

CodeAGI

Ver 1.11

目次

1	本誌の目的	3
2	初期設定	4
2.1	セットアップ	4
2.1.1	Windows でのセットアップ方法	4
2.1.2	macOS でのセットアップ方法	4
2.1	アカウントの作成	5
2.2	生成 AI の設定について	6
2.2.1	生成 AI の設定	6
3	プログラム生成に利用する設計書について	10
3.1	Excel のバージョン [重要]	10
3.2	Excel のシート名 [重要]	10
4	プロジェクト情報の指定	11
4.1	プロジェクトの新規作成	11
4.2	言語、フレームワーク、パッケージの依存環境設定について	14
4.3	プロジェクトの選択、修正、削除	15
4.3.1	プロジェクトを選択する	15
4.3.2	プロジェクトを編集する	16
4.3.3	プロジェクトを削除する	16
5	設計書 (仕様書) の指定について	17
5.1	テーブル定義書を指定する	17
5.1.1	解析するテーブル定義書のイメージ	17
5.2	機能設計書を指定する	18
5.2.1	機能情報の新規作成	18
5.2.2	機能情報の変更、削除	19
5.2.3	解析する機能設計書のイメージ	20
5.3	画面イメージを指定する	20
5.4	共通仕様書を指定する	21
5.4.1	解析する共通仕様書のイメージ	21
5.5	共通部品指定	22
5.5.1	共通部品指定の編集と削除	23
5.6	指定情報の確認	24
6	プログラム生成	25
6.1	プログラム生成	25
6.2	プログラム生成画面について	26
6.3	連続プログラム生成について	26
7	生成したプログラムの確認およびファイル出力	27
7.1	生成完了プログラムの確認およびファイル出力	27
7.2	生成履歴を確認する	28

8	テストケース・データ生成	29
8.1.	テストケース・データ生成	29
8.2.	テストケース・データ生成画面について	30
8.3.	連続テストケース・テストデータ生成について	30
9	生成したテストケース・データの確認およびファイル出力	31
9.1.	生成完了テストケース・データの確認およびファイル出力	31
9.2.	生成履歴を確認する	32
10	その他	33
10.1.	複雑な SQL 複数のテーブルの結合について	33
10.2.	生成コードのファイル名などに機能 ID を使用する方法	34
10.3.	解析する設計書 (仕様書) のシート名について	35
10.4.	機能名を取得できない場合	36
10.5.	デザインの生成 a version	37
11	付録	38
11.1.	アンインストールについて	38
11.2.	GPT の登録について	38
11.3.	Google Cloud Platform Vertex AI の登録について	38
11.4.	macOS の証明書がないバージョンのインストール	39
11.5.	シート名をマニュアル通りに設定しても読み込めない場合	39

1 本誌の目的

本誌は機能設計書やテーブル定義書からプログラムやテストケース・データを自動生成する CodeAGI の使用方法を解説します。

2 初期設定

CodeAGI の初期設定を行います。

この章ではセットアップ方法並びにお客様が使用される生成 AI の選択方法及びその生成 AI の設定方法を説明します。

また本システムを使用する上で生成 AI (※) の契約をお客様ご自身で行って頂く必要がございます。

※ChatGPT または Google Cloud Platform(Vertex AI)

2.1. セットアップ

2.1.1. Windows でのセットアップ方法



Download したファイルをクリックすると左図が表示されます。

[インストール] ボタンをクリックすることでインストールが行えます。

2.1.2. macOS でのセットアップ方法



Download したファイルをクリックすると左図が表示されます。

フォルダーを指定してインストールを完了してください。

※：インストールファイルをクリックしても左図が開かない時は 11.4 「macOS の証明書がないバージョンのインストール」を参照ください。

2.1. アカウントの作成

①：セットアップが完了すると左図が表示されます。[アカウント登録] ボタンをクリックしてください。

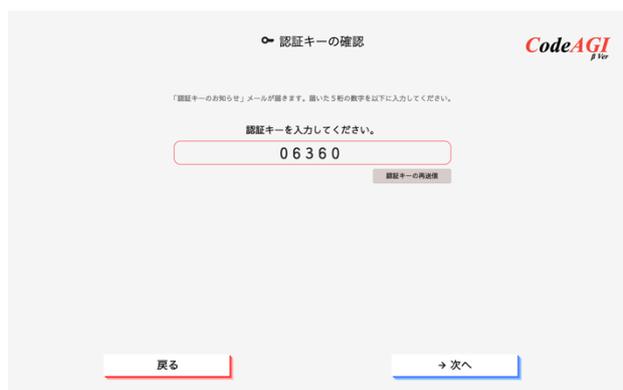
②：CodeAGI を使うための利用規約が表示されます。内容を確認して頂き [利用規約に同意して次へ] ボタンをクリックしてください。

③：それぞれの項目に添って、あなたのアカウント情報を登録してください。

メールアドレスには本人確認のためのメールが送信されますので、メールアドレスは正確に入力してください。



④：認証キーがご登録されたメールアドレスに届きます。



⑤：④で届いた認証キーを入力し [次へ] ボタンをクリックしてください。メールが届かない場合、もしくは誤って削除した場合等は、[認証キーの再送信] ボタンをクリックすることで新しい認証キーが送付されます。アカウント登録画面で [登録する] ボタンを押してから 30 分以内に認証キーを入力してください。

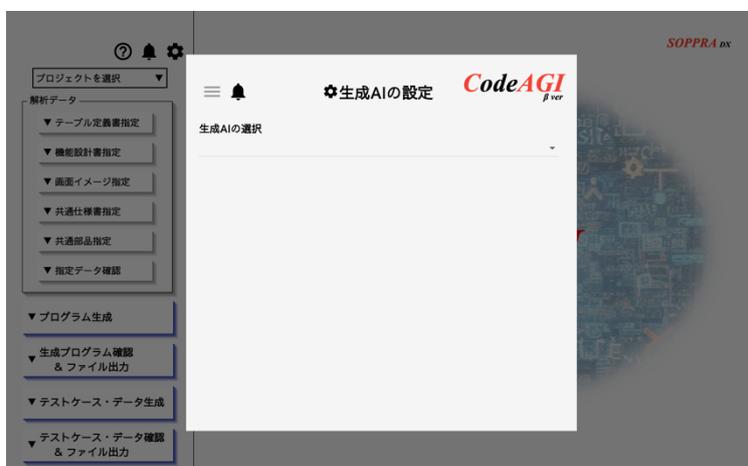


⑥：アカウントの登録が完了しました。「次回から自動でログインする」にチェックを入れることで、次回からはログイン操作をスキップして自動的にログインされます。初期状態はチェックが入っておりませんので、毎回ログインするのが面倒な場合は、チェックを入れて [次へ] ボタンをクリックしてください。

2.2. 生成 AI の設定について

CodeAGI は複数の生成 AI に対応しています。始めにあなたが使用する生成 AI を選択し、それぞれの生成 AI が使用できるように設定情報を入力してください。

2.2.1. 生成 AI の設定

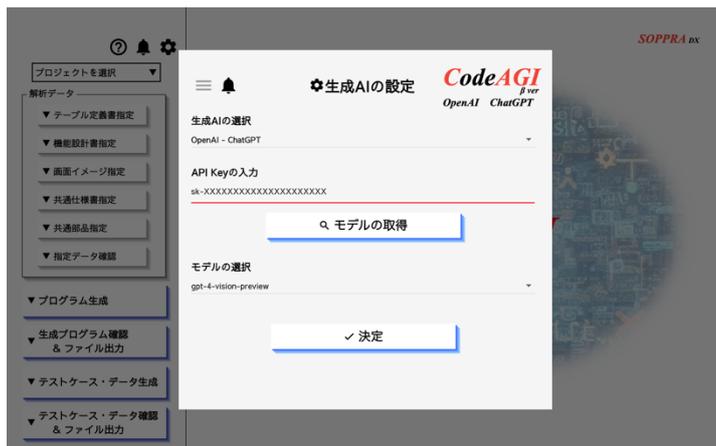


アカウント登録が完了すると左図の画面が表示されます。始めに生成 AI を選択してください。選択した生成 AI の設定画面が表示されます。なお、ここで選択する生成 AI は、お客様がご契約されている生成 AI となります。※：現在は OpenAI – ChatGPT、Azure – OpenAI、Google – Vertex AI に対応

【OpenAI – ChatGPT (以下 GPT と表現)】 を選択した場合

① : GPT を選択することで右図の画面が表示されます。始めに GPT と契約した API Key を入力してください (sk-で始まる文字列です)。

※ : API Key をお持ちでない場合は、これ以上設定を進めることができません。OpenAI のホームページより、API Key の取得を行ってから再度本画面より実施してください。取得方法については「11.2 GPT の登録について」に記載しています。



② : API Key を入力したら [モデルの取得] ボタンをクリックしてください。API Key からあなたが契約されているモデルが取得されます。取得されたモデルから gpt-4 以上の物を選択することをお勧めします。モデルに gpt-4 未満しか表示されない場合は、OpenAI との契約内容をお確かめの上、gpt-4 以上が使える契約をお勧めします。gpt-3.5 等でも動作はしますが token 量の関係でエラーになる可能性があります。

③ : [決定] ボタンをクリックすることで生成 AI の設定は完了です。

【Azure - OpenAI (以下 Azure と表現)】 を選択した場合

① : Azure を選択することで右図の画面が表示されます。始めに契約されている Azure Open AI Service から提供されている API Key とエンドポイントを入力してください。



② : 次にデプロイ名を入力してください。デプロイ名は Azure OpenAI Studio にてデプロイした際に命名したデプロイ名を入力してください。

※デプロイ名 (画像認識モデル) の入力については任意です。画面イメージ指定機能をご利用の際には画像認識モデルのデプロイ名を入力してください。

③ : [決定] ボタンをクリックすることで生成 AI の設定は完了です。

【Google – Vertex AI】を選択した場合

① : Google – Vertex AI を選択することで右図の画面が表示されます。始めに Google Cloud Platform (以下 GCP) から入手したプロジェクト ID を入力してください。

※ : サービスアカウントキーをお持ちでない場合は、これ以上設定を進めることができません。GCP のホームページより、サービスアカウントキーの取得を行ってから再度本画面より実施してください。取得方法については「11.3 Google Cloud Platform Vertex AI の登録について」に記載しています。



② : 次に GCP の API 認証情報で作成したサービスアカウントキーをダウンロードし、ダウンロードしたファイルのパスを選択してください。CodeAGI は毎回そのパスからサービスアカウントキーを取得し GCP との接続に用います。そのため、本ファイルは削除しないようにしてください。

③ : [決定] ボタンをクリックすることで生成 AI の設定は完了です。

設定した生成 AI の変更について



⚙️ をクリックすることで生成 AI 設定 / アカウント設定等が表示されます。

生成 AI 設定を選択してください。
上記で示した画面が表示されます。

3 プログラム生成に利用する設計書について

この章ではプログラム生成に用いる設計書について説明します。

3.1. Excel のバージョン[重要]

設計書は Excel 形式で作成されている必要があります。

また 2007 以降の **xlsx** 形式を対象としています。

※ : Excel2003 の xls 形式は読み込めません。

3.2. Excel のシート名[重要]

CodeAGI は設計書の判定にシート名を用いています。

テーブル定義書は“テーブル定義 (品目マスタ)”等の名前にして頂き“()”の中にはテーブル名を記載して頂く必要があります。

アプリケーションの設計書は“アクション定義”、“画面項目定義”等の名前で作成する必要があります。CodeAGI が解析できるシート名の詳細については「10.3 解析する設計書 (仕様書) のシート名について」を参照ください。

4 プロジェクト情報の指定

この章ではプロジェクトの指定に必要なそれぞれの項目について説明します。

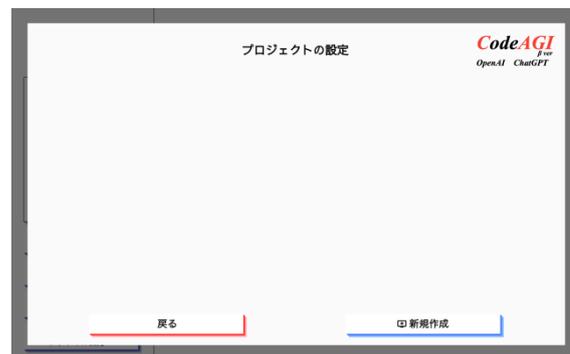
4.1. プロジェクトの新規作成



始めにプロジェクトを指定します。

左図にある「プロジェクトを選択」をクリックしてください。

これをクリックすることで、下図の画面が表示されます。



①：プロジェクトが作成されていない状態で本画面を表示すると [戻る] ボタンと [新規作成] ボタンのみが配置された画面が表示されます。
[新規作成] ボタンをクリックしてください。



②：プロジェクト情報を設定する画面が表示されます。

それぞれの項目に沿って登録してください。

「コード出力先フォルダ」及び「テストケース・データ出力先フォルダ」は、プログラムやテストケース・データ生成後にファイル出力を行う際のカレントを示します。また OS、DB、ブラウザ等は本プロジェクトで実

運用することを想定して選択してください。

また HTML 等のユーザインターフェースが存在しないプロジェクトでもブラウザの選択はしても結構ですし、しなくても結構です。

(※：[生成プログラムに画面 ID・バッチ ID を使う] については、「10.2 生成コードのファイル名などに機能 ID を使用する方法」で説明します)



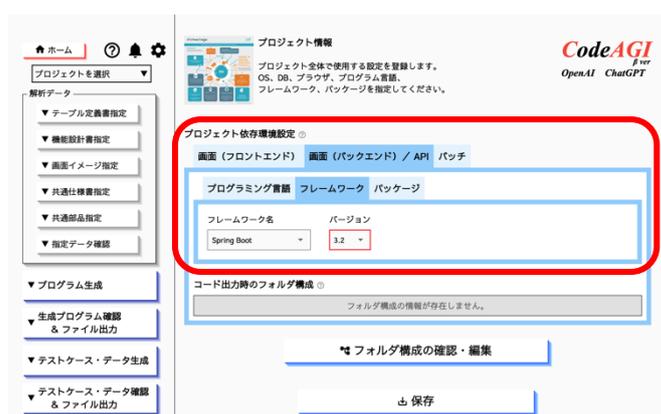
③:次にプロジェクトの依存環境設定を行います。CodeAGIを使用する上で重要なポイントの一つとなります。プロジェクトでは画面(フロントエンド)、画面(バックエンド)/API、バッチなどの機能を作成することになりますが、それらの種別のことをCodeAGIの中では「機能種別」と呼んでいます。この機能種別単位にプログラミング言語、フレームワーク及び使用する

パッケージ(ORM等)を設定する必要があるので、④以降の手順を参考にそれぞれのタブをクリックして必要な内容を設定してください。なおバッチ機能が無い場合など未使用の機能種別は設定不要です。

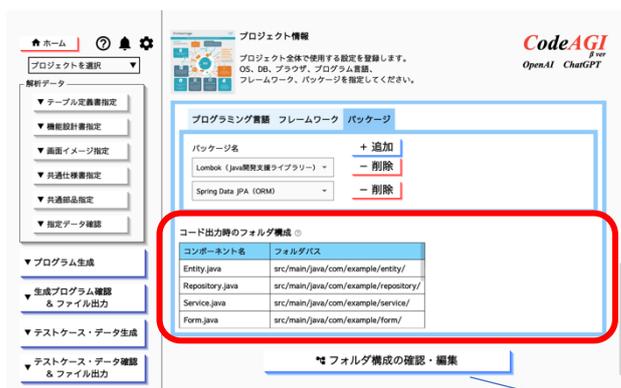
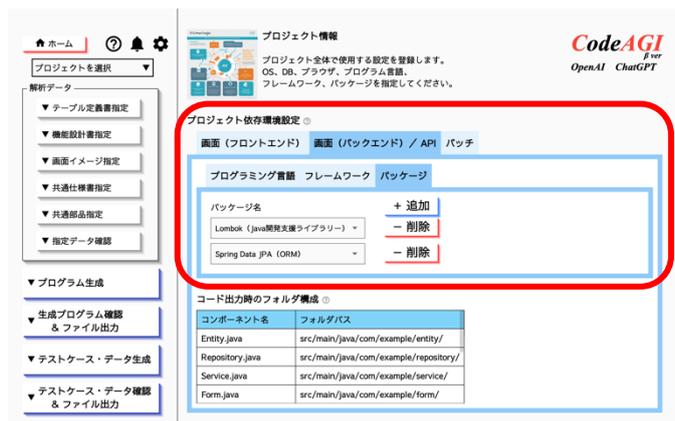
④:始めにプログラミング言語を設定します。設定する機能種別を選択の上、使用するプログラミング言語を選択し、バージョンを選択してください。バージョンが不明確な場合は空白でも結構です。



⑤:次にフレームワークを設定します。選択した機能種別とプログラミング言語に応じて選択可能なフレームワークが表示されるので、使用するフレームワークを選択してください。なお、Webブラウザ上で動作するWebアプリケーション等でフロントエンド、バックエンドで使用するパッケージが異なる場合などの設定方法については「4.2 言語、フレームワーク、パッケージの依存環境設定について」で説明します。



⑥:最後にパッケージを設定します。選択した機能種別、プログラミング言語、フレームワークに応じて選択可能なパッケージが表示されるので、UIのテンプレートやORM等を選択してください。指定方法は⑤で説明した通りです。また複数設定したい場合は必要な分[追加]ボタンにより追加してください。



⑦:プロジェクトの依存環境設定を入力することでコンポーネント単位の出力フォルダが自動で表示されます。フォルダを変更したい場合は、[フォルダ構成の確認・編集]ボタンをクリックして下図の画面を表示し、変更してください。



h

⑧:以上の設定が完了したら[保存]ボタンをクリックしてください。これでプロジェクト情報の指定は完了となります。

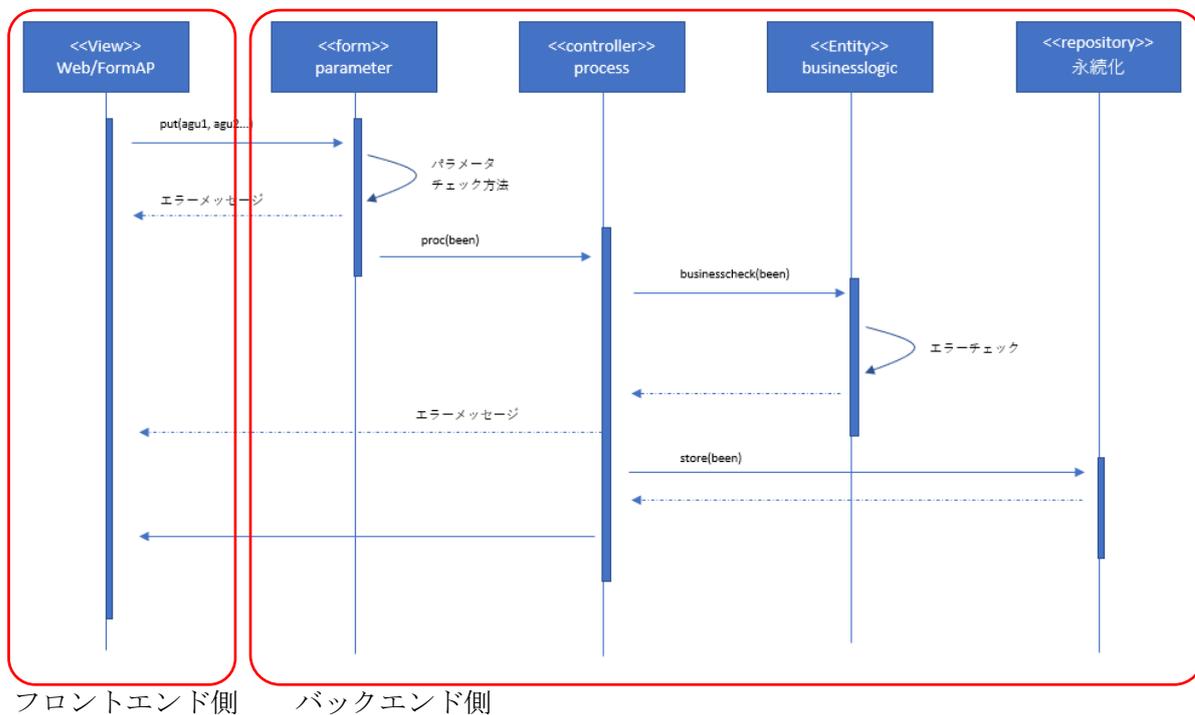
4.2. 言語、フレームワーク、パッケージの依存環境設定について

画面（フロントエンド）および画面（バックエンド）の設定内容について説明します。
例えば、言語がフロントエンド・バックエンド共に Java で、Thymeleaf + SpringBoot + JPA(ORM)を使用するプロジェクトの場合、以下のような設定を行います。

【フロントエンド】 フレームワーク : SpringBoot パッケージ : Thymeleaf

【バックエンド】 フレームワーク : SpringBoot パッケージ : JPA, Lombok

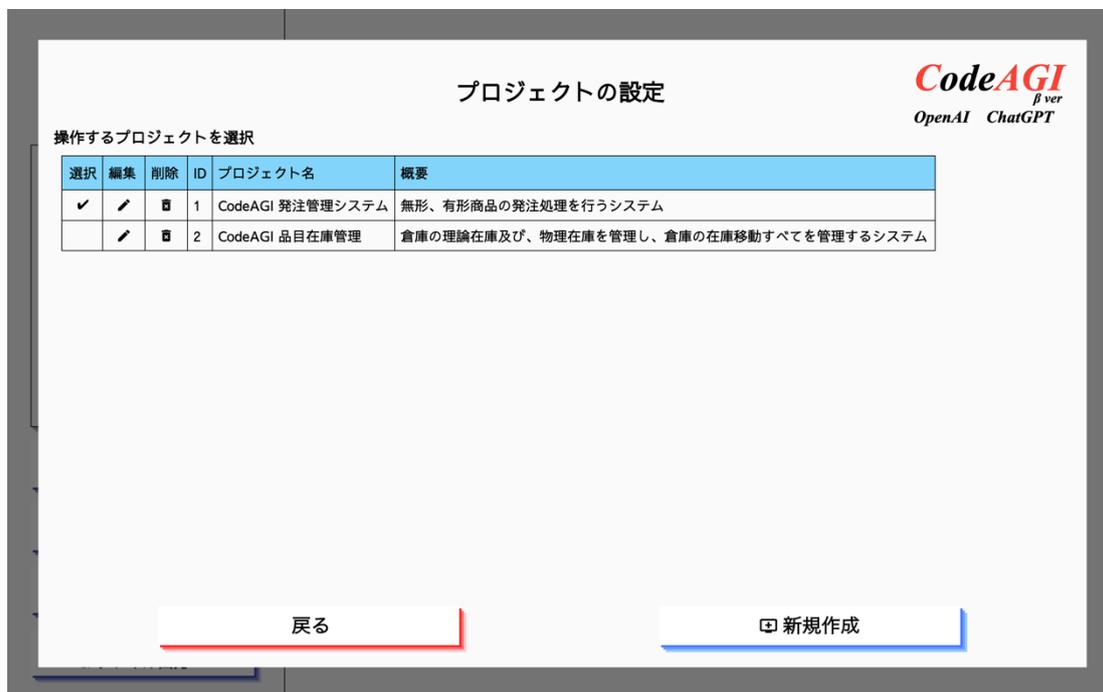
■上記開発プロジェクトにおけるアーキテクチャーのイメージ



※フロントエンドとバックエンドで使用するプログラミング言語が異なるプロジェクトの場合（フロントエンド : JavaScript、バックエンド : Java など）は、それぞれ設定してください。

4.3. プロジェクトの選択、修正、削除

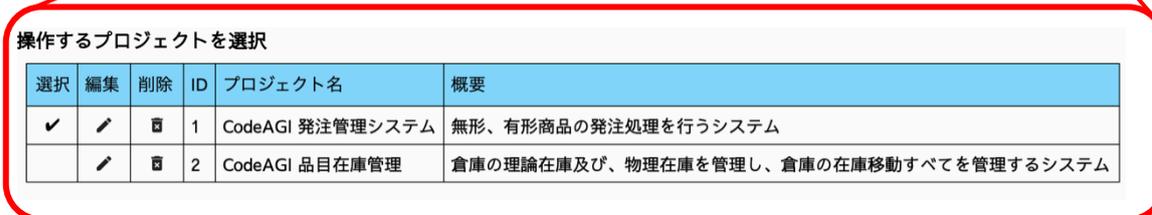
プロジェクトを複数作成すると下図のように作成したプロジェクトがスプレッドに表示されます。



4.3.1. プロジェクトを選択する



プロジェクトをクリックすることで下図の画面が表示されます。



スプレッドの行を選択することでプロジェクトが選択されます。

4.3.2. プロジェクトを編集する



操作するプロジェクトを選択

選択	編集	削除	ID	プロジェクト名	概要
<input checked="" type="checkbox"/>			1	CodeAGI 発注管理システム	無形、有形商品の発注処理を行うシステム
<input type="checkbox"/>			2	CodeAGI 品目在庫管理	倉庫の理論在庫及び、物理在庫を管理し、倉庫の在庫移動すべてを管理するシステム

編集列のペンのアイコンをクリックすることで右図の画面が表示されます。「4.1 プロジェクトの新規作成」にて説明した内容と同じ画面となります。修正を行い [保存] ボタンをクリックしてください。変更した内容が書き換わります。

また修正しない場合は [保存] ボタンをクリックせず次の作業 (別画面の表示) を行うことで変更は取り消されます。



4.3.3. プロジェクトを削除する



操作するプロジェクトを選択

選択	編集	削除	ID	プロジェクト名	概要
<input checked="" type="checkbox"/>			1	CodeAGI 発注管理システム	無形、有形商品の発注処理を行うシステム
<input type="checkbox"/>			2	CodeAGI 品目在庫管理	倉庫の理論在庫及び、物理在庫を管理し、倉庫の在庫移動すべてを管理するシステム

削除列のごみ箱アイコンをクリックすることで削除されます。プロジェクトの関連情報はもとよりプロジェクトに紐づく設計書等すべてを削除します。削除の取り消しは行えません。十分注意して使用してください。

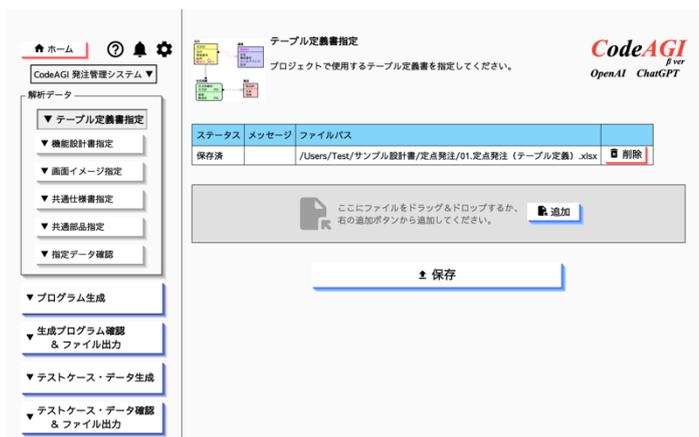
5 設計書(仕様書)の指定について

この章では Excel(xlsx 形式)で作成された様々な設計書を CodeAGI に指定する手順を示します。

5.1. テーブル定義書を指定する



[テーブル定義書指定] ボタンをクリックすることで下図が表示されます。



①:生成するプログラムで使用するテーブル定義書を追加し [保存] ボタンをクリックしてください。

②:完了後に表示されるメッセージで [確認] をクリックするか、左のメニューから [指定データ確認] ボタンをクリックし内容を確認します。

確認する手順の詳細については「5.6 指定情報の確認」を参照してください。

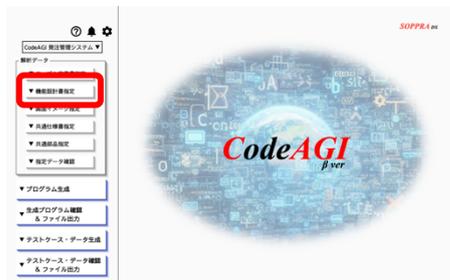
※:テーブル定義が複数の Excel ファイルにまたがっている場合は、必要なファイル分 [追加] ボタンにより追加してください。また誤って追加したファイルについては [削除] ボタンにより削除を行ってください。

5.1.1. 解析するテーブル定義書のイメージ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z AA AB AC AD AE AF AG AH AI AJ AK AL AM											
テーブル名			品目テーブル				テーブル名			ITEM	
No	論理名	物理名	データの型	桁数	PK	必須	初期値	FK	備考		
1	品目ID	ID	Number		○	○					
2	品目コード	ITEM_CODE	VarChar	32		○					
3	品目名	ITEM_NAME	VarChar	32		○					
4	品目種別	ITEM_CATEGORY_ID	VarChar	20					品目を分類するためのカテゴリや種別		
5	製造元	MANUFACTURER	VarChar	50					品目を製造した企業や工場の情報		
6	仕入先コード	SUPPLIER_CODE	Number	10		○		SUPPLIER.ID			
7	単位	UNIT_OF_MEASURE	Number	5							
8	基本価格	BASE_PRICE	Number	10							
9	安全在庫数	SAFETY_STOCK	Number	10							
10	リードタイム	LEAD_TIME	Number	10							
11	最終更新日	LAST_UPDATED_DATE	Date				sysdate				
12	最終更新者	LAST_UPDATED_BY	VarChar								

5.2. 機能設計書を指定する

5.2.1. 機能情報の新規作成



[機能設計書指定] ボタンをクリックすることで下図の画面が表示されます。

- ①: 初めての場合は [新規作成] ボタンをクリックしてください。画面に添って情報の入力を行い、機能設計書を追加し [保存] ボタンをクリックしてください。
- ②: 完了後に表示されるメッセージで [確認] をクリックするか、左のメニューから [指定データ確認] ボタンをクリックし内容を確認します。



確認する手順の詳細については「5.6 指定情報の確認」を参照してください。



左図の機能種別について、4つの選択肢が表示されます。

指定した設計書で出力する、機能種別を選択してください。

【機能種別：画面】

画面の項目定義、それぞれのアクションの設計、データベースからの取得・編集定義が記載されている設計書を指定し、プロジェクト依存環境設定に関しては、画面（フロントエンド）、画面（バックエンド）のアーキテクチャーを設定している必要があります。

【機能種別：画面（フロントエンドのみ）】

画面の項目定義、それぞれのアクションの設計、使用する API へのパラメータが記載されてい

る設計書を指定し、プロジェクト依存環境設定に関しては、画面（フロントエンド）のアーキテクチャを設定している必要があります。

【機能種別：画面（フロントエンドのみ）／API】

APIのアクション、受け取るパラメータ及び返却するパラメータ、データベースからの取得・編集定義が記載されている設計書を指定しプロジェクト依存環境設定に関しては、画面（バックエンド）／APIのアーキテクチャを設定している必要があります。

【機能種別：バッチ】

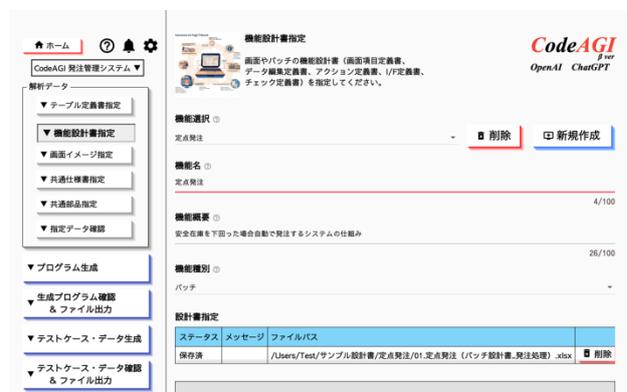
バッチプログラムの設計書を指定しプロジェクト依存環境設定に関しては、バッチのアーキテクチャを設定している必要があります。

5.2.2. 機能情報の変更、削除



②：削除したい場合は機能選択の右側に表示している [削除] ボタンをクリックすることで機能に紐づく情報の全てが削除されます（プログラムを生成していた場合も削除されます）。

①：機能選択をクリックすることで今まで作成した機能のリストが表示されます。
選択した機能の詳細情報が表示されます。



③：変更したい場合は表示された内容を新規作成時と同様にそれぞれの項目に従い情報を書き換え、[保存] ボタンをクリックすることで変更することができます。

5.2.3. 解析する機能設計書のイメージ

画面設計書									
画面項目説明	システム名称 CodeAGI	サブシステム名称 品目管理	業務名称 入出荷管理業務	画面名称 品目在庫管理一覧画面	作成日 修正日	作成者 修正者	SOPPEA		
No	論理名	物理名	object種類 (クラス名)	I/O	型	エラーチェック 必須 最大桁数 MAX MIN	フォーマット	備考	
1	品目在庫管理一覧画面		label					画面の見出し	
2	品目コード		label					項目見出しlabel	
3	品目コード	itemCodeSearch	TextBox	I	テキスト				
4	保管場所		label						
5	保管場所	storageLocationSearch	Listbox	I	テキスト				
6	検索	searchButton	button	-	-				
7	品目ID	itemID	table hidden	O	数値				
8	品目コード	itemCode	table アンカーリンク	O	テキスト				
9	品目名	itemName	table	O	テキスト				
10	品目種別	itemType2	table	O	テキスト				
11	主要取引先	mainTradingPartner	table	O	テキスト				
12	復社購買	companyPurchase	table アンカーリンク	O	テキスト			品目テーブルの仕入先 空文字	
13	月初在庫	beginningOfMonthInventory	table	O	数値		###,###,###		
14	前日迄入庫数	previousDayStockNumber	table	O	数値		###,###,###		
15	前日迄出庫数	previousDayDispatchNumber	table	O	数値		###,###,###		
16	当日入庫数	currentStockNumber	table	O	数値		###,###,###		
17	当日出庫数	currentDispatchNumber	table	O	数値		###,###,###		
18	現在在庫数	currentInventory	table	O	数値		###,###,###		
19	入出庫処理	InOutProcessing	table button						

画面設計書									
画面項目説明	システム名称 CodeAGI	サブシステム名称 品目管理	業務名称 入出荷管理業務	画面名称 品目在庫管理一覧画面	作成日 修正日	作成者 修正者	SOPPEA		
アクション									
処理内容									
1. 初期処理	1. 初期処理を行う 1.1. 現在のユーザが本画面を使用できるかどうかの権限チェックを行う。 1.1.1 共通権限チェック 1.2. 保管場所の取得 1.2.1 保管場所テーブルより、全ての保管場所取得する。 1.2.2 取得した保管場所テーブルを、ID、保管場所名をListBoxの要素としてセットする。								
2. 検索ボタン	2. 入出庫情報を検索し、時点在庫の算出を行い、一覧表示する。品目コードを条件にデータを検索する。 検索するデータは、品目テーブルを基本に、入出庫テーブルを検索し、理論在庫を算出する。 2.1. 始めに品目テーブルを検索する。 2.1.1 品目テーブルの検索条件は、画面から入力された、それぞれの項目が存在していれば、品目コードを取得する条件とする。 AND 品目名 LIKE '%画面パラメータ品目コード%' 2.2. 検索した品目テーブルを元に、Loopを行い、入出庫テーブルを取得する。 2.2.1 品目テーブルから取得した、品目IDと画面パラメータ保管場所IDを元に月初在庫テーブルを検索し数量を取得する取得した数量が【月初在庫数量】となる。 2.2.1 品目テーブルから取得した、品目IDと画面パラメータ保管場所IDを元に入出庫テーブルを検索する。 2.2.1.1 処理実施日が1日なら前日迄入庫数・前日迄出庫数をとし、1日以外は入出庫テーブルから数量を取得する。 2.2.1.1.1 品目テーブル、品目IDを元に、入出庫テーブルから月初から昨日までの入出庫数を 入出庫区分 = 1の条件でサマリ算出する。取得条件は以下に示す【前日迄入庫数】 品目ID = 取得した品目テーブル.ID and 入出庫テーブル.入出庫日時 >= 当月01日00時00分 and 入出庫テーブル.入出庫日時 <= 昨日23時59分59秒 and 入出庫区分 = 1 2.2.1.1.2 品目テーブル、品目IDを元に、入出庫テーブルから月初から昨日までの入出庫数を 入出庫区分 = 2の条件でサマリ算出する。取得条件は以下に示す【前日迄出庫数】 品目ID = 取得した品目テーブル.ID and 入出庫テーブル.入出庫日時 >= 当月01日00時00分 and 入出庫テーブル.入出庫日時 <= 昨日23時59分59秒 and 入出庫区分 = 2 2.2.1.3 品目テーブル、品目IDを元に、入出庫テーブルから入出庫区分 = 1の条件で当日の入出庫数をサマリ算出する。【当日入庫数】 品目ID = 取得した品目テーブル.ID and 入出庫テーブル.入出庫日時 >= 当日00時00分 and 入出庫テーブル.入出庫日時 <= 当日23時59分59秒 and 入出庫区分 = 1 2.2.1.4 品目テーブル、品目IDを元に、入出庫テーブルから入出庫区分 = 2の条件で当日の入出庫数をサマリ算出する。【当日出庫数】 品目ID = 取得した品目テーブル.ID and 入出庫テーブル.入出庫日時 >= 当日00時00分 and 入出庫テーブル.入出庫日時 <= 当日23時59分59秒 and 入出庫区分 = 2 2.2.1.5 上記で算出したそれぞれの在庫を元に (月初在庫+前日迄入庫数+当日入庫数) - (前日迄出庫数+当日出庫数) を行い現在在庫を算出する【現在在庫】 2.3. 取得したデータを画面に返却するリザルトセットにセットしデータを画面に返却し、HTML側でLoopしながら、そのリザルトセットを表示する。								
3. 品目コードアンカークリック	3. マスター管理システムの品目マスターを表示する。								

画面設計書									
画面項目説明	システム名称 CodeAGI	サブシステム名称 品目管理	業務名称 入出荷管理業務	画面名称 例外入出庫画面	作成日 修正日	作成者 修正者	SOPPEA		
編集処理の内容									
出庫編集									
No	論理項目名	物理項目名	編集元	編集元項目	編集方法				
1	入出庫ID	ID	-	-	自動採番				
2	入出庫区分	TRANSACTION_TYPE	-	-	"2"固定				
3	入出庫日付	TRANSACTION_DATE	例外入出庫画面	入出庫日時	-				
4	品目ID	ITEM_ID	例外入出庫画面	品目ID	隣り項目の品目ID				
5	入出庫数量	TRANSACTION_QUANTITY	例外入出庫画面	入出庫数量	-				
6	入出庫場所	TRANSACTION_LOCATION	例外入出庫画面	出庫場所	-				
7	入出庫理由	TRANSACTION_REASON	例外入出庫画面	入出庫理由	-				

※：上記以外にエラーチェック一覧等も解析することができます。

5.3. 画面イメージを指定する

「10.5 デザインの生成 α version」参照

5.4. 共通仕様書を指定する



[共通仕様書指定] ボタンをクリックすることで下図の画面が表示されます。

- ① : 生成するプログラムで使用する共通仕様書 (コード定義、定数定義、メッセージ一覧、環境定義) を追加し [保存] ボタンをクリックしてください。
 - ② : 保存完了後に表示されるメッセージで [確認] をクリックするか左のメニューから [指定データ確認] ボタンをクリックし内容を確認します。
- 確認する手順の詳細については「5.6 指定情報の確認」を参照してください。



5.4.1. 解析する共通仕様書のイメージ

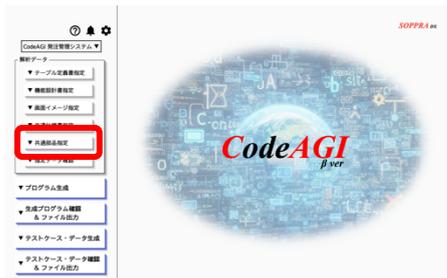
アプリケーション共通設定 (application.yml)								
キー				設定値	デフォルト値	必須	説明	関連部品
L-0	L-1	L-2	L-3					
server	server	context-path		/api	/	○	アプリケーションのコンテキストパス	なし
	port			8080	8080		アプリケーションのポート番号	なし
spring	datasource	hikari	connection-timeout	120000	30000		DBの接続タイムアウト	DB接続、トランザクション制御
spring	datasource	hikari	maximum-pool-size	20	10		コネクションプールの最大数	DB接続、トランザクション制御
spring	datasource	hikari	minimum-idle	15	10		アイドル数	DB接続、トランザクション制御
spring	datasource	driver-class-name		oracle.jdbc.driver.OracleDriver			OracleDB接続情報	DB接続、トランザクション制御
spring	datasource	url		○○○			OracleDB接続情報	DB接続、トランザクション制御
spring	datasource	username		○○○			OracleDB接続情報	DB接続、トランザクション制御
spring	datasource	password		○○○			OracleDB接続情報	DB接続、トランザクション制御

プロジェクト名	業務名	画面ID	画面名	作成日	作成者	更新日	更新者

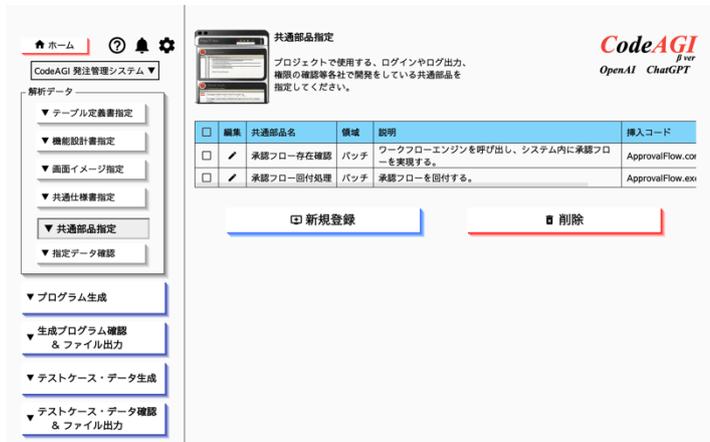
No.	コード区分	コード名称	項目値	項目名	項目型	項目桁数	説明	備考
2	C0001	保管場所対応	1	小物	テキスト	1		
			2	中物				
			3	大物				

会議予約システム		予約管理	会議室予約登録画面			
No.	分類	クラス名	定数論理名	定数物理名	型	値
1	入出庫区分		入庫区分	RECEIPT_CLASSIFICATION1	Integer	1
2			出庫区分	ISSUE_CLASSIFICATION	Integer	2

5.5. 共通部品指定



[共通部品指定] ボタンをクリックすることで以下の画面が表示されます。



①: [新規登録] ボタンをクリックすることで共通部品の登録画面が表示されます。画面に沿ってそれぞれの項目を設定し、[保存] ボタンをクリックしてください。

当画面で登録した共通部品名を設計書の記述によりソースコードとして出力されます。

(登録された共通部品は生成 AI のサーバーに送信されることなくコードとして生成されます)

```

AND 入出荷テーブル.在庫変更タイプ = 2 出荷
AND 入出荷テーブル.入出荷日時 <= 当月+"01"+"00:00:00"
AND 入出荷テーブル.入出荷日時 >= 当日
1.4 現時点の理論在庫を算出する。
    月初在庫 + 入荷数量 - 出荷数量 = 現時点の理論在庫
1.5 算出した理論在庫と安全在庫を比べる
    (商品テーブル.安全在庫数量 > 1.4で算出した理論在庫) だった場合
1.5.1 商品テーブルの在庫レベルを参照し、承認フローだった場合
    商品テーブル.在庫レベル = 1 (承認フロー)
1.5.1.1 同一の承認フローがフロー中かの確認を行い発行されていない場合は、承認フローAPIを実行する
    1.5.1.1.1 (共通部品を代入) 共通承認フロー存在確認 (商品レコードセット.商品ID、商品レコードセット.商品名、商品レコードセット.取引先ID、商品レコードセット.平均仕
    1.5.1.1.2 (共通部品を代入) 共通承認フロー回付 (商品レコードセット.商品ID、商品レコードセット.商品名、商品レコードセット.取引先ID、商品レコードセット.平均仕
1.5.2 商品レコードセットの在庫レベルを参照し、自動発注だった場合
    商品レコードセット.在庫レベル = 2 (自動発注)
1.5.2.1 発注発注データを作成する。
    1.5.2.1.1 発注テーブルを参照し、発注中のデータ存在確認を行う以下の条件でデータが存在した場合は、発注テーブルを作成しない
        発注テーブル.商品ID = 商品レコードセット.商品ID
        発注テーブル.発注ステータス IN (1,2,3)
  
```

5.5.1. 共通部品指定の編集と削除

共通部品指定

プロジェクトで使用する、ログインやログ出力、権限の確認等各社で開発をしている共通部品を指定してください。

<input type="checkbox"/>	編集	共通部品名	領域	説明	挿入コード
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	承認フロー存在確認	バッチ	ワークフローエンジンを呼び出し、システム内に承認フローを実現する。	ApprovalFlow.cor
<input type="checkbox"/>	<input checked="" type="checkbox"/>	承認フロー回付処理	バッチ	承認フローを回付する。	ApprovalFlow.exe

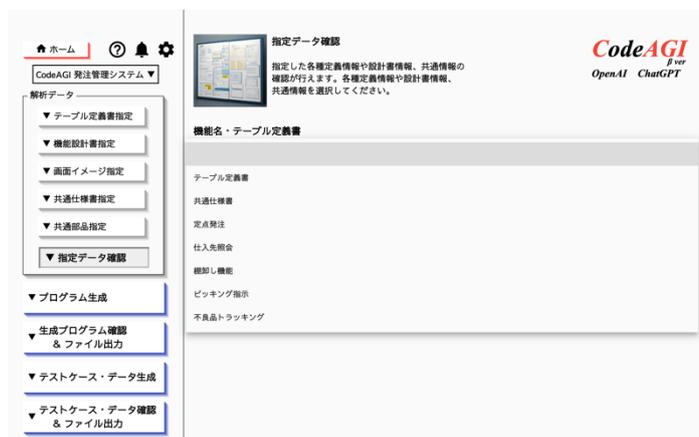
①：編集する場合は編集列のペンのアイコンをクリックすることで上記で説明した新規登録画面と同じ画面が表示されます。

②：削除する場合はチェックボックスをクリックしチェックを付けて[削除]ボタンをクリックすれば削除できます。

5.6. 指定情報の確認



Excel の設計書を指定した結果を以下に表示します。



①：機能名・テーブル定義書のリストボックスをクリックすると3種類の定義書が表示されます

- ・テーブル定義書
- ・共通仕様書
- ・機能設計書 (機能名になります)

これらを選択することでその中に含まれている Excel シートが設計書・シート名一覧に表示されます。



②：そのシート名をクリックすることで CodeAGI に指定したデータをスプレッド形式で表示します。Excel で作成した形と異なりますが設計内容はこの機能で確認することができます。

6 プログラム生成

この章ではプログラム生成を行うための操作等を説明します。

6.1. プログラム生成



画面左下の「プログラム生成」ボタンをクリックすることでプログラム生成画面が表示されます。

①: 生成する機能を選択の上、画面中央の「プログラム生成」ボタンをクリックし、確認画面で「はい」ボタンをクリックしてください。

②: 右図の画面が表示されます。この画面はプログラム生成の状況をリアルタイムで表す画面となります。

プロジェクト情報画面で選択した言語、フレームワーク、パッケージにより必要なプログラムを表示します。

生成はプログラム単位で行われ完了と同時にソースが表示されます。

途中で処理を中断したい場合は「中止」ボタンをクリックすることで生成を中断することができます。

③: ソースを確認したい場合は中央に表示されている「生成済プログラム」から任意のプログラムをクリック又は右側に表示されている「生成済プログラムコード」をクリックすることでプログラムコードが表示されます。

プロジェクト名: CodeAGI 発注管理システム
機能名: 定発注

61% <<< 中止 再生成 戻る

●生成開始日時: 2024-01-31 14:49:06 (経過時間: 00:03:35)

ステータス	プログラム名	生成開始日時 YYYY-MM-DD HH:mm:ss	生成時間 hh:mm:ss
✓ 生成完了	定発注処理 Constant.java	2024-01-31 14:49:07	00:00:03
✓ 生成完了	Enum.java	2024-01-31 14:49:11	00:00:24
✓ 生成完了	messages.properties	2024-01-31 14:49:36	00:00:16
✓ 生成完了	Entity.java	2024-01-31 14:49:51	00:00:41
✓ 生成完了	Repository.java	2024-01-31 14:50:23	00:00:48
✓ 生成完了	ItemProcessor.java	2024-01-31 14:51:22	00:01:12
○ 生成中	Configuration.java	2024-01-31 14:52:35	
生成待ち	JobLauncher.java		
生成待ち	SystemApplication.java		
生成待ち	プログラムの最終構築		

●生成済プログラム (合計コードライン数: 250)

- 定発注処理 ProductRepository.java (コードライン数: 10)
- 定発注処理 MonthlyStockRepository.java (コードライン数: 10)
- 定発注処理 ReceiptShipmentRepository.java (コードライン数: 12)
- 定発注処理 OrderRepository.java (コードライン数: 10)
- 定発注処理 FixedPointOrderProcessor.java (コードライン数: 64)

●生成済プログラムコード (コードライン数: 64)

```
package com.example;

import com.example.entity.ProductEntity;
import com.example.entity.OrderEntity;
import com.example.repository.ProductRepository;
import com.example.repository.MonthlyStockRepository;
import com.example.repository.ReceiptShipmentRepository;
import org.slf4j.Logger;
import org.springframework.batch.item.ItemProcessor;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.service.ApprovalFlow;
import java.math.BigDecimal;

public class FixedPointOrderProcessor implements ItemProcessor<ProductEntity, OrderEntity> {

    private static final Logger log =
        LoggerFactory.getLogger(FixedPointOrderProcessor.class);
```



6.2. プログラム生成画面について

プロジェクト名: CodeAGI 発注管理システム
機能名: 定点発注

61% << 中止 再生成 戻る

●生成開始日時: 2024-01-31 14:49:06
(経過時間: 00:03:35)

ステータス	プログラム名	生成開始日時 yyyy-MM-dd hh:mm:ss	生成時間 hh:mm:ss
✓ 生成完了	Constant.java	2024-01-31 14:49:07	00:00:03
✓ 生成完了	Enum.java	2024-01-31 14:49:11	00:00:24
✓ 生成完了	messages.properties	2024-01-31 14:49:36	00:00:15
✓ 生成完了	Entity.java	2024-01-31 14:49:51	00:00:41
✓ 生成完了	Repository.java	2024-01-31 14:50:33	00:00:48
✓ 生成完了	ItemProcessor.java	2024-01-31 14:51:22	00:01:12
● 生成中	Configuration.java	2024-01-31 14:52:35	
生成待ち	JobLauncher.java		
生成待ち	SystemApplication.java		
生成待ち	プログラムの最終調整		

CodeAGI
β Ver
OpenAI ChatGPT

累計生成回数 : 1
累計コードライン数: 250

リセット

●生成済プログラム (合計コードライン数: 250)

```
<Repository>
```

- 定点発注処理 ProductRepository.java コードライン数: 10
- 定点発注処理 MonthlyStockRepository.java コードライン数: 10
- 定点発注処理 ReceiptShipmentRepository.java コードライン数: 12
- 定点発注処理 OrderRepository.java コードライン数: 10

```
<ItemProcessor>
```

- 定点発注処理 FixedPointOrderProcessor.java コードライン数: 64

●生成済プログラムコード (コードライン数: 64)

```
package com.example;

import com.example.entity.ProductEntity;
import com.example.entity.OrderEntity;
import com.example.repository.ProductRepository;
import com.example.repository.MonthlyStockRepository;
import com.example.repository.ReceiptShipmentRepository;
import com.example.repository.OrderRepository;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.batch.item.ItemProcessor;
import org.springframework.beans.factory.annotation.Autowired;

import com.example.service.ApprovalFlow;
import java.math.BigDecimal;

public class FixedPointOrderProcessor implements
ItemProcessor<ProductEntity, OrderEntity> {

    private static final Logger log =
LoggerFactory.getLogger(FixedPointOrderProcessor.class);
```

- ①: 中止…プログラム生成を中止します。中止した時点のソースは参照可能となります。
- ②: 再生成…プログラム生成をもう一度始めから実行することができます。
- ③: リセット…累計生成回数・累計コードライン数 が全て0にクリアされます。

6.3. 連続プログラム生成について

プログラム生成

プログラム生成する機能を選択

ID	機能名	機能説明
1	定点発注	安全基準を下回った場合に自動で発注するシステムの仕組み
2	仕入先管理	仕入先管理
3	納期・数量	品目の納期・数量を行う機能
4	ピッキング指示	工場管理へのピッキング指示を行う
5	不良品トラッキング	不良品のトラッキングを行うリシュー登録及び検閲機能

プログラム生成

①: 左図の画面にてプログラム生成を行う機能をチェックボックスで選択してください。

②: [プログラム生成] ボタンをクリックしてください。

③: 以下の画面が表示されます。

④: 選択した機能が連続して生成されます。

※: 選択した機能が表示され進捗が見えます。

プロジェクト名: CodeAGI 発注管理システム
機能名: 仕入先紹介

35% << 中止 戻る

●生成開始日時: 2024-01-31 14:58:47
(経過時間: 00:01:58)

ステータス	プログラム名	生成開始日時 yyyy-MM-dd hh:mm:ss	生成時間 hh:mm:ss
✓ 生成完了	SystemApplication.java	2024-01-31 14:58:48	00:00:06
✓ 生成完了	Repository.java	2024-01-31 14:59:03	00:00:18
✓ 生成完了	Service.java	2024-01-31 14:59:04	00:00:14
✓ 生成完了	Form.java	2024-01-31 14:59:19	00:00:09
✓ 生成完了	Controller.java	2024-01-31 14:59:29	00:00:13
● 生成中	Item	2024-01-31 14:59:43	
生成待ち	仕入先管理画面 Constant.java		
生成待ち	Enum.java		
生成待ち	messages.properties		
生成待ち	Entity.java		
生成待ち	SystemApplication.java		

CodeAGI
β Ver
OpenAI ChatGPT

累計生成回数 : 3
累計コードライン数: 858

リセット

●生成済プログラム (合計コードライン数: 165)

- 品目在庫管理一覧画面 Repository.java コードライン数: 12
- 品目在庫管理一覧画面 Service.java コードライン数: 17
- 品目在庫管理一覧画面 Form.java コードライン数: 10
- 品目在庫管理一覧画面 Controller.java コードライン数: 29

●生成済プログラムコード (コードライン数: 29)

```
package com.example.controller;

import com.example.form.SearchForm;
import com.example.service.MonthlyStockService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;

import javax.validation.Valid;
@Controller
```

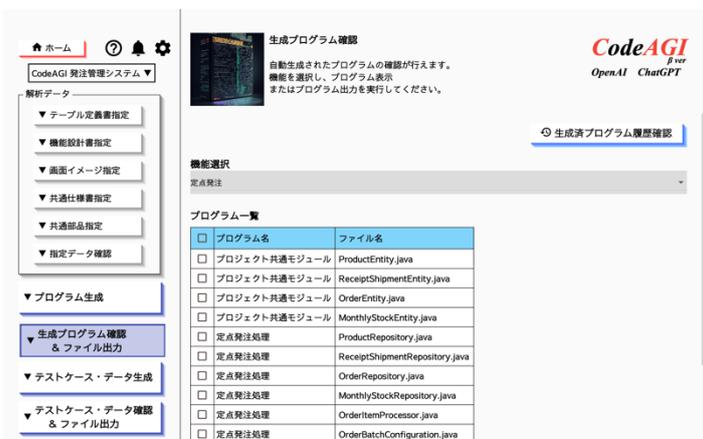
7 生成したプログラムの確認およびファイル出力

この章では生成したプログラムを確認できると同時に生成時に必要とした生成 AI リソースやそれにかかった時間、生成したコードライン数を確認する方法を説明します。

7.1. 生成完了プログラムの確認およびファイル出力



①: [生成プログラム確認&ファイル出力] ボタンをクリックすることで左図の画面が表示されます。



②: 機能選択から出力・表示したい機能を選択してください。選択することでプログラムの一覧が表示されます。



③: 出力したいプログラムを選択チェックボックスにて選択し (スプレッド一番上の青のヘッダ部分にあるチェックを選択すれば全てチェックされます)。[プログラム表示] 又は [ファイル出力] ボタンをクリックしてください。

7.2. 生成履歴を確認する



[生成履歴確認] ボタンをクリックすることで
下図の画面が表示されます。

以下の画面は、生成したプログラムを表示する画面となります。

β ver
OpenAI ChatGPT

生成済プログラム履歴

選択	削除	生成日時	生成時間	プロジェクトID	プロジェクト名	機能ID	機能名	生成ファイル数	生成コードライン数
<input type="checkbox"/>	<input type="checkbox"/>	2024-04-05 18:23:55	00:01:42	1	CodeAGI 発注管理システム	2	仕入先照会	3	135
<input type="checkbox"/>	<input type="checkbox"/>	2024-04-05 18:21:53	00:02:02	1	CodeAGI 発注管理システム	1	定点発注	12	417
<input checked="" type="checkbox"/>	<input type="checkbox"/>	2024-04-05 18:17:47	00:03:28	1	CodeAGI 発注管理システム	1	定点発注	12	450

プログラム一覧

<input type="checkbox"/>	プログラム名	ファイル名
<input checked="" type="checkbox"/>	プロジェクト共通モジュール	ProductEntity.java
<input type="checkbox"/>	プロジェクト共通モジュール	ReceiptShipmentEntity.java
<input type="checkbox"/>	プロジェクト共通モジュール	OrderEntity.java
<input type="checkbox"/>	プロジェクト共通モジュール	MonthlyStockEntity.java
<input type="checkbox"/>	定点発注処理	ProductRepository.java
<input type="checkbox"/>	定点発注処理	ReceiptShipmentRepository.java
<input type="checkbox"/>	定点発注処理	OrderRepository.java
<input type="checkbox"/>	定点発注処理	MonthlyStockRepository.java

プログラムコード

```

package com.example.entity;

import lombok.Data;

import javax.persistence.*;
import java.math.BigDecimal;

/**
 * 商品テーブル (PRODUCT)
 */
@Entity
@Table(name = "PRODUCT")
@Data
public class ProductEntity {

```

戻る
<> プログラム表示
↓ ファイル出力

① : 生成した履歴の一覧が表示されます。

また、履歴全て残りますので不必要になった場合には削除列のごみ箱アイコンをクリックし定期的に整理することをお勧めします。

② : ①のспредを選択することでその時に生成された全てのプログラムが一覧で表示されます。チェックボックスを選択し [プログラム表示] ボタンで選択した全てのプログラムを表示することができます。また [ファイル出力] ボタンをクリックすることで選択した全てのプログラムを指定フォルダに出力することができます。

③ : ②のプログラム一覧の表示列を選択することでプログラムコードが表示されます。

8 テストケース・データ生成

この章ではテストケース・データ生成を行うための操作等を説明します。

8.1. テストケース・データ生成



画面左下の「[テストケース・データ生成]」ボタンをクリックすることでテストケース・データ生成画面が表示されます。

①：生成する機能を選択し「[テストケース・データ生成]」ボタンをクリックしてください。

②：右図の画面が表示されます。この画面はテストケース及びテストデータ生成の状況をリアルタイムで表す画面となります。生成は機能およびテーブル単位で行われ、完了と同時にテストケース及びテストデータが表示されます。途中で処理を中断したい場合は、「[中止]」ボタンをクリックすることで生成を中断することができます。

③：テストケース及びテストデータを確認したい場合は、中央に表示されている「生成済テストケース・データ」から任意のテストケース及びテストデータをクリック又は右側に表示されている「[生成済テストケース・データ]」をクリックすることでテストケース及びテストデータが表示されます。

8.2. テストケース・データ生成画面について

プロジェクト名: CodeAGI 発注管理システム
機能名: 定点発注

100% 中止 再生成 戻る

●生成開始日時: 2024-04-05 18:28:01
(生成時間: 00:02:32)

ステータス	テストケース・データ名	生成開始日時 yyyy-MM-dd hh:mm:ss	生成時間 hh:mm:ss
✓ 生成完了	定点発注処理 テストケース	2024-04-05 18:28:04	00:01:18
✓ 生成完了	テストデータ	2024-04-05 18:29:23	00:01:10

CodeAGI β ver
OpenAI ChatGPT

累計生成回数: 1
累計テストケース数: 8
累計テストデータ数: 56 リセット

●生成済テストケース・データ
(合計ファイル数: 5)

<定点発注処理>

- テストケース
定点発注処理 テストケース
テストケース(定点発注処理).xlsx
- テストデータ
PRODUCT
商品テーブル.csv
- テストデータ
RECEIPT_SHIPMENT
入出荷テーブル.csv
- テストデータ
ORDER
発注テーブル.csv
- テストデータ
MONTHLY_STOCK
月初在庫テーブル.csv

●生成済テストケース・データ
(テストケース数: 8)

No	テストケース	テスト観点	確認内容
1	安全在庫以下の商品の確認	データ取得	安全在庫数量が現時点の
2	承認フローの確認	データ取得	承認フローが必要な商品
3	自動発注の確認	データ取得	自動発注が必要な商品で
4	安全在庫以下で自動発注データの登録	ビジネスチェック	安全在庫数量を下回った
5	承認フロー必要商品の承認フロー開始	ビジネスチェック	安全在庫数量を下回った
6	発注データの編集パターン	データ編集	発注データが編集定義に
7	システム障害時の処理確認	システム障害	データベース接続障害時
8	パフォーマンス要件の確認	非機能要件	大量データ処理時のパ

- ①中止: テストケース・データ生成を中止します。中止した時点のデータは参照可能となります。
- ②再生成: テストケース・データ生成をもう一度始めから実行することができます。
- ③リセット: 累計生成回数、累計テストケース、データ数 が全て0にクリアされます。

8.3. 連続テストケース・テストデータ生成について

- ①: 左図の画面にて生成を行う機能をチェックボックスで選択してください。
- ②: [テストケース・データを作成] ボタンをクリックしてください。
- ③: 以下の画面が表示されます。

④: 選択した機能が連続して生成されます。

※: 選択した機能が表示され、進捗が見えます。

9 生成したテストケース・データの確認およびファイル出力

この章では生成したテストケース及びテストデータを確認できると同時に生成時に必要とした生成 AI リソースやそれにかかった時間、生成したケース数・データ数を確認する方法を説明します。

9.1. 生成完了テストケース・データの確認およびファイル出力



①: [テストケース・データ確認&ファイル出力]ボタンをクリックすることで左図の画面が表示されます。



②: 機能選択から出力・表示したい機能を選択してください。選択することでテストケース・データの一覧が表示されます。



③: 出力したいテストケース・データをチェックボックスにて選択し(スプレッド一番上の青のヘッダ部分にあるチェックを選択すれば全てチェックされます)。[テストケース・データ表示] 又は [ファイル出力]ボタンをクリックしてください。

9.2. 生成履歴を確認する



[生成済テストケース・データ履歴確認] ボタンをクリックすることで下図の画面が表示されます。

以下の画面は生成したテストケース・データを表示する画面となります。



β ver
OpenAI ChatGPT

生成済テストケース・データ履歴

選択	削除	生成日時	生成時間	プロジェクトID	プロジェクト名	機能ID	機能名	生成ファイル数	生成ケース数	生成データ数
<input checked="" type="checkbox"/>		2024-04-05 18:42:30	00:03:47	1	CodeAGI 発注管理システム	1	定点発注	5	22	56
<input type="checkbox"/>		2024-04-05 18:28:02	00:02:31	1	CodeAGI 発注管理システム	1	定点発注	5	8	56

テストケース・データ一覧

<input type="checkbox"/>	プログラム名	ファイル種別	データ名	ファイル名
<input checked="" type="checkbox"/>	定点発注処理	テストケース	定点発注処理 テストケース	テストケース(定点)
<input type="checkbox"/>	定点発注処理	テストデータ	商品テーブル	PRODUCT
<input type="checkbox"/>	定点発注処理	テストデータ	入出荷テーブル	RECEIPT_SHIPME
<input type="checkbox"/>	定点発注処理	テストデータ	発注テーブル	ORDER
<input type="checkbox"/>	定点発注処理	テストデータ	月初在庫テーブル	MONTHLY_STOCK

テストケース・データ

No	テストケース	テスト観点	確認内容
1	安全在庫確認	データ取得	商品テーブル
2	月初在庫数量取得	データ取得	月初在庫データ
3	入荷数量算出	データ取得	入出荷テーブル
4	出荷数量算出	データ取得	入出荷テーブル
5	理論在庫算出	データ取得	理論在庫データ
6	安全在庫比較	データ取得	理論在庫データ
7	承認フロー確認	データ取得	承認フローデータ
8	自動発注データ作成	データ取得	発注テーブル
9	安全在庫以下での発注データ登録	業務チェック	安全在庫データ

戻る
[1] テストケース・データ表示
↓ ファイル出力

①：生成した履歴の一覧が表示されます。

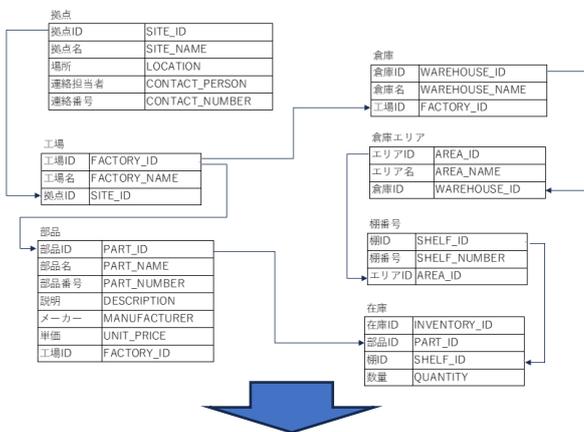
また、履歴が全て残りますので不必要になった場合には削除列のごみ箱アイコンをクリックし定期的に整理することをお勧めします。

②：①の Spredd を選択することで、その時に生成された全てのテストケース・データが一覧で表示されます。チェックボックスを選択し [テストケース・データ表示] ボタンで選択した全てのテストケース・データを表示することができます。また [ファイル出力] ボタンをクリックすることで選択した全てのテストケース・データを指定フォルダに出力することができます。

③：②のテストケース・データ一覧の表示列を選択することで、テストケース・データが表示されます。

10 その他

10.1. 複雑な SQL 複数のテーブルの結合について



左図のような ER-D に対して CodeAGI で結合用の query を生成させるための方法を説明します。

```

/**
 * 在庫データにアクセスするためのMapperインターフェース
 */
Mapper
public interface InventoryMapper {
    /**
     * 在庫情報を取得するためのクエリを定義する
     * @return 在庫情報のリスト
     */
    @Select("SELECT s.siteName, w.warehouseName, p.partId, p.partName, SUM(i.quantity) as quantity "
        + "FROM SITES s "
        + "JOIN FACTORIES f ON s.siteId = f.siteId "
        + "JOIN WAREHOUSES w ON f.factoryId = w.factoryId "
        + "JOIN WAREHOUSEAREAS wa ON w.warehouseId = wa.warehouseId "
        + "JOIN SHELFNUMBERS sn ON wa.areaId = sn.areaId "
        + "JOIN INVENTORY i ON sn.shelfId = i.shelfId "
        + "JOIN PARTS p ON i.partId = p.partId "
        + "LEFT JOIN ORDERDETAILS od ON p.partId = od.productId "
        + "LEFT JOIN ORDERS o ON od.orderId = o.orderId AND o.orderDate >= TRUNC(SYSDATE, 'MM') "
        + "WHERE o.orderId IS NULL "
        + "GROUP BY s.siteName, w.warehouseName, p.partId, p.partName")
    List<InventoryModel> FindInventory();
}
    
```

左のソースコードは CodeAGI で生成したものになりますが、JOIN 句により全てのテーブルの結合が正確に記述されていることが分かります。このようなソースコードを出力させるためにはテーブル定義に外部キーの設定を記入する必要があります。

以下のようにテーブル定義に外部キーを明記することで CodeAGI は外部キーに関する情報を基にソースコードを生成します。

テーブル名 (論理)		在庫	テーブル名 (物理)		INVENTORY		
No	論理項目名	物理項目名	データ型	桁数	PK	外部キー参照先	
1	在庫ID	INVENTORY_ID	NUMBER	13	1	テーブル	項目名
2	部品ID	PART_ID	NUMBER	10	-	-	部品ID
3	棚ID	SHELF_ID	NUMBER	7	-	棚番号	棚ID
4	数量	QUANTITY	NUMBER	7	-	-	-

10.2. 生成コードのファイル名などに機能 ID を使用する方法



- ①: プロジェクト情報画面より [生成プログラムに画面 ID・バッチ ID を使う] を ON します。
- ②: [共通仕様書指定] ボタンにて (5.3. 共通仕様書 参照) 機能一覧を読み込んでください。この機能一覧には機能名と機能 ID が対になっている必要があります。

「読む機能一覧の例」

TYPE	機能ID	作成者	区分	機能概要の範囲	画面名(WEB表示) バッチ名(NETOS登録)	メニュー	位置	他	内容
B	SIAB02	羽狭	改修	(01. 出荷情報管理)	生産計画情報受信				BOOSダウンロードの生産計画情報に基き、出荷情報を作成する。
B	SIAB03	羽狭	改修	(01. 出荷情報管理)	発注データ受信				BOOSダウンロードの発注情報に基き中間DBを更新する
B	SIAB04	羽狭	改修	(01. 出荷情報管理)	指図・指配データ受信				指図情報、指配情報の取り込み
B	SIAB05	羽狭	改修	(01. 出荷情報管理)	発注・指図・指配データ作成				上記の情報を基き出荷情報を作成する。
B	SIAB07	羽狭	新規	(01. 出荷情報管理)	搬入予約情報受信				BOOSダウンロードの搬入予約情報に基き中間DBを更新する
B	SIAB08	羽狭	新規	(01. 出荷情報管理)	搬入予約情報反映				生産計画情報に基き、搬入予定日を決定する。
B	SIAB09	羽狭	新規	(01. 出荷情報管理)	アセンブル計画化				アセンブルの半準化を行い、アセンブルローレット日を決定する。
B	SIAB10	羽狭	新規	(01. 出荷情報管理)	鉄筋指図情報受信				BOOSダウンロードの鉄筋指図情報に基き中間DBを更新する。
B	SIAB11	羽狭	新規	(01. 出荷情報管理)	鉄筋指図情報反映				中間DBの鉄筋指図情報に基き、出荷情報を作成する。
B	CTRL06	羽狭	Vup	(01. 出荷情報管理)	BOOSファイル退避				更新に使用したBOOSデータを退避する(ANISが動作している前提の機能です)
CSV帳票		XXXXXXXX	仕工	(01. 出荷情報管理)	帳票: 出荷指図書				
CSV帳票		XXXXXXXX	仕工	(01. 出荷情報管理)	帳票: 未登録部材エラーリスト				
CSV帳票		XXXXXXXX	仕工	(01. 出荷情報管理)	帳票: 運移エラーリスト				
B	SIAB01	羽狭	改修	(01. 出荷情報管理)	入金情報DB作成				ACQSMOSデータを支払データ受信後、ANIS2へ入金へ連携する

機能種別	サブシステム	カテゴリ	Category	PG	機能ID	備考 (Function Level)	年	月	締	日	随	オン	共	PG TYPE	処理の順序			
販売売掛管理 Sales in Receivables Management	日次売掛残高更新	update	Closing Balance	マスタ情報洗替	RC8010									更新	Z			
				契約単価洗替	RC8020										更新	Z		
	請求書印刷データ作成	create	Invoice Data	日次売掛残高更新	RC8030										更新	D		
				締次売掛残高更新	R10010										更新	C		
				請求書印刷データ作成	R10020											更新	Z	
				資料印刷データ作成	R10030											更新	Z	
	引落依頼データ作成	create	Withdrawal Request Data	値引・値増用データ作成	R10040										更新	Z		
				油外												更新	P	
	振替依頼	Transfer Request	銀行系データ作成	RWR010											出力	D		
			信販系データ作成	RWR020												出力	D	
	請求先送付データ作成	create	Billing Data	リスト出力												出力	P	
				ダイヤビジネス	RB0010												出力	D
				松下LEC	RB0020												出力	Z
				センコー商事	RB0030												出力	Z
				和歌山学生協	RB0040												出力	Z
				和歌山農協連	RB0050												出力	Z
				エスジーエス	RB0060												出力	Z
	間口運輸														出力	P		
	請求データ一覧送信														出力	P		
	発行指示画面				R11010										画面	D		
請求書発行自動				R11020										出力	F			
緊急分出力機能														更新	F			

10.3. 解析する設計書(仕様書)のシート名について

Excel シートは以下のルールに沿ってシート名を作成する必要があります。

CodeAGI では下記の文字列が Excel シートのシート名に記述されていることで、より正確な設計書や定義書の解析が可能となります。

データ解析	設計書	設計書の説明	文字列
テーブル定義書指定	テーブル定義書	DDL文の元になるテーブル単位の定義	テーブル定義書 (※:テーブル名) DB定義書 (※:テーブル名)
機能設計書指定	画面項目定義書	画面の項目 (object) を定義しその項目のI/Oやデータ起源、バリデーションチェックを定義する設計書	項目定義 画面項目説明 画面項目 画面入出力項目
	データ編集定義書	データベースのテーブルに対してプログラム中から書き込む要領を定義した設計書	編集定義 編集仕様 編集項目 データ入出力 DB更新
	アクション定義書	アクション単位 (ボタンやアンカー等) のプログラムの処理手順を記述した設計書	アクション 処理記述 処理説明 機能説明 処理機能記述 入出力処理設計 処理仕様
	I/F定義	フラットファイルや、ソケット通信等に対して、項目単位の編集を定義した設計書	IF仕様 IF定義
	チェック定義	バリデーションチェックや、ビジネスチェック等を定義した設計書	チェック一覧 チェックエラー関連処理 チェック仕様 チェック定義
	APIリクエスト定義書	画面等から送信される、パラメータを定義した設計書	リクエスト項目 リクエスト定義
	APIレスポンス定義書	APIのレスポンスを定義した設計書	リクエスト項目 リクエスト定義
	特記事項	フォーマットの制約では記述しにくい情報を記載する設計書	特記事項 補足説明 補足事項
共通仕様書指定	コード定義	enum等の列挙型構造体を定義した設計書	コード定義 コード一覧
	定数定義	constantのスタティックな定数を定義した設計書	定数定義 定数一覧
	環境定義	DBへの接続文字列や、カレントURL、ポート等環境に関する内容を定義した設計書	環境定義 application.yml
	メッセージ一覧	情報メッセージや、警告メッセージ、エラーメッセージ等のコードと内容を一覧にした設計書	メッセージ一覧 INFORMATION ERROR
	機能一覧	プログラム名とプログラムIDを一覧にした定義書	プログラム一覧 機能一覧

10.4. 機能名を取得できない場合

画面設計書									
画面項目説明	システム名称	サブシステム名称	業務名称	画面名称	作成日	作成者	修正日	修正者	SOFPRA
	CodeAGI	品目管理	入出荷管理業務	例外入出荷画面					
アクション		処理内容							
1. 初期処理	1. 受け取った品目IDを元に品目テーブルを参照し、品目名称を取得する、取得した品目名称を、画面に初期状態にする。 1.1 受け取った品目IDを元に、品目テーブルを参照する。 1.2 保管場所テーブルを全件取得し、例外入出荷画面の出庫場所と、入庫場所のListBox要素としてセットする。								
2. 登録	2. 入力された値を、入出荷テーブルにデータを登録する。 2.1 入力された値のエラーチェックを行う。 2.1.1 バリデーションチェックについては項目定義書を参照。 2.1.1.1 バリデーションエラーの場合は、処理を中断し、エラーメッセージを表示する。								

上記のような設計書が存在した場合、赤枠で囲った設計書ヘッダ (※1) からプログラム名を探します。この場合 AI による検索が行われるため意図しないものがプログラム名と判断する可能性があります。そのため CodeAGI は、画面機能の場合は〇〇画面、GUI を持っていない処理 (バッチ等) は〇〇処理と記述すれば、優先的にそれをプログラム名と判断することができます。

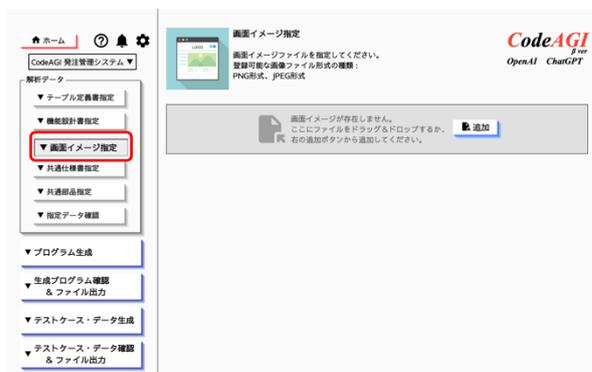
※1：設計書 (仕様書) のプロジェクト名や、作成者、機能名等を記載している共通の箇所を設計書ヘッダと定義

10.5. デザインの生成 α version

モックアップ等で作成される画像を CodeAGI が読み込み HTML ソースにデザイン情報として付加する試みを行っています。本機能は現在 α 版として実験的にリリースされております。

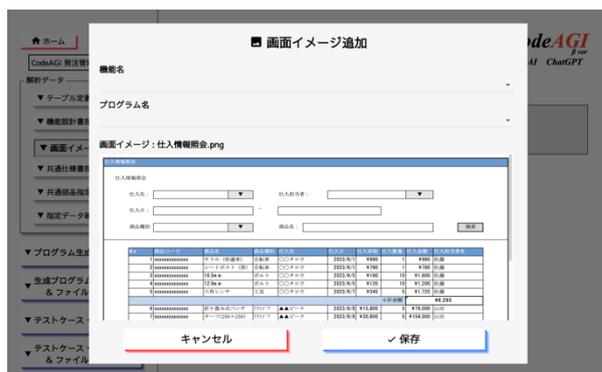
その為 Web アプリケーションの Java プロジェクトに対応しております。

また生成 AI は OpenAI の gpt-4o にて解析を行っており、その他の生成 AI では行えません。



①: メニューの [画面イメージ指定] ボタンをクリックすることで左図のような画面が表示されます。

②: png、jpg、jpeg 等の画面イメージの画像ファイルを設定します。設定するには [追加] ボタンをクリックするかエクスプローラー等のファイル参照からドラッグしてください。



③: ファイルを追加すると左のような Dialog がポップアップします。選択した画面イメージに該当する機能名とプログラム名を選択し、[保存] ボタンをクリックしてください。設計書との関連を作成します。

④: 設定できれば、6章「プログラム生成」の手順に沿ってプログラムを生成してください。デザインを含めた HTML (CSS) が作成されます。右図のイメージは、デザインを設定した、HTML を Web ブラウザーで表示した画面になります。

仕入情報照会

仕入先: 仕入担当者:

仕入開始日: ~ 仕入終了日:

商品種別: 商品名:

No	商品コード	商品名	商品種別	仕入先コード	仕入先	仕入日	仕入単価	仕入数量	仕入金額	仕入担当者名	小計金額
合計金額:											

11 付録

11.1. アンインストールについて



①: [設定] の [アカウント設定] を選択し [アカウント情報の削除] をクリックすることで、左図の画面が表示されます。

[現在の Login 中のアカウント情報を削除します。] をチェックし [削除する] ボタンをクリックしてください。アカウントが削除されます。

(注) OS の機能からアンインストールだけを行った場合、アカウントは削除されません。

②: アカウントの削除が完了した後に Windows、macOS それぞれの OS の機能からアンインストールを行ってください。

11.2. GPT の登録について

- ①: GPT のアカウントを作成してください。
 - ②: Add payment method から支払い情報の入力を行ってください。
 - ③: ホーム画面より View API keys をクリックしてください。
 - ④: +Create new secret key をクリックし sk で始まる Key を生成してください。
- 詳細については OpenAI GPT のマニュアルを参照ください。

<https://platform.openai.com/docs/overview>

11.3. Google Cloud Platform Vertex AI の登録について

- ①: [ハンバーガーメニュー] → [IAM と管理] → [サービスアカウント] を選択してください。
- ②: [+サービスアカウントを作成] をクリックしてください。
- ③: サービスアカウント名等の項目を入力し [完了] ボタンをクリックしてください。
- ④: 作成したサービスアカウントを選択してください。
- ⑤: 表示されたサービスアカウントの詳細画面でキータブをクリックしてください。
- ⑥: 鍵を追加→新しい鍵を作成をクリックしてください。
- ⑦: 表示された Dialog の JSON を選択して [作成] をクリックすることでキーが Download されます。

詳細については、GCP のマニュアルをご覧ください。

<https://cloud.google.com/?hl=ja>

11.4. macOS の証明書がないバージョンのインストール

- ①: Download したファイルを右クリックから [開く] を選択しセットアップを行ってください。
 - ②: 右図のような注意を促すメッセージが表示されますが [開く] ボタンをクリックしてインストールを行ってください。
 - ③: インストールが完了したらアプリケーションのアイコンを選択する際もう一度右クリックにて [開く] ボタンをクリックしアプリケーションを実行してください。
- ※: 2回目以降アプリケーションを立ち上げる場合は通常のアプリケーションを立ち上げると同じ動作にて立ち上げることができます。



11.5. シート名をマニュアル通りに設定しても読み込めない場合

「10.3.解析する設計書（仕様書）のシート名について」の内容通りに設定しても「指定された Excelbook に CodeAGI が解析できるシート名が存在しません。シート名のルールは、「設定」...」というエラーが稀に発生する場合がございます。その大半の原因は Excel シートの破損(※1)が原因です。

その場合は新しい Excel Book を作成し、全てのシートを張りなおして Excel Book を再作成してください。

- (※1) Excel の自動修復機能により Excel シートの破損が見て取れない場合もございます。
- お手数ですが上記手順にて Excel Book の再作成をお願い致します。