

URUTO Product Description

製品紹介資料



Cony&Toad
株式会社



Agenda

1. URUTOについて
2. 機能一覧
3. ログイン
4. ユーザー設定
5. 項目属性
6. インスタンス
7. テーブル
8. プロシージャ
9. プロダクト
10. 機能
11. レイアウト
12. レイアウト項目
13. トリガー
14. ドキュメント
15. 例外

1. *URUTO*について

本製品は、.NET MAUI Blazor Hybrid と Radzen を基盤に開発された、あらゆる分野のWebシステム・アプリケーションを短期間で構築できる開発プラットフォームです。



- ・ECサイト、マッチングサイト、予約システム、販売管理など、基本構成を備えたテンプレートをコピーするだけで開発を開始できます。

- ・PC・スマートフォン・タブレットなど、あらゆるデバイスで最適に動作。

.NET MAUI Hybrid により、Web・デスクトップ・モバイルアプリを共通コードで展開可能。

- ・生成AIによる自動翻訳で、30言語以上の多言語環境に対応。

海外チームとのオフショア開発にも最適です。

- ・Radzen のコンポーネントを活用し、UIのカスタマイズや機能追加が容易。

管理画面、マイページ、メッセージ機能などの標準機能も利用可能です。

2. 機能一覧

本プラットフォームは、アプリケーション開発・運用に必要な要素を統合的に管理できる設計となっています。
各構成要素が連携し、効率的な開発・運用・拡張を実現します。



- ログイン
- ユーザー設定
- 項目属性
- インスタンス
- テーブル
- プロシージャ
- プロダクト
- 機能
- レイアウト
- レイアウト項目
- トリガー
- ドキュメント
- 例外

3. ログイン

ご契約が完了すると、ログインに必要な情報をお送りします。
受け取った情報でログイン後、「ユーザー設定」画面からパスワードなどの
ログイン情報を変更することができます。
セキュリティを高めるため、初回ログイン後はお早めの変更をおすすめします。



会社コード、ユーザーID、パスワードを入力してログインをしてください。
ログインが成功すると会社コードとユーザーIDが表示されます。

パスワードを忘れた場合は、パスワード忘れの画面に遷移してメールアドレスを入力
して再設定をお願いします。

4. ユーザー設定

本システムでは、ユーザーごとに認証と権限を設定できます。

権限レベルに応じて、利用できる操作範囲を制限することができます。

また、利用ユーザー数に制限はありません。

複数の開発者が開発内容を共有しながら、役割に応じた安全な運用が行えます。



- 権限レベルに応じて、操作範囲を制限することができます。
- 参照ユーザー：登録内容の閲覧のみ可能
- 更新ユーザー：登録・編集・削除などの操作が可能
- また、ChatGPT や Gemini などの生成AIとの連携にも対応しています。
ユーザー設定画面から各AIの API情報を登録することで、システム内で自動生成・翻訳・要約などのAI機能を活用できます。

5. 項目属性

システム内で使用される物理名称と論理名称を紐づけて管理できます。
開発者と業務担当者の間で名称の統一と理解の共有が可能になります。
また、名称定義を一元管理することで、システムの拡張や保守もスムーズに行えます。



・物理名称

データベース上で使用される実際の項目名（例：user_id, product_name など）

・論理名称

画面表示やドキュメントで使用する分かりやすい名称（例：「ユーザーID」「商品名」など）

・生成AIを活用し、最大30カ国語の多言語名称を自動生成・編集可能です。

これにより、グローバル展開を視野に入れたプロダクト開発や、オフショア開発時の名称共有をスムーズに行えます。

6. インスタンス

システム内で利用するデータ構造やオブジェクトの定義を管理します。
また、既存データベースのDDL（データ定義言語）を取り込むことで、
リバースエンジニアリングにも対応しています。

既存システムの構造を自動的に反映し、再利用や改修をスムーズに行うことができます。



- ・データ構造・オブジェクトを統合管理
- ・DDL取込によるリバースエンジニアリング対応
- ・SQL文・データ定義・アクセスプログラムを自動生成
- ・項目属性・レイアウトと連携し、一貫性ある設計を実現

7. テーブル

システム内で利用するデータベーステーブルの構造や属性を定義・管理します。

定義したテーブル情報をもとに、DDLスクリプト・SQL文・プロシージャ・データアクセス用ソースコードを自動生成できます。

コーディング作業を大幅に削減し、開発から運用まで一貫した構成を実現します。



- ・データベース構造・属性を定義・管理
- ・DDL・SQL・プロシージャ・データアクセスコードを自動生成
- ・Radzen向けの入力フォームやリスト項目も自動生成
- ・データ区分値の定義

8. プロシージャ

データベース処理（ストアドプロシージャ）を定義・管理します。
検索を利用する検索条件項目と検索結果項目を明確に定義でき、
開発者・設計者・UI担当者間で検索仕様の認識を統一することが可能です。



- ・検索（サーチ）系に特化したプロシージャ管理
- ・検索条件項目・検索結果項目を定義して仕様を統一
- ・JOIN補助機能で複数テーブルを簡単に結合
- ・Radzenで検索フォーム・検索結果リストを自動生成

9. プロダクト

開発対象となるシステムやアプリケーションの単位（プロジェクト単位）を定義・管理します。複数の機能・データベース・レイアウトなどを包含し、開発・運用・テストを統合的に行うための基本単位として扱われます。



- ・ プロダクト（プロジェクト）単位で開発要素を一元管理
- ・ 顧客・仕入先・職員のベースプロダクトをコピーしてすぐに開発可能
- ・ 既存製品をコピーして別プロダクトを効率的に展開
- ・ マークダウン形式でプロダクト概要を記載

9. プロダクト（ベースについて）

顧客ベース

ECサイトや会員制システムの基盤として利用できる標準構成です。
顧客が自ら情報を確認・操作できるマイページ機能と商品やサービスの注文を管理する受注受付機能を中心に構成されています。

仕入先ベース

取引先やパートナー企業との連携を効率化するための標準構成です。
主に、取引情報を管理する受注管理機能と、取扱商品を一元管理できる商品管理機能を中心として構成されています。

職員ベース

職員ベース製品は、社内業務全般を統合的に管理できる標準構成です。
日々の運用で必要となる各種マスタ管理・商品管理・受注管理・会計管理機能を中心に構成されています。

10. 機能

製品を構成するアプリケーション単位の処理や画面のまとめを定義します。

1つの機能は、画面・テーブル・プロシージャ・レイアウトなどの要素と紐づき、業務の流れに沿ったアプリケーション動作を構築するための基本単位となります。



- ・機能単位でアプリの構造・処理・画面を定義
- ・トリガーを紐づけ、動作をイベントベースで制御可能
- ・利用テーブルの紐づけ
- ・機能をまとめてプログラムソースの生成
- ・マークダウンでの機能説明

11. レイアウト

システム内で使用される画面や帳票の構成を定義します。
ユーザーインターフェースの設計要素として、画面上に表示する項目・トリガーなどを
詳細に設定できます。



- ・画面および帳票レイアウトを定義・管理
- ・入力欄・ボタン・リストなどの画面項目を設定
- ・各項目にトリガーを紐づけて動作を制御
- ・帳票の出力項目・デザイン定義に対応
- ・HTMLやRadzen画面のインポートが可能

12. レイアウト項目

レイアウトを構成する個々のUI要素を定義・管理する領域です。
生成AIによって自動生成された多言語ラベルを手動で修正・調整できます。
業務に合わせてニュアンスを微調整します。



- ・画面を構成するUI要素（入力欄・ボタン・リスト等）の詳細設定
- ・生成AIによる多言語化ラベルを手動で調整可能
- ・多言語・多画面開発を効率化する中心要素

13. トリガー

画面や項目の操作に応じて自動的に処理を呼び出すイベント制御機能です。
ユーザーの操作やデータ更新のタイミングに合わせて、
機能・プロシージャ・外部APIなどの処理を実行させることができます。



- ・画面・項目の操作に応じて処理を自動実行
- ・プロシージャ・SQL・外部APIなどの呼び出しを制御
- ・クリックや値変更など多様なイベントに対応
- ・レイアウト、レイアウト項目、機能と連携して動作
- ・イベント駆動型アプリケーションを容易に構築

14. ドキュメント

プロダクト仕様を常に最新状態で出力できるため、開発の認識合わせに非常に有効です。日本語+英語、英語+ベトナム語など、任意の組み合わせを並列で生成できるため、オフショア開発における仕様の読み違いを防ぎます。



- 製品概要書
- 機能概要書
- レイアウト定義書&テスト仕様書
- トリガー定義書&テスト仕様書
- テーブル定義書
- プロシージャ定義書
- API仕様書

15. 例外

システム内で発生したすべての.NET例外をログとして自動的に保管します。
アプリケーション層、ビジネスロジック層、データアクセス層など、
記録されるため、運用時のトラブルシューティングや品質改善に役立ちます。



- ・.NETで発生したあらゆる例外を自動でログ保存
- ・発生箇所、スタックトレース、入力値、ユーザー情報など詳細を保持
- ・蓄積ログから傾向分析ができ、潜在バグの早期発見に有効
- ・運用監視や品質管理の精度が向上



是非是非
ご検討してしてください。

素晴らしいプロダクトが完成し
ますように