


@tsclass/tsclass

Documentation for @tsclass/tsclass

- [readme.md for @tsclass/tsclass](#)
- [changelog.md for @tsclass/tsclass](#)

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

Contracts

Type	Description
<code>TContract</code>	Union of all contract types

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

Jobs & Projects

Type	Description
<code>IJob</code>	Full job posting with salary, requirements, languages, and scraping history
<code>ISalaryInfo</code>	Salary range with currency and period
<code>IProject</code>	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
<code>TAccountingDoc</code>	Union of <code>TInvoice</code> <code>TCreditNote</code> <code>TDebitNote</code> <code>TSelfBilledInvoice</code>
<code>TInvoice</code>	Standard invoice
<code>TCreditNote</code>	Credit note
<code>TDebitNote</code>	Debit note
<code>TSelfBilledInvoice</code>	Self-billed invoice
<code>TAccountingDocItem</code>	Line item with position, quantity, net price, VAT
<code>TAccountingDocStatus</code>	'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
<code>ISepaConnection</code>	SEPA bank details (IBAN, BIC)
<code>IPayPalConnection</code>	PayPal email connection
<code>IPaymentOptionInfo</code>	Available payment methods
<code>ICheckingAccount</code>	Bank account with currency and transactions
<code>ITransaction</code>	A financial transaction

Other Finance Types

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

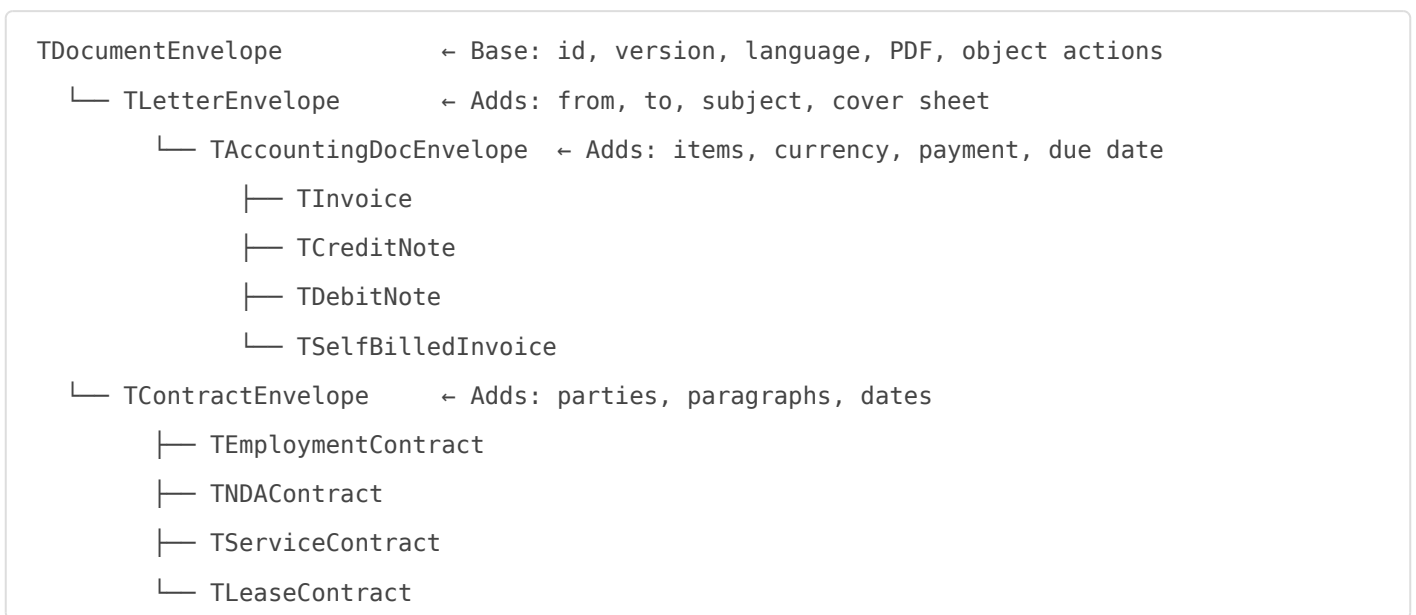
website — Links, Menus & Navigation

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.

changelog.md for @tsclass/tsclass

2026-03-14 - 9.4.0 - feat(storage)

add generic storage descriptor type and re-export it from descriptors

- introduces IStorageDescriptor with optional storage type support for s3 and smartstorage
- keeps IS3Descriptor as a deprecated alias for backward compatibility
- re-exports storage descriptor types from the descriptors module
- updates tooling dependencies and migrates tests to the new tstest tapbundle import

2025-09-23 - 9.3.0 - feat(finance)

Add IStockPrice type and export; update README and add project tooling configs

- Introduce IStockPrice interface in ts/finance/stockprice.ts (ticker, price, currency, change, changePercent, previousClose, timestamp, provider, marketState, optional exchange and exchangeName).
- Export IStockPrice from ts/finance/index.ts so it becomes part of the public finance API.
- Update README to reference concrete file paths for types and add a Stock Price usage example.
- Add project tooling configuration files: .serena/project.yml, .serena/.gitignore and .claude/settings.local.json.

2025-05-05 - 9.2.0 - feat(descriptors)

Add new descriptors module with Cloudflare and LetterExpress interfaces and documentation

- Introduced new descriptors interfaces (ICloudflareDescriptor and ILetterExpressDescriptor) in ts/descriptors/index.ts
- Added descriptors readme to document their purpose
- Integrated descriptors into the main module export in ts/index.ts

2025-04-29 - 9.1.0 - feat(network)

Add DNS convenience interface to support ACME DNS-01 challenge management in the network module.

- Added IConvenientDnsProvider interface in ts/network/dnsconvenience.ts
- Updated ts/network/index.ts to export the DNS convenience interface

2025-04-16 - 9.0.0 - BREAKING CHANGE(finance)

refactor: migrate invoice APIs to unified accounting document types

- Introduced new accounting document types in ts/finance/accountingdoc.ts to standardize invoice, credit note, debit note, and self-billed invoice representations
- Updated ts/finance/index.ts to export the new accounting document module
- Removed the legacy ts/finance/invoice.ts module

2025-04-12 - 8.2.1 - fix(business/job)

Refactor job interface to support expanded employment details and improve type safety

- Changed exported IJob from a class to an interface
- Removed obsolete properties (monthlyTotal, currency)
- Added new types for contract, work location, experience level, and job status
- Introduced detailed fields: url, location, postedDate, salary, contractType, skillTags, qualificationTags, languages, and history

2025-04-04 - 8.2.0 - feat(finance/payment)

Add optional 'description' field to payment option info.

- Enhanced the IPaymentOptionInfo interface in ts/finance/payment.ts by adding an optional 'description' property for improved context.

2025-03-26 - 8.1.1 - fix(business/letter)

Remove extraneous inline comment from TLetter type union in business/letter.ts

- Clean up redundant comment to improve code clarity
- Maintain consistency in type definitions

2025-03-26 - 8.1.0 - feat(business/letter)

Extend TLetter to include invoice support and update dependency versions

- Bump type-fest to ^4.38.0 and @types/node to ^22.13.13 in package.json
- Modify TLetter type in ts/business/letter.ts to support finance.TInvoice alongside simple letters

2025-03-25 - 8.0.4 - fix(license/business-letter)

Remove legacy commented-out code from letter module and update copyright header

- Removed unused commented block in ts/business/letter.ts
- Updated copyright header in license file

2025-03-24 - 8.0.3 - fix(ci)

Add Gitea workflow files for CI/CD, update repository URLs, and standardize formatting

- Added .gitea/workflows/default_nottags.yaml and default_tags.yaml to manage CI for tagged and non-tagged pushes
- Updated package.json repository, homepage, and bugs URLs to reflect the new code.foss.global location
- Modified .gitignore custom section comment for clarity
- Standardized code formatting and string quoting across various source files
- Enhanced tsconfig.json by adding baseUrl and paths settings

2025-03-24 - 8.0.2 - fix(commitinfo)

Refresh autogenerated commit info and internal metadata for version consistency

- No functional changes were made
- Updated internal commit metadata to reflect the current project state

2025-03-24 - 8.0.1 - fix(documentation)

Improve documentation clarity and update package metadata

- Correct author field in package.json to 'Task Venture Capital GmbH'
- Enhance readme.md with comprehensive usage examples and detailed type structure for better guidance

2025-03-24 - 8.0.0 - BREAKING CHANGE(tsclass)

Trigger patch release with no code changes

2025-03-20 - 7.1.1 - fix(finance)

Add optional date property to the Invoice interface

- Introduced an optional 'date' field with explanatory comments to capture the invoice date if not provided from the encompassing letter

2025-03-20 - 7.1.0 - feat(ts/finance/invoice)

Add electronicAddress field to Invoice for circular xinvoice support

- Introduce optional electronicAddress property with scheme and value
- Enhance documentation for buyer reference in invoice interface

2025-03-20 - 7.0.0 - BREAKING CHANGE(business/address)

Remove eAddress property from IAddress interface

- Removed eAddress field that allowed storage of electronic contact means (phone, email, peppolId)

2025-03-20 - 6.1.2 - fix(business/address)

Improve documentation comments for address interface properties

- Added detailed comments for the 'coordinates' property to explain its usage
- Added detailed comments for the 'eAddress' property to describe the storage of electronic contact information

2025-03-20 - 6.1.1 - fix(finance)

Add clarifying comment for the buyerReference field in the invoice interface

- Document buyerReference as an optional field to help buyers identify the invoice

2025-03-20 - 6.1.0 - feat(business/address)

Add optional countryCode and coordinates properties to IAddress interface

- Extend address interface with an optional countryCode property
- Add optional coordinates object with lat and lng for geographic metadata

2025-03-19 - 6.0.1 - fix(business)

Remove exports for 'company' and 'person' modules from the business index

- Removed export statement for './company.js' in ts/business/index.ts
- Removed export statement for './person.js' in ts/business/index.ts

2025-03-19 - 6.0.0 - BREAKING CHANGE(TContact)

Reaffirm project metadata and documentation consistency

- Verified commitinfo data, package.json, and README content for correctness
- No functional code changes were introduced

2025-03-11 - 5.0.0 - BREAKING CHANGE(network)

Update reverse proxy configuration to support multiple destination IPs and ports

- Replaced 'destinationIp' and 'destinationPort' with 'destinationIps' (array) and 'destinationPorts' (array) to enable multiple destinations

2025-03-10 - 4.4.4 - fix(business)

Fixes typo in ILetter interface

- Removed unused property 'letterData' from ILetter content structure.

2025-03-07 - 4.4.3 - fix(network)

Fix destinationPort type for IReverseProxyConfig

- Corrected the type of destinationPort in IReverseProxyConfig from 'string' to 'number'.

2025-03-07 - 4.4.2 - fix(dependencies)

Updated package dependencies for improved stability and performance.

- Updated 'type-fest' dependency to version ^4.37.0.
- Updated '@git.zone/tsbuild' to version ^2.2.6.
- Updated '@git.zone/tstest' to version ^1.0.96.
- Updated '@push.rocks/tapbundle' to ^5.5.6.
- Updated '@types/node' to ^22.13.9.

2025-03-07 - 4.4.1 - fix(business)

Fix missing letterData property in ILetter interface.

- Added missing letterData property to the content structure of the ILetter interface.
- Ensured consistency and completeness of ILetter interface for proper data handling of letter components.

2025-01-11 - 4.4.0 - feat(website)

Add ILinkBlock interface for managing groups of links

- Introduced a new interface ILinkBlock in ts/website/linkblock.ts for managing grouped links.
- Exported ILinkBlock from ts/website/index.ts to make it available for external usage.

2025-01-10 - 4.3.0 - feat(website)

Added new ILink interface to the website module.

- Introduced ILink interface with properties like url, text, description, icon, and target.
- Updated ts/website/index.ts to export the new link interface.

2024-12-11 - 4.2.0 - feat(dependencies)

Update dependencies and enhance code documentation

- Updated 'type-fest' dependency from ^4.20.1 to ^4.30.0 for improved type definitions.
- Updated development dependencies for better development experience.
- Enhanced code documentation for the ICommitInfo interface with detailed JSDoc comments.

2024-07-26 - 4.1.2 - fix(finance)

Updated Invoice interface to support various currencies.

- Changed the 'currency' field in the Invoice interface from fixed 'EUR' to type TCurrency.

2024-07-26 - 4.1.1 - fix(finance)

Fix Invoice interface to correctly include the currency field

- Corrected the Invoice interface definition in `ts/finance/invoice.ts`
- Moved `position` field to the top of the InvoiceItem interface

2024-07-26 - 4.1.0 - feat(finance)

Added notes field to Invoice interface

- Expanded Invoice interface to include notes field allowing additional notes for invoices.

2024-07-24 - 4.0.65 - fix(finance)

Added 'type' property to Invoice interface

- Added 'type' property to the Invoice interface in the finance domain.
- This property indicates whether the invoice is a 'creditnote' or 'debitnote'.

2024-07-18 - 4.0.64 - fix(business)

Add relationship field to contact interface

- Added 'relationship' field to the IContact interface in ts/business/contact.ts

2024-06-28 - 4.0.63 - fix(saas)

Refactor IProduct to separate IProductFeature interface

- Updated @types/node from 20.14.8 to 20.14.9
- Refactored IProduct's features property to a standalone IProductFeature interface in saas/product.ts

2024-06-24 - 4.0.62 - fix(dependencies)

Updated dependencies to latest versions

- Updated 'dependencies > type-fest' to version ^4.20.1
- Updated 'devDependencies > @git.zone/tsbuild' to version ^2.1.82
- Updated 'devDependencies > @git.zone/tsrun' to version ^1.2.49
- Updated 'devDependencies > @git.zone/tstest' to version ^1.0.90
- Updated 'devDependencies > @push.rocks/tapbundle' to version ^5.0.23
- Updated 'devDependencies > @types/node' to version ^20.14.8

2024-06-24 - 4.0.61 - fix(dependencies)

Update dependencies to latest versions

- Updated 'dependencies > type-fest' to version ^4.20.1
- Updated 'devDependencies > @git.zone/tsbuild' to version ^2.1.82
- Updated 'devDependencies > @git.zone/tsrun' to version ^1.2.49
- Updated 'devDependencies > @git.zone/tstest' to version ^1.0.90
- Updated 'devDependencies > @push.rocks/tapbundle' to version ^5.0.23
- Updated 'devDependencies > @types/node' to version ^20.14.8

2024-06-19 - 4.0.59 - core

Fixes and updates to core functionalities.

- Multiple updates and fixes applied to the core codebase

2024-06-17 - 4.0.59 to 4.0.56 - core

Routine updates and fixes to the core.

- Multiple core updates and fixes

2024-06-08 - 4.0.54 - core

Core fixes applied.

- Various fixes in the core functionality

2024-04-17 - 4.0.53 - core

Core functionality updated with fixes.

- Updates and fixes to core features

2024-04-16 - 4.0.52 - documentation

Documentation update.

- Documentation improved and updated

2024-02-17 - 4.0.51 - core

Fixes for core applied.

- Multiple core fixes implemented

2024-02-10 - 4.0.48 - core

Routine core updates and fixes.

- Several updates and fixes for core functionality

2023-10-15 - 4.0.46 - core

Core updates.

- Core functionalities updated and fixes applied

2023-10-13 - 4.0.44 - core

Routine fixes for core functionalities.

- Various updates to core functionalities

2023-09-13 - 4.0.43 - core

Core updates and fixes.

- Several core updates and fixes

2023-05-08 - 4.0.41 - core

Core updates.

- Core functionalities updated

2023-05-06 - 4.0.39 - core

Core updates and fixes.

- Multiple core updates and fixes

2023-04-19 - 4.0.37 - core

Routine core updates.

- Core fixes and updates applied

2023-04-05 - 4.0.35 - core

Core functionalities updated.

- Core functionalities updated and fixes applied

2023-03-14 - 4.0.33 - core

Routine core fixes and updates.

- Core functionalities updated

2023-03-09 - 4.0.32 - core

Routine maintenance for core.

- Core updates and fixes

2023-02-15 - 4.0.31 - core

Core updates applied.

- Fixes and updates to core functionalities

2023-02-06 - 4.0.30 - core

Routine fixes.

- Multiple core updates and fixes

2023-01-12 - 4.0.29 - core

Core updates applied.

- Fixes and updates to core functionalities

2022-11-20 - 4.0.27 - core

Routine core fixes.

- Core functionalities updated and fixed

2022-10-31 - 4.0.26 - core

Routine maintenance and fixes.

- Core functionalities updated

2022-10-24 - 4.0.24 - business.IContract

Added business logic for contracts.

- Added IContract logic to business module

2022-09-14 - 4.0.21 - core

Routine core updates and fixes.

- Core functionalities updated and fixed

2022-08-17 - 4.0.18 - core

Routine maintenance for core.

- Core functionalities updated and fixes applied

2022-06-16 - 4.0.17 - core

Routine fixes.

- Core functionalities updated

2022-04-28 - 4.0.0 - core

Major update with significant core changes.

- Switch to ESM format
- Multiple core functionalities updated and fixed