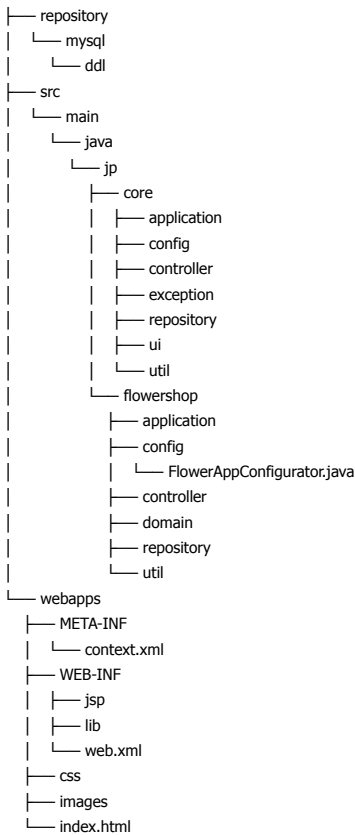


# デモアプリケーション解説

## 全体像

- 1. 主なファイル構成
  - ルートフォルダ
    - ビルド(gradle)設定ファイル
  - src フォルダ
    - Javaプログラムを格納
    - src/main/javaがクラスパスのルートディレクトリ
    - jp.core.\*の階層はPure Javaの簡易フレームワーク
    - jp.flowershop.\*の階層は花屋さんの販売管理のデモアプリ
  - repository フォルダ
    - mysql/ddl 販売管理用のDBテーブル及びデータ生成用
  - webapps フォルダ
    - META-INF JavaからMySQLへの接続設定
    - WEB-INF Webアプリ設定、JSPファイル 実行負ライブラリ
    - Webコンテンツ HTML image CSS JavaScript



- 2. Javaプログラムのパッケージ構成
  - アプリケーションはクリーンアーキテクチャ、ドメイン駆動設計をベースに構築しており、jp.coreパッケージは簡易フレームワークになっている。Springの構成とも比較的似た構成になっている。

## システムを構成するパッケージの役割

- ui
  - Webからの処理を捌きコントローラやビューとなるJSP、画面遷移などのルーティングを担うメインサーブレット
  - [Spring]DispatcherServletが担う。Spring Bootではpom.xmlでのマッピングも可能だがアノテーションが書かれたクラスを自動スキャンして設定ファイルが不要になる様にしている
- controller
  - Webベースの一つのユースケースを必要なアプリケーションに委譲し組み合わせることで実現する

- [Spring]@Controllerアノテーションを記載したControllerクラスが同じ役割を担う。画面でもAPIでもエンドポイントとControllerを対応させる
- application
  - 複数のビジネスルールとビジネス情報を組み合わせ特定領域で完結するビジネス機能を提供し情報の保管や整合性に責任を持つ
  - [Spring]Spring BootではServiceアノテーションを記載したServiceクラスが同様の役割。デモアプリではControllerがインスタンスを生成しているがSpringでは自動生成されるため自分で開発したコンストラクタで生成してしまうとリポジトリ処理の自動生成や連携が受けられないため注意
- domain
  - 業務特有のビジネスルールと必要な情報の扱いを担う。ITや個別のユースケースに依存させない。
  - [Spring]同様の実装。SpringでもLombokを利用したメソッドの自動生成は有効。データを保管する場所としてだけで実装されるケースが多いが、データとビジネスルールがセットで実現されるとアプリケーション（サービス）からの利用シーンで再利用性が高まる。
- repository
  - アプリケーションが指示するタイミングで情報をデータベース、ファイル、APIと連携し永続化
  - 
  - [Spring]一般的なデータアクセスの昨日は大半が自動生成されすぐに利用できるようになる。
- config
  - 技術変化の大きいUI固有のユースケースを担うController及び永続化をするrepositoryの領域と、ビジネスドメイン固有のユースケースとルール・情報を担うビジネスアプリケーション、ビジネスドメインの領域を組合せ切替するための設定を担う
- exception
  - システム共通の例外処理を担う
- util
  - ログや日付処理などの全体で共通する処理を担う

## 1. デモアプリケーションの動き

