

string

Parties

Gets the collection of parties that represent this customer in various contracts.

```
public List<CustomerParty>? Parties { get; init; }
```

Property Value

List<CustomerParty>

Class CustomerParty

Namespace: Instaclause.Domain.Customers

Assembly: Instaclause.Domain.dll

Represents a party related to a customer, detailing information about an individual or organization's role or identity in the context of contracts.

```
public class CustomerParty
```

Inheritance

object ← CustomerParty

Inherited Members

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

Properties

CustomerId

Gets the identifier of the customer to which this party belongs.

```
public int CustomerId { get; init; }
```

Property Value

int

Fields

Gets the collection of field values associated with this party, detailing specific information or attributes of the party.

```
public List<CustomerPartyFieldValue>? Fields { get; init; }
```

Property Value

List<CustomerPartyFieldValue>

Id

Gets the identifier for the customer party.

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

Property Value

int

Name

Gets the name of the party, which may represent an individual's name or an organization's name, depending on the party type.

```
public required string Name { get; init; }
```

Property Value

string

Type

Gets the type of the party, providing additional context about the party's role or identity.

```
public PartyType? Type { get; init; }
```

Property Value

PartyType

TypeId

Gets the type identifier for the party, indicating whether the party is an individual, company, or another entity type.

```
public int TypeId { get; init; }
```

Property Value

int

Class CustomerPartyFieldValue

Namespace: Instaclause.Domain.Customers

Assembly: Instaclause.Domain.dll

Represents a specific value or attribute associated with a customer party, detailing information such as contact details, identifiers, or preferences.

```
public class CustomerPartyFieldValue
```

Inheritance

object ← CustomerPartyFieldValue

Inherited Members

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

Properties

CustomerParty

Gets the customer party to which this field value is associated, representing the entity (individual or organization) the field value describes.

```
public CustomerParty? CustomerParty { get; init; }
```

Property Value

CustomerParty

CustomerPartyId

Gets the identifier of the customer party to which this field value belongs.

```
public int CustomerPartyId { get; init; }
```

Property Value

int

Field

Gets the field associated with this value, providing context about the type of information this value represents.

```
public Field? Field { get; init; }
```

Property Value

Field

FieldId

Gets the identifier of the field that this value corresponds to, linking to a specific attribute or information type.

```
public int FieldId { get; init; }
```

Property Value

int

Id

Gets the identifier for the customer party field value.

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

Property Value

int

Value

Gets the actual value for the field, which could be text, a numeric value, or another data type depending on the field's nature.

```
public required string Value { get; init; }
```

Property Value

string

Class User

Namespace: Instaclause.Domain.Customers

Assembly: Instaclause.Domain.dll

Represents a user in the system, typically an individual who interacts with the application, such as a customer or an administrator.

```
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

Customer

Gets the customer associated with this user, representing the relationship between the user and a customer profile in the system.

```
public Customer? Customer { get; init; }
```

Property Value

Customer

CustomerId

Gets the customer identifier associated with this user, linking the user to a customer profile if applicable.

```
public int CustomerId { get; init; }
```

Property Value

int

Email

Gets the email address for the user, used for communication and possibly for login.

```
public required string Email { get; init; }
```

Property Value

string

Id

Gets the identifier for the user.

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

Property Value

int

Username

Gets the username for the user, used for identification and login purposes.

```
public required string Username { get; init; }
```

Property Value

string

Namespace Instaclause.Domain.Documents

Classes

Document

Represents a document within a model. The document is a collection of clauses and is what orchestrates the clauses into a coherent document. It's the equivalent of the old system's Google Docs templates.

DocumentClause

Represents a clause within a document, with their conditions to be rendered and order in the document. These conditions are based on the questionnaire answers and party assignments. The conditions can also be described as part of the old system's modules.

Class Document

Namespace: Instaclause.Domain.Documents

Assembly: Instaclause.Domain.dll

Represents a document within a model. The document is a collection of clauses and is what orchestrates the clauses into a coherent document. It's the equivalent of the old system's Google Docs templates.

```
public class Document : TranslatableEntity<Document>
```

Inheritance

object ← TranslatableEntity<Document> ← Document

Inherited Members

TranslatableEntity<Document>.BaseLanguageEntityId , TranslatableEntity<Document>.BaseLanguageEntity , TranslatableEntity<Document>.LanguageId , TranslatableEntity<Document>.TranslationStatusId , TranslatableEntity<Document>.Language , TranslatableEntity<Document>.TranslationStatus , TranslatableEntity<Document>.Translations , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Clauses

Gets the collection of clauses associated with this document, representing the individual parts or sections of the document.

```
public List<DocumentClause>? Clauses { get; init; }
```

Property Value

List<DocumentClause>

ContractModel

Gets the contract model associated with this document, providing a link to the model if this document is part of a contract model's documentation.

```
public ContractModel? ContractModel { get; init; }
```

Property Value

ContractModel

ContractModelId

Gets the contract model identifier associated with this document, indicating the model from which this document originates if applicable.

```
public int ContractModelId { get; init; }
```

Property Value

int

Id

Gets the identifier for the document.

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

Property Value

int

Name

Gets the name of the document, used for identification and display purposes.

```
public required string Name { get; init; }
```

Property Value

string

Class DocumentClause

Namespace: Instaclause.Domain.Documents

Assembly: Instaclause.Domain.dll

Represents a clause within a document, with their conditions to be rendered and order in the document. These conditions are based on the questionnaire answers and party assignments. The conditions can also be described as part of the old system's modules.

```
public class DocumentClause
```

Inheritance

object ← DocumentClause

Inherited Members

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

Properties

Clause

Gets the clause entity associated with this document clause, detailing the content or provision this document clause represents.

```
public Clause? Clause { get; init; }
```

Property Value

Clause

ClauseId

Gets the clause identifier, linking this document clause to a specific clause entity.

```
public int ClauseId { get; init; }
```

Property Value

int

Condition

Gets the condition under which this clause is included or applied within the document.

```
public string? Condition { get; init; }
```

Property Value

string

Document

Gets the document to which this clause belongs, representing the larger entity that encompasses this clause.

```
public Document? Document { get; init; }
```

Property Value

Document

DocumentId

Gets the document identifier, linking this clause to the document it is part of.

```
public int DocumentId { get; init; }
```

Property Value

int

Id

Gets the identifier for the document clause.

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

Property Value

int

Order

Gets the order of this clause within the document, determining its sequence among other clauses.

```
public int Order { get; init; }
```

Property Value

int

Namespace Instaclause.Domain.Fields

Classes

Field

Represents a field within the system, which can be a part of a party, question, or clause.

FieldGroup

Represents a group of fields, allowing for the logical grouping and organization of fields within the system.

FieldOption

Represents an option within a field, specifically for fields that allow selecting from predefined values, such as dropdowns or radio buttons.

FieldType

Represents a type of field within the system, defining how a field behaves and is interacted with, such as text, number, date, etc.

FieldTypeFieldTypeUnit

Represents the association between a field type and a field type unit, allowing for the specification of units for fields that require them.

FieldTypeUnit

Represents a unit of measurement or categorization for field types that require or support specifying a unit, such as length, weight, currency, etc.

Class Field

Namespace: Instaclause.Domain.Fields

Assembly: Instaclause.Domain.dll

Represents a field within the system, which can be a part of a party, question, or clause.

```
public class Field : TranslatableEntity<Field>
```

Inheritance

object ← TranslatableEntity<Field> ← Field

Inherited Members

TranslatableEntity<Field>.BaseLanguageEntityId , TranslatableEntity<Field>.BaseLanguageEntity , TranslatableEntity<Field>.LanguageId , TranslatableEntity<Field>.TranslationStatusId , TranslatableEntity<Field>.Language , TranslatableEntity<Field>.TranslationStatus , TranslatableEntity<Field>.Translations , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Clauses

Gets the collection of clauses that incorporate this field, linking the field to its application within various clauses.

```
public List<Clause>? Clauses { get; init; }
```

Property Value

List<Clause>

Description

Gets the description of the field, providing additional details about its purpose or usage.

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

Property Value

string

FieldType

Gets the field type of this field, detailing how the field is rendered and the kind of data it holds.

```
public FieldType FieldType { get; init; }
```

Property Value

FieldType

FieldTypeId

Gets the field type identifier, linking this field to its defined type which dictates how the field is rendered and interacted with.

```
public int FieldTypeId { get; init; }
```

Property Value

int

Group

Gets the field group to which this field belongs, providing a means to categorize and organize fields.

```
public FieldGroup? Group { get; init; }
```

Property Value

FieldGroup

GroupId

Gets the group identifier to which this field belongs, allowing for organizational structuring of fields.

```
public int? GroupId { get; init; }
```

Property Value

int?

Id

Gets the identifier for the field.

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

Property Value

int

Identifier

Gets the identifier used to reference the field within the system, ensuring unique identification.

```
public required string Identifier { get; init; }
```

Property Value

string

Label

Gets the label for the field, used for display purposes to identify the field to users.

```
public required string Label { get; init; }
```

Property Value

string

Options

Gets the collection of options available for this field, applicable to fields that provide a selection of values.

```
public List<FieldOption>? Options { get; init; }
```

Property Value

List<FieldOption>

PartyTypes

Gets the collection of party types that utilize this field, linking the field to its use within different party contexts.

```
public List<PartyType>? PartyTypes { get; init; }
```

Property Value

List<PartyType>

Questions

Gets the collection of questions that utilize this field, linking the field to its use cases within questions.

```
public List<Question>? Questions { get; init; }
```

Property Value

List<Question>

ValidationRegex

Gets the validation regex pattern for the field, used to validate input data for the field.

```
public string? ValidationRegex { get; init; }
```

Property Value

string

Class FieldGroup

Namespace: Instaclause.Domain.Fields

Assembly: Instaclause.Domain.dll

Represents a group of fields, allowing for the logical grouping and organization of fields within the system.

```
public class FieldGroup : TranslatableEntity<FieldGroup>
```

Inheritance

object ← TranslatableEntity<FieldGroup> ← FieldGroup

Inherited Members

TranslatableEntity<FieldGroup>.BaseLanguageEntityId , TranslatableEntity<FieldGroup>.BaseLanguageEntity , TranslatableEntity<FieldGroup>.LanguageId , TranslatableEntity<FieldGroup>.TranslationStatusId , TranslatableEntity<FieldGroup>.Language , TranslatableEntity<FieldGroup>.TranslationStatus , TranslatableEntity<FieldGroup>.Translations , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Fields

Gets the collection of fields that belong to this group, providing a means to categorize and organize fields under a common grouping.

```
public List<Field>? Fields { get; init; }
```

Property Value

List<Field>

Id

Gets the identifier for the field group.

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

Property Value

int

Name

Gets the name of the field group, used for identification and organizational purposes.

```
public required string Name { get; init; }
```

Property Value

string

Class FieldOption

Namespace: Instaclause.Domain.Fields

Assembly: Instaclause.Domain.dll

Represents an option within a field, specifically for fields that allow selecting from predefined values, such as dropdowns or radio buttons.

```
public class FieldOption : TranslatableEntity<FieldOption>
```

Inheritance

object ← TranslatableEntity<FieldOption> ← FieldOption

Inherited Members

TranslatableEntity<FieldOption>.BaseLanguageEntityId , TranslatableEntity<FieldOption>.BaseLanguageEntity , TranslatableEntity<FieldOption>.LanguageId , TranslatableEntity<FieldOption>.TranslationStatusId , TranslatableEntity<FieldOption>.Language , TranslatableEntity<FieldOption>.TranslationStatus , TranslatableEntity<FieldOption>.Translations , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Field

Gets the field to which this option belongs, establishing a relationship between the option and its parent field.

```
public Field? Field { get; init; }
```

Property Value

Field

FieldId

Gets the field identifier, linking this option to the field it belongs to.

```
public int FieldId { get; init; }
```

Property Value

int

Id

Gets the identifier for the field option.

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

Property Value

int

Label

Gets the label for the option, used for display purposes to present the option to users.

```
public required string Label { get; init; }
```

Property Value

string

Order

Gets the order in which this option appears relative to other options within the same field, affecting display sequence.

```
public int Order { get; init; }
```

Property Value

int

Value

Gets the value associated with this option, which is the actual data stored or transmitted when this option is selected.

```
public required string Value { get; init; }
```

Property Value

string

Class FieldType

Namespace: Instaclause.Domain.Fields

Assembly: Instaclause.Domain.dll

Represents a type of field within the system, defining how a field behaves and is interacted with, such as text, number, date, etc.

```
public class FieldType : TranslatableEntity<FieldType>
```

Inheritance

object ← TranslatableEntity<FieldType> ← FieldType

Inherited Members

TranslatableEntity<FieldType>.BaseLanguageEntityId , TranslatableEntity<FieldType>.BaseLanguageEntity , TranslatableEntity<FieldType>.LanguageId , TranslatableEntity<FieldType>.TranslationStatusId , TranslatableEntity<FieldType>.Language , TranslatableEntity<FieldType>.TranslationStatus , TranslatableEntity<FieldType>.Translations , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

CastAs

Gets the data type to which the field value should be cast, facilitating data handling and validation.

```
public required string CastAs { get; init; }
```

Property Value

string

Id

Gets the identifier for the field type.

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

Property Value

int

Type

Gets the type of the field, which determines how the field is rendered and how data validation is performed.

```
public required string Type { get; init; }
```

Property Value

string

Units

Gets the collection of units associated with this field type, applicable for types that support or require units, such as measurements.

```
public List<FieldTypeUnit>? Units { get; init; }
```

Property Value

List<FieldTypeUnit>

ValidationRegex

Gets the regex pattern used to validate field values, ensuring that data entered matches the expected format.

```
public string? ValidationRegex { get; init; }
```

Property Value

string

Class FieldTypeFieldTypeUnit

Namespace: Instaclause.Domain.Fields

Assembly: Instaclause.Domain.dll

Represents the association between a field type and a field type unit, allowing for the specification of units for fields that require them.

```
public class FieldTypeFieldTypeUnit
```

Inheritance

object ← FieldTypeFieldTypeUnit

Inherited Members

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

Properties

FieldType

Gets the field type in this association.

```
public FieldType? FieldType { get; init; }
```

Property Value

FieldType

FieldTypeId

Gets the field type identifier, linking to the field type in the association.

```
public int FieldTypeId { get; init; }
```

Property Value

int

FieldTypeUnit

Gets the field type unit in this association, specifying the unit applicable to the field type.

```
public FieldTypeUnit? FieldTypeUnit { get; init; }
```

Property Value

FieldTypeUnit

FieldTypeUnitId

Gets the field type unit identifier, linking to the field type unit in the association.

```
public int FieldTypeUnitId { get; init; }
```

Property Value

int

Class FieldTypeUnit

Namespace: Instaclause.Domain.Fields

Assembly: Instaclause.Domain.dll

Represents a unit of measurement or categorization for field types that require or support specifying a unit, such as length, weight, currency, etc.

```
public class FieldTypeUnit : TranslatableEntity<FieldTypeUnit>
```

Inheritance

object ← TranslatableEntity<FieldTypeUnit> ← FieldTypeUnit

Inherited Members

TranslatableEntity<FieldTypeUnit>.BaseLanguageEntityId ,
TranslatableEntity<FieldTypeUnit>.BaseLanguageEntity ,
TranslatableEntity<FieldTypeUnit>.LanguageId ,
TranslatableEntity<FieldTypeUnit>.TranslationStatusId ,
TranslatableEntity<FieldTypeUnit>.Language ,
TranslatableEntity<FieldTypeUnit>.TranslationStatus ,
TranslatableEntity<FieldTypeUnit>.Translations , object.Equals(object) ,
object.Equals(object, object) , object.GetHashCode() , object.GetType() ,
object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

FieldTypes

Gets the collection of field types associated with this unit, linking it to the field types that can use this unit.

```
public List<FieldType>? FieldTypes { get; init; }
```

Property Value

List<FieldType>

Id

Gets the identifier for the field type unit.

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

Property Value

int

Name

Gets the name of the unit, used for display and identification purposes.

```
public required string Name { get; init; }
```

Property Value

string

Unit

Gets the symbol or abbreviation for the unit, used for compact display purposes.

```
public required string Unit { get; init; }
```

Property Value

string

Namespace Instaclause.Domain.Jurisdictions

Classes

Jurisdiction

Represents a legal jurisdiction under which contracts, clauses, and other legal documents can be governed, such as a country, state, or specific legal area.

Class Jurisdiction

Namespace: Instaclause.Domain.Jurisdictions

Assembly: Instaclause.Domain.dll

Represents a legal jurisdiction under which contracts, clauses, and other legal documents can be governed, such as a country, state, or specific legal area.

```
public class Jurisdiction
```

Inheritance

object ← Jurisdiction

Inherited Members

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

Properties

Id

Gets the identifier for the jurisdiction.

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

Property Value

int

Models

Gets the collection of contract models associated with this jurisdiction, indicating the models that are applicable under this legal jurisdiction.

```
public List<ContractModel>? Models { get; init; }
```

Property Value

List<ContractModel>

Name

Gets the name of the jurisdiction, used for identification and display purposes.

```
public required string Name { get; init; }
```

Property Value

string

Parent

Gets the parent jurisdiction, linking this jurisdiction to its parent in the hierarchy, if applicable.

```
public Jurisdiction? Parent { get; init; }
```

Property Value

Jurisdiction

ParentId

Gets the parent jurisdiction identifier, allowing for hierarchical structuring of jurisdictions.

```
public int? ParentId { get; init; }
```

Property Value

int?

Namespace Instaclause.Domain.PartyTypes

Classes

PartyType

Represents a type of party that can be involved in contracts, such as an individual or company, defining the roles and information requirements for parties in contracts.

PartyTypeField

Represents the association between a party type and a field, defining the fields or information that are required or applicable for a specific type of party.

Class PartyType

Namespace: Instaclause.Domain.PartyTypes

Assembly: Instaclause.Domain.dll

Represents a type of party that can be involved in contracts, such as an individual or company, defining the roles and information requirements for parties in contracts.

```
public class PartyType : TranslatableEntity<PartyType>
```

Inheritance

object ← TranslatableEntity<PartyType> ← PartyType

Inherited Members

TranslatableEntity<PartyType>.BaseLanguageEntityId , TranslatableEntity<PartyType>.BaseLanguageEntity , TranslatableEntity<PartyType>.LanguageId , TranslatableEntity<PartyType>.TranslationStatusId , TranslatableEntity<PartyType>.Language , TranslatableEntity<PartyType>.TranslationStatus , TranslatableEntity<PartyType>.Translations , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Fields

Gets the collection of fields that are associated with this party type, defining the information requirements or attributes for parties of this type.

```
public List<Field>? Fields { get; init; }
```

Property Value

List<Field>

Id

Gets the identifier for the party type.

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

Property Value

int

Name

Gets the name of the party type, used for identification and categorization purposes.

```
public required string Name { get; init; }
```

Property Value

string

TemplateData

Gets the template data for the party type, which may define a standard format or structure for representing parties of this type in documents or contracts.

```
public required string TemplateData { get; init; }
```

Property Value

string

Class PartyTypeField

Namespace: Instaclause.Domain.PartyTypes

Assembly: Instaclause.Domain.dll

Represents the association between a party type and a field, defining the fields or information that are required or applicable for a specific type of party.

```
public class PartyTypeField
```

Inheritance

object ← PartyTypeField

Inherited Members

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

Properties

Field

Gets the field associated with this party type, detailing the specific information or attribute this field represents.

```
public Field? Field { get; init; }
```

Property Value

Field

FieldId

Gets the field identifier, linking to the field that is associated with the party type.

```
public int FieldId { get; init; }
```

Property Value

int

FieldTypeUnit

Gets the field type unit associated with this field within the context of the party type, if applicable.

```
public FieldTypeUnit? FieldTypeUnit { get; init; }
```

Property Value

FieldTypeUnit

FieldTypeUnitId

Gets the field type unit identifier, if applicable, specifying the unit or type of data expected for this field within the context of the party type.

```
public int? FieldTypeUnitId { get; init; }
```

Property Value

int?

Id

Gets the identifier for the association.

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

Property Value

int

PartyType

Gets the party type in this association, linking the field to the type of party it applies to.

```
public PartyType? PartyType { get; init; }
```

Property Value

PartyType

PartyTypeId

Gets the party type identifier, linking to the party type in the association.

```
public int PartyTypeId { get; init; }
```

Property Value

int

Required

Gets a value indicating whether the field is required for parties of this type, ensuring necessary information is collected or presented.

```
public bool Required { get; init; }
```

Property Value

bool

ValidationRegex

Gets the validation regex pattern for the field within the context of this party type, if applicable, used to validate input data.

```
public string? ValidationRegex { get; init; }
```

Property Value

string

Namespace Instaclause.Domain.Questions

Classes

Question

Represents a question that may be posed within the context of a contract or document. Used to gather information or make decisions that affect the document's content or clauses. Are mainly used in the conditional clauses.

Class Question

Namespace: Instaclause.Domain.Questions

Assembly: Instaclause.Domain.dll

Represents a question that may be posed within the context of a contract or document. Used to gather information or make decisions that affect the document's content or clauses. Are mainly used in the conditional clauses.

```
public class Question : TranslatableEntity<Question>
```

Inheritance

object ← TranslatableEntity<Question> ← Question

Inherited Members

TranslatableEntity<Question>.BaseLanguageEntityId , TranslatableEntity<Question>.BaseLanguageEntity , TranslatableEntity<Question>.LanguageId , TranslatableEntity<Question>.TranslationStatusId , TranslatableEntity<Question>.Language , TranslatableEntity<Question>.TranslationStatus , TranslatableEntity<Question>.Translations , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

ChildQuestions

Gets the collection of child questions that may follow based on the response to this question.

```
public List<Question>? ChildQuestions { get; init; }
```

Property Value

List<Question>

ContractModel

Gets the contract model associated with this question, linking it to the context in which the question is relevant.

```
public ContractModel? ContractModel { get; init; }
```

Property Value

ContractModel

ContractModelId

Gets the identifier of the contract model to which this question is associated, indicating its relevance to specific types of contracts.

```
public int ContractModelId { get; init; }
```

Property Value

int

Field

Gets the field associated with this question, defining the response type and validation for the question's answer.

```
public Field Field { get; init; }
```

Property Value

Field

FieldId

Gets the field identifier, linking the question to a specific field that defines the type of response expected for the question.

```
public int FieldId { get; init; }
```

Property Value

int

Id

Gets the identifier for the question.

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

Property Value

int

Parent

Gets the parent question, establishing a hierarchical relationship between questions.

```
public Question? Parent { get; init; }
```

Property Value

Question

ParentId

Gets the optional identifier of a parent question, allowing for the structuring of questions in a hierarchical manner, where answers may lead to additional questions.

```
public int? ParentId { get; init; }
```

Property Value

int?

Title

Gets the title of the question, which may be presented to the user as part of the questionnaire or form.

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

Property Value

string

Namespace Instaclause.Domain.Translations

Classes

Language

Represents a language supported by the system, used for translations and localization of content, documents, and user interfaces.

Translation

Represents a translation of content within the system, facilitating multilingual support for documents, questions, and other textual content.

TranslationStatus

Represents the status of a translation within the system, indicating whether a translation is pending, approved, or requires review, thus managing the workflow and quality control of translations.

Class Language

Namespace: Instaclause.Domain.Translations

Assembly: Instaclause.Domain.dll

Represents a language supported by the system, used for translations and localization of content, documents, and user interfaces.

```
public class Language
```

Inheritance

object ← Language

Inherited Members

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

Properties

Code

Gets the ISO language code, facilitating standardized language identification across the system and external integrations.

```
public required string Code { get; init; }
```

Property Value

string

Country

Gets the country or region code associated with this language, helping to specify regional dialects or variations.

```
public required string Country { get; init; }
```

Property Value

string

Id

Gets the identifier for the language.

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

Property Value

int

Name

Gets the name of the language, used for display and selection purposes.

```
public required string Name { get; init; }
```

Property Value

string

Class Translation

Namespace: Instaclause.Domain.Translations

Assembly: Instaclause.Domain.dll

Represents a translation of content within the system, facilitating multilingual support for documents, questions, and other textual content.

```
public class Translation
```

Inheritance

object ← Translation

Inherited Members

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

Properties

Approved

Gets a value indicating whether the translation has been approved for use, ensuring quality and accuracy of translated content.

```
public bool Approved { get; init; }
```

Property Value

bool

Id

Gets the identifier for the translation.

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

Property Value

int

Result

Gets the result of the translation, containing the translated text in the target language.

```
public required string Result { get; init; }
```

Property Value

string

SourceLanguage

Gets the source language of the translation, linking to the Language entity that represents the language from which the content was translated.

```
public Language? SourceLanguage { get; init; }
```

Property Value

Language

SourceLanguageId

Gets the source language identifier, indicating the original language of the text.

```
public int SourceLanguageId { get; init; }
```

Property Value

int

TargetLanguage

Gets the target language of the translation, linking to the Language entity that represents the language into which the content is translated.

```
public Language? TargetLanguage { get; init; }
```

Property Value

Language

TargetLanguageId

Gets the target language identifier, indicating the language into which the text is translated.

```
public int TargetLanguageId { get; init; }
```

Property Value

int

Text

Gets the text to be translated, serving as the source content for the translation process.

```
public required string Text { get; init; }
```

Property Value

string

Uses

Gets the number of times this translation has been used, providing insights into the relevance and utility of specific translations within the system.

```
public int Uses { get; init; }
```

Property Value

int

Class TranslationStatus

Namespace: Instaclause.Domain.Translations

Assembly: Instaclause.Domain.dll

Represents the status of a translation within the system, indicating whether a translation is pending, approved, or requires review, thus managing the workflow and quality control of translations.

```
public class TranslationStatus
```

Inheritance

object ← TranslationStatus

Inherited Members

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

Properties

Id

Gets the identifier for the translation status.

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

Property Value

int

Name

Gets the name of the translation status, used for display and to indicate the current state of a translation within the translation management process.

```
public required string Name { get; init; }
```

Property Value

string