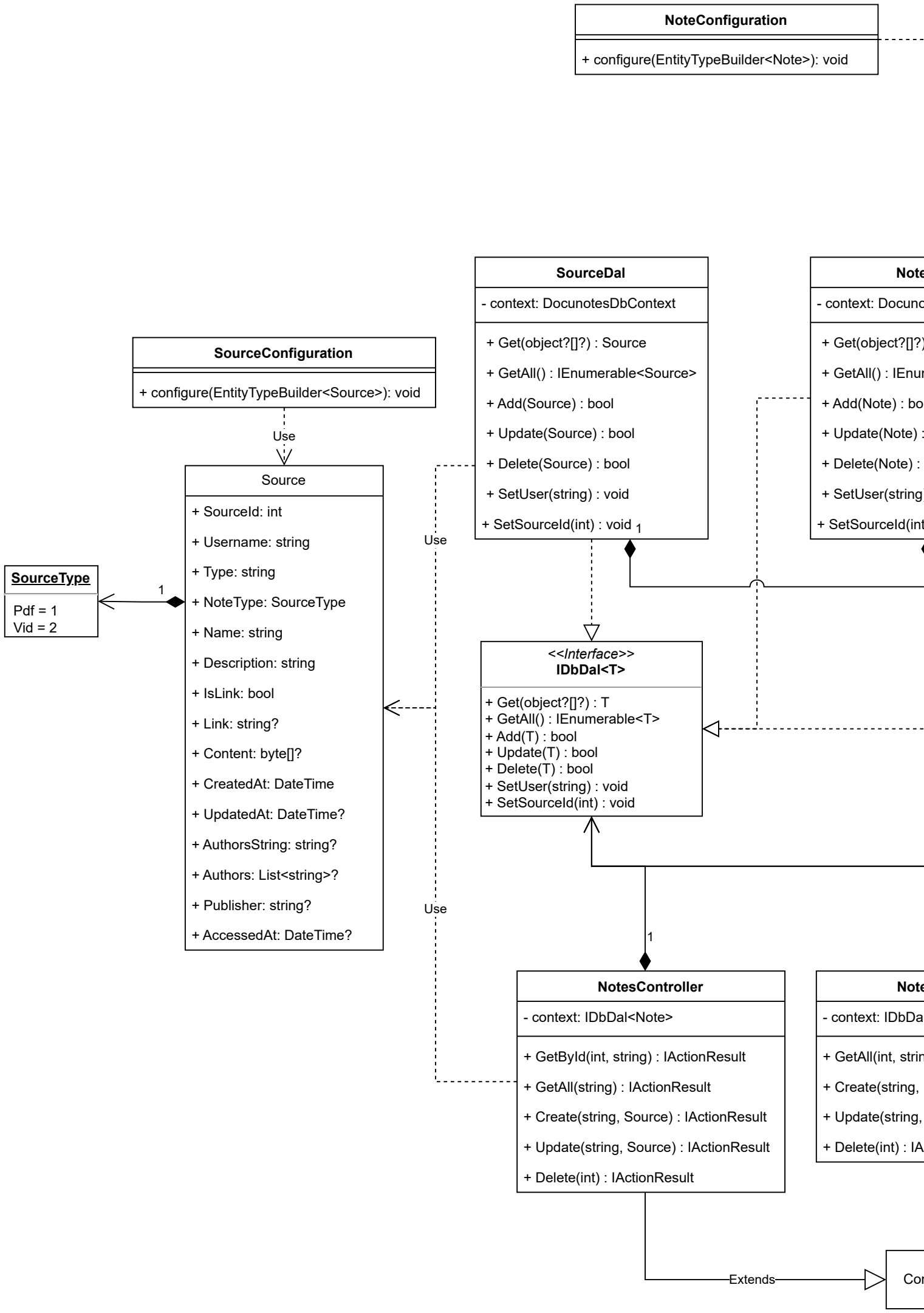
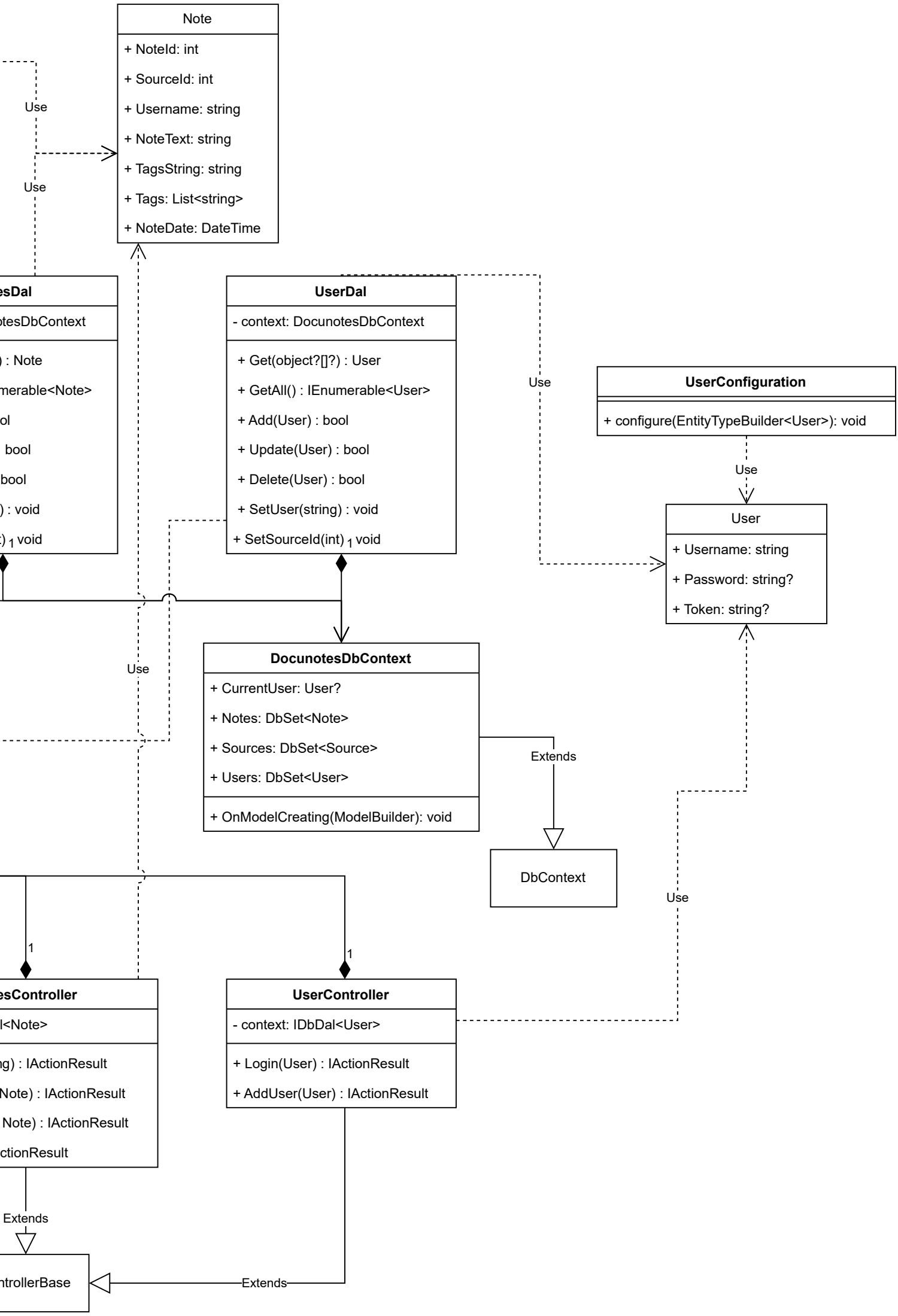
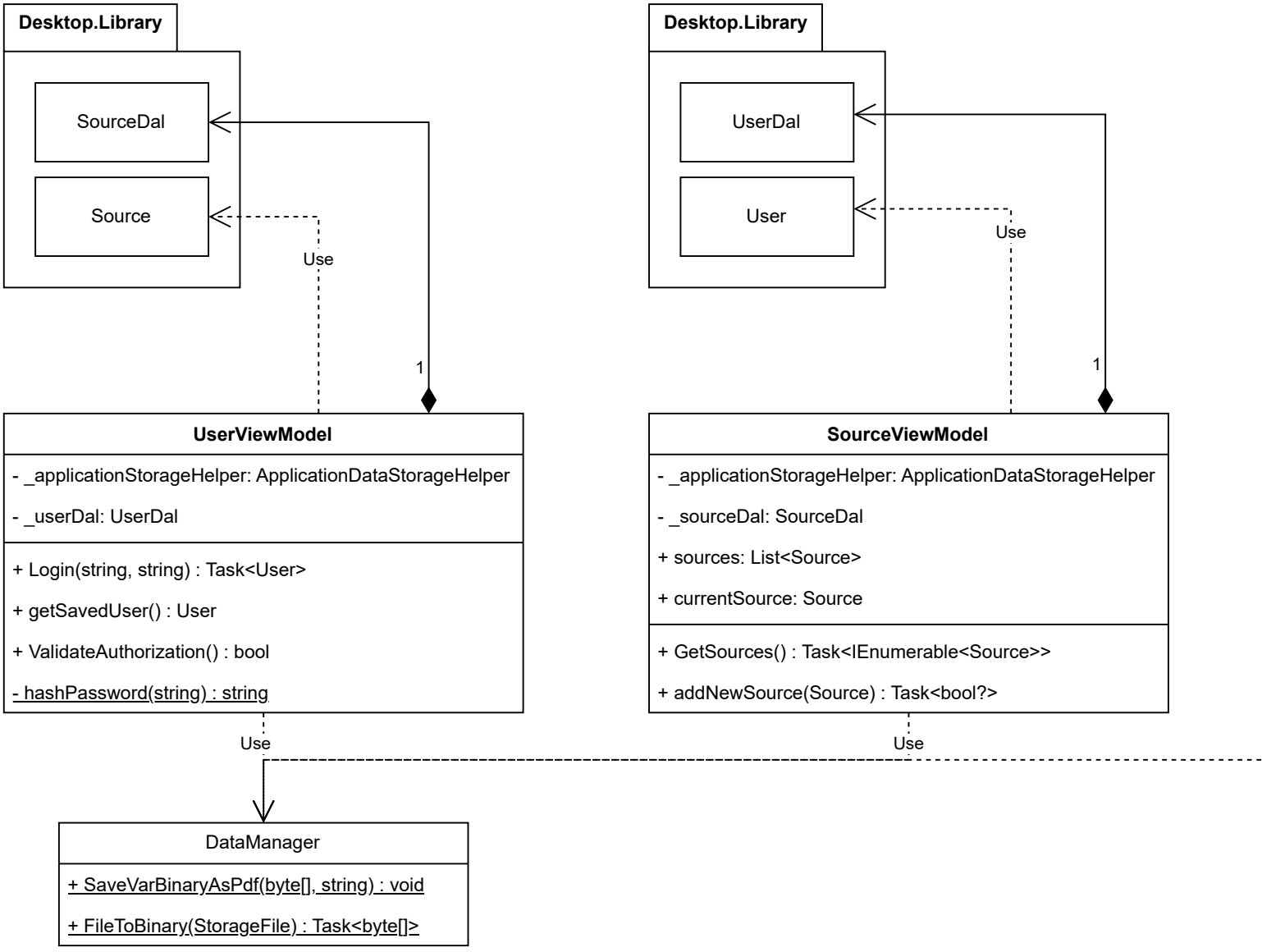
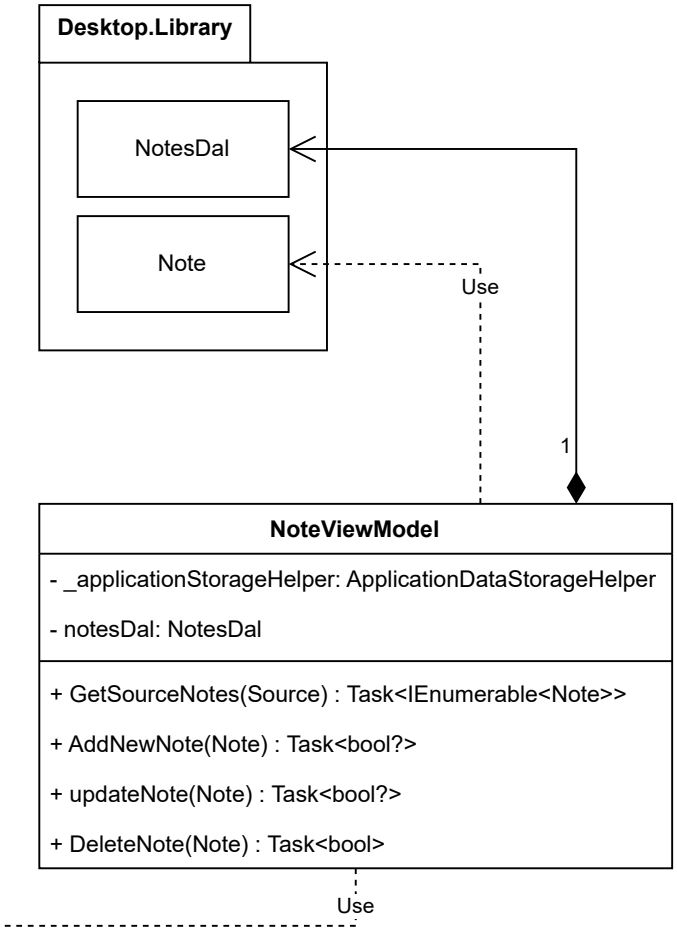
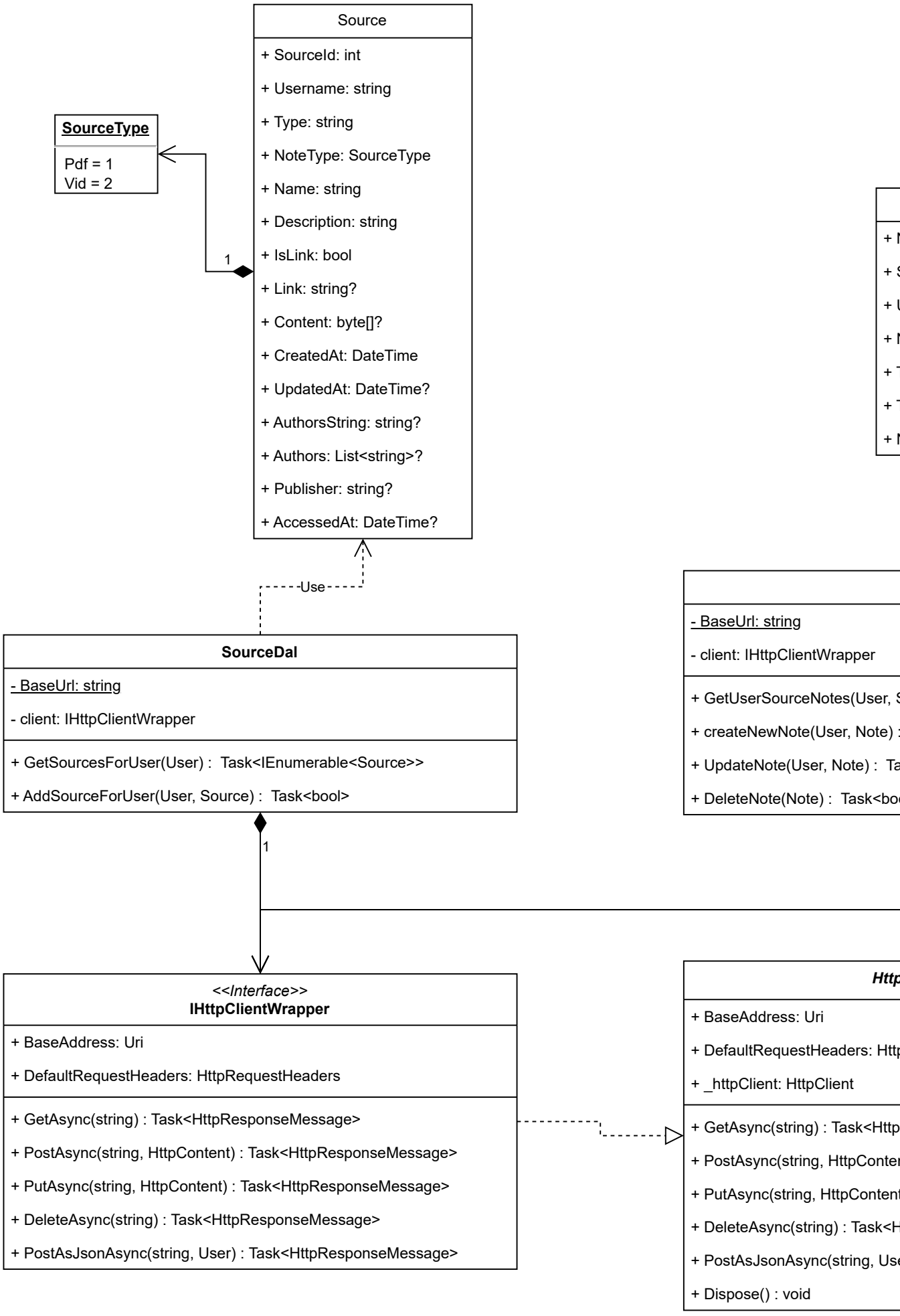
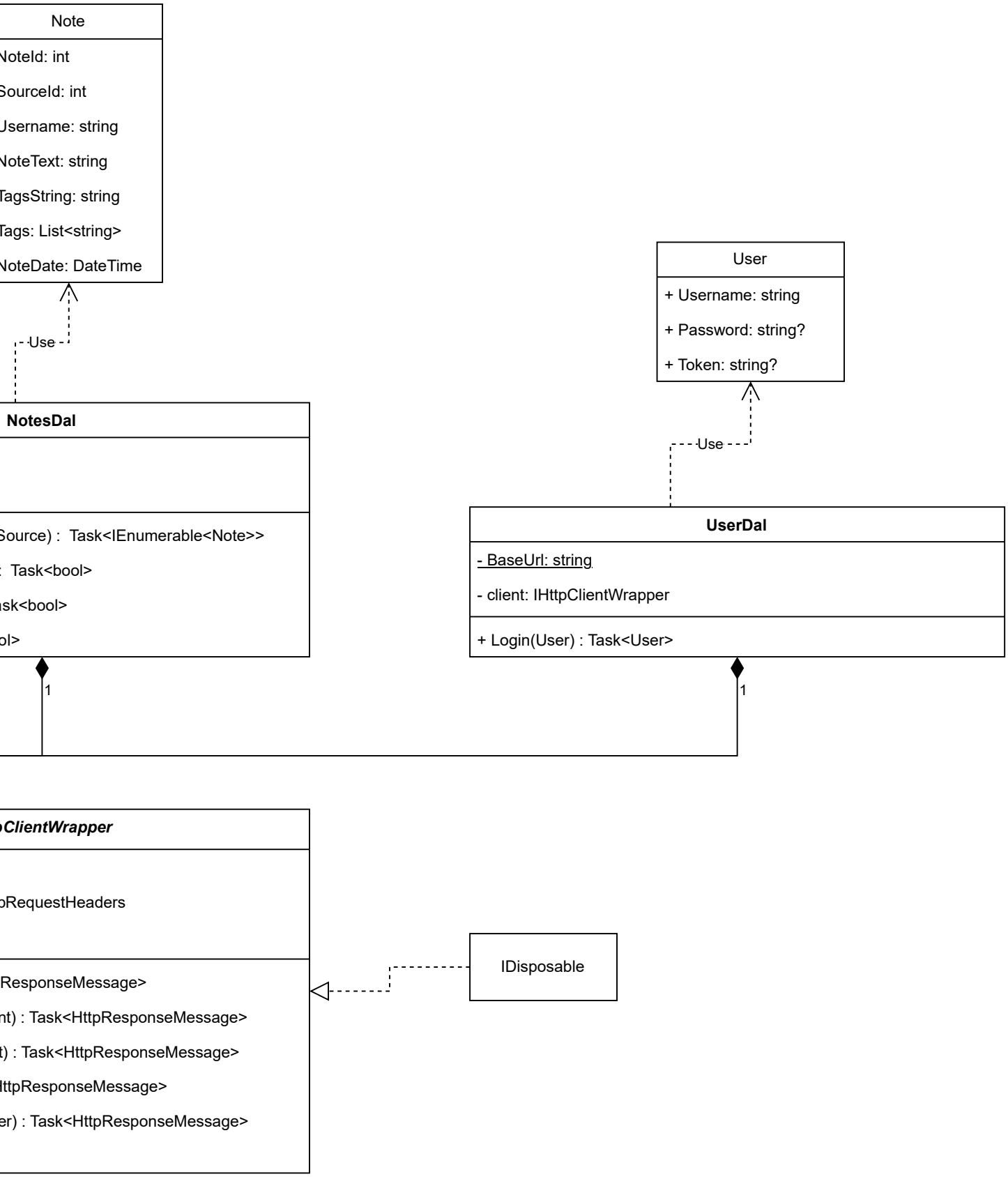

NoteConfiguration
+ configure(EntityTypeBuilder<Note>): void
SourceDal
- context: DocunotesDbContext
+ Get(object?[]?) : Source
+ GetAll() : IEnumerable<Source>
+ Add(Source) : bool
+ Update(Source) : bool
+ Delete(Source) : bool
+ SetUser(string) : void
+ SetSourceId(int) : void
Note
- context: Docunc
+ Get(object?[]?)
+ GetAll() : IEnur
+ Add(Note) : bo
+ Update(Note) :
+ Delete(Note) :
+ SetUser(string
+ SetSourceId(int
SourceConfiguration
+ configure(EntityTypeBuilder<Source>): void
Use
Source
+ SourceId: int
+ Username: string
+ Type: string
+ NoteType: SourceType
+ Name: string
+ Description: string
+ IsLink: bool
+ Link: string?
+ Content: byte[]?
+ CreatedAt: DateTime
+ UpdatedAt: DateTime?
+ AuthorsString: string?
+ Authors: List<string>?
+ Publisher: string?
+ AccessedAt: DateTime?
SourceType
Pdf = 1
Vid = 2
1
Use
1
<<Interface>>
IDbDal<T>
+ Get(object?[]?) : T
+ GetAll() : IEnumerable<T>
+ Add(T) : bool
+ Update(T) : bool
+ Delete(T) : bool
+ SetUser(string) : void
+ SetSourceId(int) : void
Use
1
NotesController
- context: IDbDal<Note>
+ GetById(int, string) : IActionResult
+ GetAll(string) : IActionResult
+ Create(string, Source) : IActionResult
+ Update(string, Source) : IActionResult
+ Delete(int) : IActionResult
Note
- context: IDbDa
+ GetAll(int, strin
+ Create(string,
+ Update(string,
+ Delete(int) : IA
Extends
Co


Note
+ NoteId: int
+ SourceId: int
+ Username: string
+ NoteText: string
+ TagsString: string
+ Tags: List<string>
+ NoteDate: DateTime
Use
Use
esDal
otesDbContext
) : Note
merable<Note>
ol
bool
bool
) : void
) 1 void
UserDal
- context: DocunotesDbContext
+ Get(object?[]?) : User
+ GetAll() : IEnumerable<User>
+ Add(User) : bool
+ Update(User) : bool
+ Delete(User) : bool
+ SetUser(string) : void
+ SetSourceId(int) 1 void
Use
UserConfiguration
+ configure(EntityTypeBuilder<User>): void
Use
User
+ Username: string
+ Password: string?
+ Token: string?
Use
DocunotesDbContext
+ CurrentUser: User?
+ Notes: DbSet<Note>
+ Sources: DbSet<Source>
+ Users: DbSet<User>
+ OnModelCreating(ModelBuilder): void
Extends
DbContext
Use
1
1
esController
l<Note>
ng) : IActionResult
Note) : IActionResult
Note) : IActionResult
ctionResult
UserController
- context: IDbDal<User>
+ Login(User) : IActionResult
+ AddUser(User) : IActionResult
Extends
ntrollerBase
Extends


Desktop.Library
SourceDal
Source
Use
1
UserViewModel
- _applicationStorageHelper: ApplicationDataStorageHelper
- _userDal: UserDal
+ Login(string, string) : Task<User>
+ getSavedUser() : User
+ ValidateAuthorization() : bool
- hashPassword(string) : string
Desktop.Library
UserDal
User
Use
1
SourceViewModel
- _applicationStorageHelper: ApplicationDataStorageHelper
- _sourceDal: SourceDal
+ sources: List<Source>
+ currentSource: Source
+ GetSources() : Task<IEnumerable<Source>>
+ addNewSource(Source) : Task<bool?>
Use
Use
DataManager
+ SaveVarBinaryAsPdf(byte[], string) : void
+ FileToBinary(StorageFile) : Task<byte[]>


Desktop.Library
NotesDal
Note
Use
1
NoteViewModel
- _applicationStorageHelper: ApplicationDataStorageHelper
- notesDal: NotesDal
+ GetSourceNotes(Source) : Task<IEnumerable<Note>>
+ AddNewNote(Note) : Task<bool?>
+ updateNote(Note) : Task<bool?>
+ DeleteNote(Note) : Task<bool>
Use


Source
+ SourceId: int
+ Username: string
+ Type: string
+ NoteType: SourceType
+ Name: string
+ Description: string
+ IsLink: bool
+ Link: string?
+ Content: byte[]?
+ CreatedAt: DateTime
+ UpdatedAt: DateTime?
+ AuthorsString: string?
+ Authors: List<string>?
+ Publisher: string?
+ AccessedAt: DateTime?
SourceType
Pdf = 1
Vid = 2
1
Use
SourceDal
- BaseUrl: string
- client: IHttpClientWrapper
+ GetSourcesForUser(User) : Task<IEnumerable<Source>>
+ AddSourceForUser(User, Source) : Task<bool>
- BaseUrl: string
- client: IHttpClientWrapper
+ GetUserSourceNotes(User,
+ createNewNote(User, Note)
+ UpdateNote(User, Note) : Ta
+ DeleteNote(Note) : Task<boo
1
<<Interface>>
IHttpClientWrapper
+ BaseAddress: Uri
+ DefaultRequestHeaders: HttpRequestHeaders
+ GetAsync(string) : Task<HttpResponseMessage>
+ PostAsync(string, HttpContent) : Task<HttpResponseMessage>
+ PutAsync(string, HttpContent) : Task<HttpResponseMessage>
+ DeleteAsync(string) : Task<HttpResponseMessage>
+ PostAsJsonAsync(string, User) : Task<HttpResponseMessage>
Http
+ BaseAddress: Uri
+ DefaultRequestHeaders: Http
+ _httpClient: HttpClient
+ GetAsync(string) : Task<Http
+ PostAsync(string, HttpConter
+ PutAsync(string, HttpContent
+ DeleteAsync(string) : Task<H
+ PostAsJsonAsync(string, Use
+ Dispose() : void


Note
NoteId: int
SourceId: int
Username: string
NoteText: string
TagsString: string
Tags: List<string>
NoteDate: DateTime
Use
NotesDal
Source) : Task<IEnumerable<Note>>
: Task<bool>
sk<bool>
ol>
1
User
+ Username: string
+ Password: string?
+ Token: string?
Use
UserDal
- BaseUrl: string
- client: IHttpClientWrapper
+ Login(User) : Task<User>
1
ClientWrapper
pRequestHeaders
ResponseMessage>
nt) : Task<HttpResponseMessage>
t) : Task<HttpResponseMessage>
ttpResponseMessage>
er) : Task<HttpResponseMessage>
IDisposable