

ファイル名 ※拡張子は「. java」である	Member_Info
対応するデータベーステーブル	Member_Info
所属パッケージ名	com. springproject. dockerspring. entity. NormalEntity

特記事項

コンストラクタやゲッターセッターを実装するライブラリとして、「lombok」を用いる。

メンバ変数は、いずれもアクセス修飾子を「private」とする。

「lombok」で、「ゲッター」「セッター」「引数無しコンストラクタ」「全てのメンバ変数に対する引数ありコンストラクタ」を実現する。

メンバ変数名とデータ型は、対応するテーブルの「**カラム名と同一**」にする事。

コンストラクトの処理内容に関しては複雑な処理は行わない為、処理内容は文章での記述にとどめる。

メソッド「makeMap」で画像を出力するときは、Base64形式に変換して出力する事。

実装インターフェース

EntitySetUp

DIするクラス

PhotoBase64Encode

実装メソッド

stringSetNull

makeMap

対になっているフォームクラス

MemberInfoForm

使用列挙型

Member\_Info\_Enum

DateFormat\_Enum

対になっている履歴用エンティティ

Member\_Info\_History

追加コンストラクタ

1番目処理内容

対になっている履歴用エンティティから、この通常データ用エンティティにデータを移し替える。

移し替えた後、Null初期化を実行する。

2番目処理内容

対になっている履歴用エンティティから、この通常データ用エンティティにデータを移し替える。

移し替えた後、Null初期化を実行する。

3番目処理内容

CSVリストから抽出したデータをマップリストにしたものを、このエンティティに移し替える。

CSVマップリストからデータを抽出し、それぞれ対応するデータ型に変換する。

移し替えた後、Null初期化を実行する。

CSVリストから抽出したデータはすべて新規追加扱いになるため、SpringDataJDBCの仕様に伴いシリアルナンバーは「Null」となる。

追加内部クラス

PhotoBase64EncodeContext

画像データのBase64変換のクラスを手動でDIし、このエンティティクラス内で用いれるようにする。

このクラスを設けた理由としては、通常インジェクションの際はフィールド変数にインジェクションするが、このエンティティに関しては余計なフィールドを追加すると、データベースのテーブルとマッピングできなくなり設けることができない為。

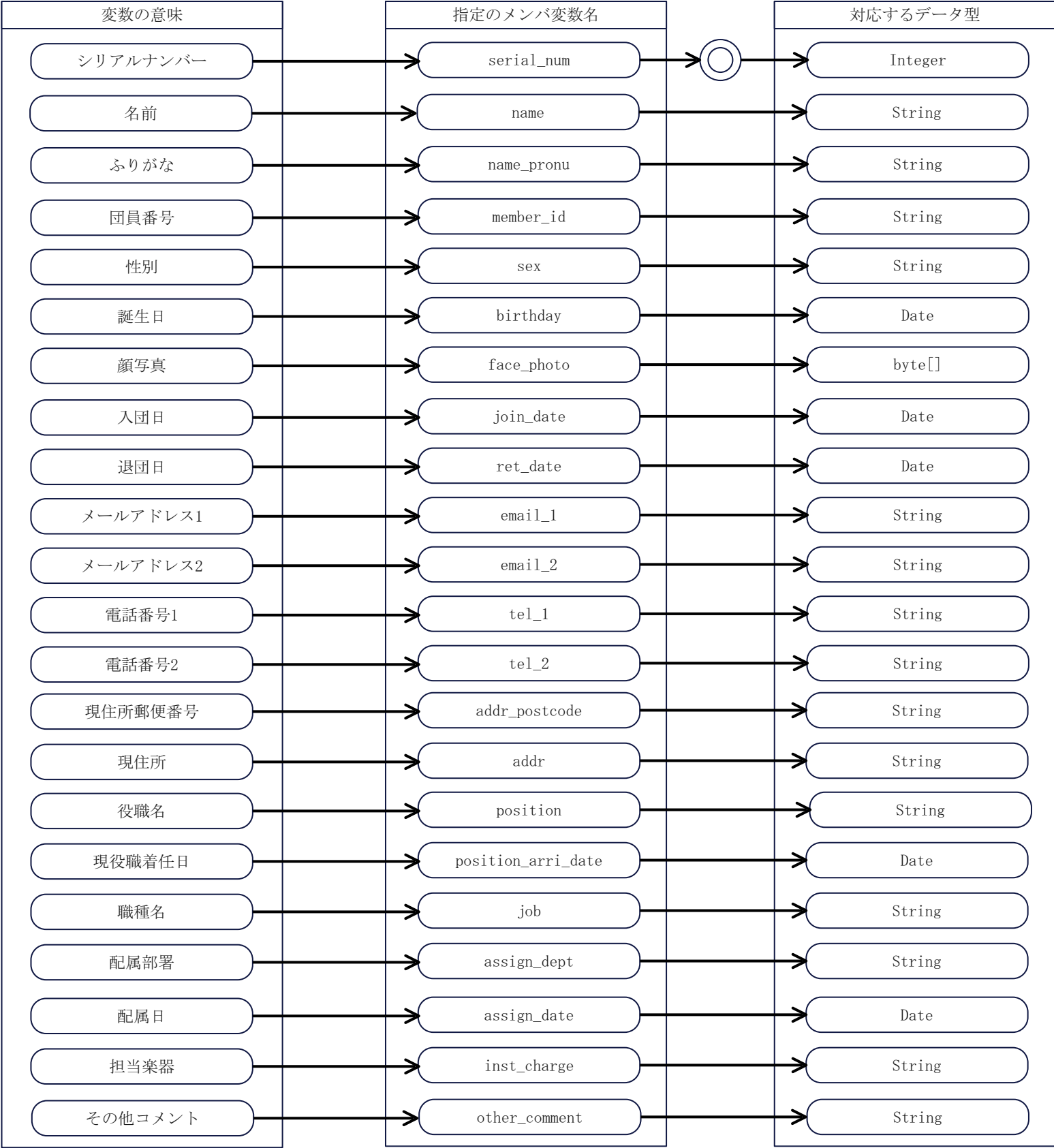
そのため独自にDIコンテナのコンテキストを取得して直接取得することによって、フィールド変数を用いずインジェクションを行う事が可能になる。

実装インターフェース

ApplicationContextAware

DIするクラス

PhotoBase64Encode



機能概要

主に[団員情報]機能のデータの、データベース保存やデータのやり取りに使用するエンティティである。

使用法としては、対象テーブルへのデータの追加更新処理、エンティティ内のデータの加工、他の関連するフォームクラスや履歴用エンティティクラスからの変換を行う。

なお、各メソッドのフローチャートに関しては、根本的な処理は他のエンティティと変わらないことから、インターフェースの設計書に統一するため、そちらを参照の事。