


readme.md for @tsclass/tsclass

A comprehensive, domain-driven TypeScript type library that gives your projects consistent, battle-tested interfaces for business logic, finance, networking, content management, and more — all in one place. 

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.

Install

```
pnpm install @tsclass/tsclass
```

or

```
npm install @tsclass/tsclass
```

Module Overview

`@tsclass/tsclass` is organized into **12 domain modules**, each focusing on a specific area. Import only what you need:

```
import {  
  business,  
  finance,
```

```
network,  
content,  
container,  
code,  
database,  
descriptors,  
events,  
general,  
saas,  
storage,  
website,  
} from '@tsclass/tsclass';
```

Additionally, the library re-exports the entire [type-fest](#) collection plus a few custom utility types:

```
import type { typeFest, typeFestOwn } from '@tsclass/tsclass';
```

☐ Modules in Detail

☐ `business` — Contacts, Contracts, Documents & More

The business module models real-world entities like people, companies, addresses, contracts, letters, and projects.

Contacts

Type	Description
<code>TContact</code>	Discriminated union of <code>TPerson</code> <code>TCompany</code>
<code>TPerson</code>	A person with name, salutation, sex, optional legal proxy
<code>TCompany</code>	A company with registration details and lifecycle status (<code>planned</code> → <code>founding</code> → <code>active</code> → <code>liquidation</code> → <code>closed</code>)
<code>IAddress</code>	Physical address with optional GPS coordinates
<code>TSocialLinks</code>	Social media link (<code>facebook</code> , <code>twitter</code> , <code>linkedin</code> , or custom)

Type	Description
TRegistrationDetails	VAT ID, registration ID, and registration name

```
import { business } from '@tsclass/tsclass';

const company: business.TCompany = {
  type: 'company',
  name: 'Acme Corp',
  address: {
    streetName: 'Innovation Blvd',
    houseNumber: '42',
    postalCode: '28199',
    city: 'Bremen',
    country: 'Germany',
    countryCode: 'DE',
  },
  description: 'Builds amazing things',
  registrationDetails: {
    vatId: 'DE123456789',
    registrationId: 'HRB 12345',
    registrationName: 'Amtsgericht Bremen',
  },
  status: 'active',
  foundedDate: { day: 1, month: 6, year: 2020 },
};
```

Documents & Letters

Type	Description
TDocumentEnvelope<TYPE, FIELDS>	Base generic for all documents — includes <code>id</code> , <code>versionInfo</code> , <code>language</code> , <code>pdf</code> , <code>objectActions</code>
TLetterEnvelope<TYPE, FIELDS>	Extends document envelope with <code>from</code> , <code>to</code> , <code>subject</code> , and optional cover sheet
TLetterSimple	A simple letter
TLetter	Union of <code>TLetterSimple</code> <code>finance.TInvoice</code>
IPdf	PDF with <code>name</code> , <code>id</code> , <code>metadata</code> , and <code>buffer</code> (Uint8Array)

Contracts

Type	Description
TContract	Union of all contract types
TEmploymentContract	Employment terms including compensation, position, probation
TNDAContract	Non-disclosure agreement with confidentiality terms
TServiceContract	Service agreement with scope, deliverables, payment terms
TLeaseContract	Property lease with address, rent, and deposit
TContractParty	A party to a contract, with signing order and signature status
TContractParagraph	Individual contract paragraph/clause

Jobs & Projects

Type	Description
IJob	Full job posting with salary, requirements, languages, and scraping history
ISalaryInfo	Salary range with currency and period
IProject	Software/business project with domain, social links, and tags

finance — Invoices, Payments, Currencies & Stock Prices

Everything money-related: invoices, credit notes, bank accounts, expenses, and market data.

Accounting Documents

Type	Description
TAccountingDoc	Union of TInvoice TCreditNote TDebitNote TSelfBilledInvoice
TInvoice	Standard invoice
TCreditNote	Credit note
TDebitNote	Debit note
TSelfBilledInvoice	Self-billed invoice
TAccountingDocItem	Line item with position, quantity, net price, VAT

Type	Description
TAccountingDocStatus	'draft' 'issued' 'paid' 'canceled' 'refunded'

```
import { finance } from '@tsclass/tsclass';

const item: finance.TAccountingDocItem = {
  position: 1,
  name: 'Consulting Services',
  unitType: 'hour',
  unitQuantity: 40,
  unitNetPrice: 150,
  vatPercentage: 19,
};
```

Payments & Banking

Type	Description
ISepaConnection	SEPA bank details (IBAN, BIC)
IPayPalConnection	PayPal email connection
IPaymentOptionInfo	Available payment methods
ICheckingAccount	Bank account with currency and transactions
ITransaction	A financial transaction

Other Finance Types

Type	Description
TCurrency	Union of ~170 ISO 4217 currency codes (including crypto: BTC, ETH, ADA, etc.)
IStockPrice	Real-time stock quote with market state, exchange info, and provider
IExpenseItem	Expense line item
IVoucher	Expense voucher/receipt

```
import { finance } from '@tsclass/tsclass';

const stock: finance.IStockPrice = {
  ticker: 'AAPL',
  price: 195.83,
```

```

currency: 'USD',
change: 2.47,
changePercent: 1.28,
previousClose: 193.36,
timestamp: new Date(),
provider: 'Yahoo Finance',
marketState: 'REGULAR',
exchange: 'NASDAQ',
exchangeName: 'NASDAQ Global Select',
};

```

network — DNS, Certificates, SSH, JWT & HTTP

Types for infrastructure, networking, and web request/response handling.

Type	Description
<code>ICert</code>	SSL/TLS certificate with keys and CSR
<code>IDnsRecord</code>	DNS record with type, value, and DNSSEC status
<code>TDnsRecordType</code>	All common DNS record types (<code>A</code> , <code>AAAA</code> , <code>CNAME</code> , <code>MX</code> , <code>TXT</code> , <code>SRV</code> , <code>CAA</code> , <code>HTTPS</code> , etc.)
<code>IDnsChallenge</code>	DNS challenge for ACME verification
<code>IConvenientDnsProvider</code>	DNS provider interface with ACME challenge methods
<code>IDomainDelegation</code>	Parsed domain components (subdomain, domain, public suffix)
<code>ICnameDescriptor</code>	CNAME proxy configuration with optional SSL
<code>IRreverseProxyConfig</code>	Reverse proxy config with authentication
<code>ISshKey</code>	SSH key pair
<code>IJwtKeypair</code>	JWT signing key pair (PEM format)
<code>INetworkNode</code>	Network node with IPv4/IPv6 and security level
<code>IDevice</code>	Network device with resolution and manufacturer
<code>ISimpleRequest</code> / <code>ISimpleResponse</code>	Simplified HTTP request/response
<code>IRenderFunction</code>	<code>(req: ISimpleRequest) => Promise<ISimpleResponse></code>

```
import { network } from '@tsclass/tsclass';

const record: network.IDnsRecord = {
  name: 'example.com',
  type: 'A',
  value: '93.184.216.34',
  dnsSecEnabled: true,
};
```

☐ authentication — Claims & JWT Processing

Type	Description
<code>IClaim</code>	JWT payload with <code>userName</code> , <code>userId</code> , <code>validUntil</code> , <code>roles</code>
<code>IProcessedClaim</code>	Verified JWT with original token, decoded claim, and <code>signed</code> flag

☐ container — Docker & OCI Containers

Type	Description
<code>IContainer</code>	Container image reference (registry, tag, version, labels)
<code>IVolumeMount</code>	Host-to-container volume mapping

☐ content — Articles, Authors & Documentation

Type	Description
<code>IArticle<T></code>	Content article with title, body, tags, optional featured image, and generic reference object
<code>IAuthor</code>	Author with name, birthday, and article list
<code>IDocumentationSet</code>	Collection of articles forming a documentation set

database — MongoDB, Object Lifecycle & Wrappers

Type	Description
<code>IMongoDescriptor</code>	MongoDB connection string and credentials
<code>IObjectAction</code>	Audit trail entry: timestamp, userId, action name, message
<code>IObjectStatus</code>	Object lifecycle status (<code>active</code> <code>inactive</code> <code>hidden</code> <code>markedForDeletion</code>) with scheduled deletion
<code>IWrappedData<T></code>	Generic data wrapper

descriptors — Service Connection Descriptors

A convenience aggregation module that collects connection descriptors from across the library, plus adds its own.

Type	Description
<code>IMongoDescriptor</code>	Re-exported from <code>database</code>
<code>IStorageDescriptor</code>	Re-exported from <code>storage</code>
<code>IS3Descriptor</code>	Deprecated alias for <code>IStorageDescriptor</code>
<code>ICloudflareDescriptor</code>	Cloudflare API auth token
<code>ILetterExpressDescriptor</code>	LetterExpress API credentials

events — Domain Event Types

Type	Description
<code>IEvent_Contract</code>	Contract/subscription event
<code>IEvent_Money</code>	Monetary transfer event (SEPA, credit card, PayPal, Cardano)
<code>IEvent_Release</code>	Software release event (npm, Docker, App Store, Google Play)
<code>IEvent_Request</code>	HTTP request event with duration

Type	Description
<code>IEvent_Session</code>	User session event (new or reactivated)
<code>IEvent_User</code>	User event placeholder

general — Foundational Types

Type	Description
<code>IDate</code>	Simple calendar date (<code>day</code> , <code>month</code> , <code>year</code>) — used across many modules
<code>ITime</code>	Time representation (placeholder for extension)

saas — Products & Subscription Plans

Type	Description
<code>IProduct</code>	SaaS product with features, landing page, and publisher
<code>IProductFeature</code>	Product feature block with heading and link
<code>ISubscriptionPlan</code>	Subscription plan with pricing terms, units, and contained products

storage — S3-Compatible Storage

Type	Description
<code>IStorageDescriptor</code>	S3-compatible storage connection (endpoint, credentials, bucket, region)
<code>IS3Descriptor</code>	⚠ Depreciated — use <code>IStorageDescriptor</code>

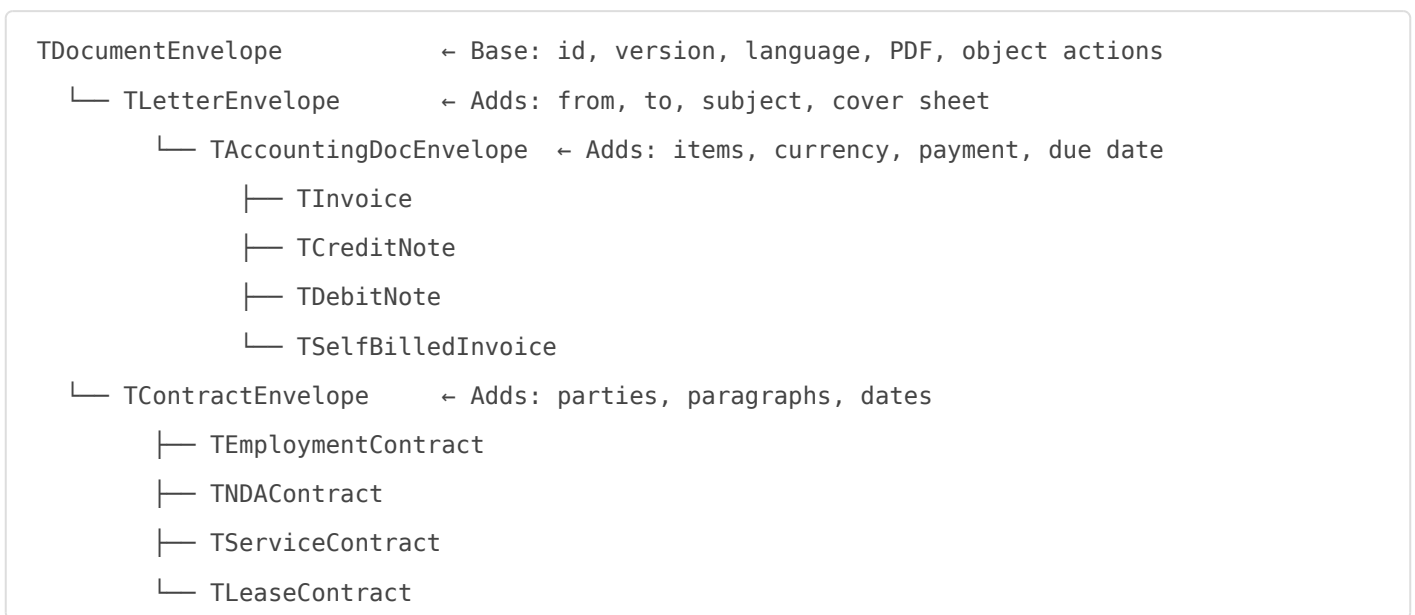
website — Links, Menus & Navigation

Type	Description
<code>ILink</code>	Hyperlink with URL, text, icon, and target
<code>ILinkBlock</code>	Titled group of links

Type	Description
<code>IMenuItem<T, U></code>	Menu item with async action callback

Architecture: The Document Hierarchy

One of the most powerful aspects of `@tsclass/tsclass` is its layered **document type hierarchy**. Documents build on each other using generic envelope types:



This means an `TInvoice` automatically has all the fields of a letter (sender, recipient, subject) and a document (version info, PDF attachment, audit trail). No duplication, full consistency. `□`

Utility Types

The library also re-exports [type-fest](#) in its entirety plus two custom utility types:

Type	Description
<code>typeFestOwn.SecondArgument<T></code>	Extracts the second argument type from a function
<code>typeFestOwn.ValueType<T></code>	Extracts the value type from an object type

```
import type { typeFest, typeFestOwn } from '@tsclass/tsclass';

type MyPartial = typeFest.PartialDeep<SomeComplexType>;
type Val = typeFestOwn.ValueType<{ a: number; b: string }>; // number | string
```

License and Legal Information

This repository contains open-source code licensed under the MIT License. A copy of the license can be found in the [LICENSE](#) file.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH or third parties, and are not included within the scope of the MIT license granted herein.

Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

Revision #3

Created 2026-03-28 11:15:10 UTC by foss.global Team

Updated 2026-03-28 12:21:53 UTC by foss.global Team