

プロジェクト名:〇〇会社製品開発システム

機能名: ユーザー情報一覧リポジトリクラス

企業名: B-FORME

所属: B-FORME第1Java開発部隊

責任者:

Copyright 2021 B-FORME inc.

プロジェクト名	〇〇会社製品開発システム	機能名	ユーザー情報一覧リポジ	作成日	2024年8月15日
				更新日	2024年8月15日

No	年月日	内容	更新者
1	2024年8月15日	新規作成	垣口
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			

プロジェクト名	〇〇会社製品開発システム	機能名	ユーザー情報一覧リポジトリクラ	作成日	2024年8月15日
				更新日	2024年8月15日

No	クラス名（論理名）	クラス名（物理名）	内容	備考
1	リポジトリ	Bfmk02Repository	ユーザー情報一覧のリポジトリクラス	

No	メソッド名（論理名）	メソッド名（物理名）	内容	備考
1	権限チェック	getAuthDiv	指定された CommonDto の情報を基に、対象ユーザーの権限区分を取得する	
2	条件検索	searchInfo	SearchForm と PaginationDto の情報を基に、条件に一致するユーザー情報をデータベースから取得する	
3	削除	delete	指定されたユーザーIDに基づき、該当ユーザーの削除フラグを更新して論理削除を行う	
4	検索条件	createdCriteria	検索条件のSQL文と条件パラメータを生成する。最終的なSQLの WHERE 句を作成し、動的に検索条件を構築する	
5	全件数取得	getTotalCount	SearchForm に基づく検索条件を使用して、削除フラグが立っていない全件数を取得する	
6	全From日取得	getAllFromDate	user_info テーブルから expire_date_from カラムの全てのレコードを取得し、リストとして返却する	
7				
8				
9				
10				

詳細設計_仕様

プロジェクト名	〇〇会社製品開発システム	機能名	ユーザー情報一覧	作成日	2024年8月15日
				更新日	2024年8月15日

引数（論理名）		メソッド名（論理名）	引数（物理名）		メソッド名（物理名）
IN	共通Dto	権限チェック	IN	commonDto	getAuthDiv
OUT	対象の権限区分		OUT	String	

■指定された CommonDto の情報を基に、対象ユーザーの権限区分を取得する

1.SQLクエリの作成

(1) ユーザー情報テーブル user_info から、指定された工場CD、部署ID、ユーザーIDに対応する権限区分 (auth_div) を取得するSQL文を作成。

2.SQL実行

(1) jdbcTemplate.queryForMap メソッドを使用して、commonDto の工場CD (facCd)、部署ID (affId)、ユーザーID (userId) をSQLクエリのパラメータとして指定し、結果を取得

3.権限区分の取得

(1) 結果として返されたデータから、権限区分 (auth_div) を取り出し、returnValue に格納

4.終了処理

(1) 結果の返却

・取得した権限区分 (returnValue) をそのまま呼び出し元へ返す

引数（論理名）		メソッド名（論理名）	引数（物理名）		メソッド名（物理名）
IN	共通Dto	条件検索	IN	commonDto	searchInfo
IN	ページネ이션Dto		IN	pageDto	
OUT	ユーザー情報リスト		OUT	List<Map<String,Object>>	

■SearchForm と PaginationDto の情報を基に、条件に一致するユーザー情報をデータベースから取得する

1.SQLクエリの作成

- ・所属ID (affiliate_id)
- ・所属名 (affiliate_name)
- ・ユーザーID (user_id)
- ・ユーザー名 (user_name)
- ・有効日 (From) (expire_date_from)
- ・有効日 (To) (expire_date_to)
- ・権限区分 (auth_div): 1~4の値に応じて、'ゲスト', '一般', '管理者', 'admin' の文字列を設定
- ・参照権限フラグ (watch_auth_flg): true/false に応じて '有'/'無' を設定
- ・操作権限フラグ (opr_auth_flg): true/false に応じて '有'/'無' を設定
- ・パスワード (pass)

2.結合条件

(1) userInfoテーブルと belonging テーブルを affiliate_id で LEFT JOIN し、所属情報を取得

3.削除フラグ条件

(1) 削除されていない (del_flg = '0') ユーザーのみを対象とする

4.検索条件の追加

(1) メソッド joukenSql(form) を呼び出し、動的に追加される検索条件 (WHERE 句) とその条件に対応するパラメータを取得。
SQL文に追加し、条件パラメータを joken に格納

5.ページング処理

(1) SQL文の末尾に ORDER BY u.affiliate_id,u.user_id による並び替えと、LIMIT と OFFSET を使用したページング処理を追加。
ページング情報 (offset, limit) を pageDto から取得し、joken に設定

6.クエリの実行

(1) 最終的に生成されたSQL文と条件パラメータ (joken) を用いて namedJdbc.queryForList(sql, joken) を実行し、検索結果を取得

7.結果の返却

(1) 削除されていない (del_flg = '0') ユーザーのみを対象とする

8.終了処理

(1) 検索の結果をListに格納し呼び出し元へ返す

詳細設計_仕様

引数（論理名）		メソッド名（論理名）	引数（物理名）		メソッド名（物理名）
IN	削除対象のID	削除	IN	userId	delete
OUT	対象の権限区分		OUT	int	

■指定されたユーザーIDに基づき、該当ユーザーの削除フラグを更新して論理削除を行う

1.SQLクエリの作成

(1)user_info テーブルの指定された user_id に該当するレコードの削除フラグ (del_flg) を '1' に更新するSQL文を生成。

2.SQLの実行

(1)jdbcTemplate.update メソッドを用いて、生成したSQL文とパラメータとして指定された userId を実行。該当するレコードの削除フラグを更新

3.結果の返却

(1)更新が正常に行われた場合、更新されたレコード数を返す

引数（論理名）		メソッド名（論理名）	引数（物理名）		メソッド名（物理名）
IN	INパラメータ	検索条件	IN	form	createdCriteria
OUT	SQL文とパラメータのMap<Object>		OUT	List<Object>	

■SearchForm の各項目に基づいて、検索条件のSQL文と条件パラメータを生成する。最終的なSQLの WHERE 句を作成し、動的に検索条件を構築する

1.SQL文と条件パラメータ用のオブジェクトの初期化

(1)空の sql 文字列と、条件パラメータを保持する Map<String, Object> オブジェクト (joken) を初期化

2.各検索条件のチェックとSQLの組み立て

(1)所属IDのチェック

・ SearchForm から affiliateld が指定されている場合、sql に AND u.affiliate_id = :affiliateld を追加し、joken に affiliateld を設定

(2)ユーザーIDのチェック

・ userId が指定されている場合、sql に AND u.user_id = :userId を追加し、joken に userId を設定

(3)ユーザー名のチェック

・ userName が指定されている場合、sql に AND u.user_name LIKE :userName を追加し、joken に userName を部分一致検索用に % で囲んだ値を設定

(4)権限区分のチェック

・ authDiv が設定され、0 または 3 の場合を除く場合、sql に AND u.auth_div IN (:authDiv) を追加し、joken に authDiv のリストを設定

(5)有効日 (From) のチェック

・ expireDateFrom が指定されている場合、sql に AND u.expire_date_from = :expireDateFrom を追加し、joken に expireDateFrom を設定

(6)有効日 (To) のチェック

・ expireDateTo が指定されている場合、sql に AND u.expire_date_to = :expireDateTo を追加し、joken に expireDateTo を設定

3.結果の格納と返却

(1)終了処理

・ 最終的に作成されたSQLの条件部分 (sql) と、対応するパラメータを保持する joken を returnObject リストに追加し、返却する

引数（論理名）		メソッド名（論理名）	引数（物理名）		メソッド名（物理名）
IN	INパラメータ	全件数取得	IN	form	getTotalCount
OUT	全件数		OUT	int	

■SearchForm に基づく検索条件を使用して、削除フラグが立っていない全件数を取得する

1.SQL文の作成

(1)user_info テーブルと belonging テーブルを結合し、削除フラグ (del_flg) が '0' であるレコードをカウントするためのSQL文を生成

2.検索条件の追加

(1)検索条件を動的に生成するメソッド joukenSql(form) を呼び出し、返却されたSQL条件文と対応する条件パラメータ (joken) をSQL文に追加

・ joukenSql(form) は、SearchForm の各フィールドに基づいてSQL条件文と検索パラメータを動的に生成

3.SQL文の実行

(1)namedJdbc.queryForObject メソッドを使用し、生成したSQL文を条件パラメータとともに実行。取得したレコード件数を countAll として取得

4.結果の返却

(1)レコードの総件数 (countAll) を返却

詳細設計_仕様

引数（論理名）		メソッド名（論理名）	引数（物理名）		メソッド名（物理名）
IN	-	全From日取得	IN	-	getAllFromDate
OUT	全From日の件数		OUT	List<Map<String,Object>>	

■user_info テーブルから expire_date_from カラムの全てのレコードを取得し、リストとして返却する

1.SQL文の作成

(1)user_info テーブルから expire_date_from カラムのデータを取得するためのSQL文を生成

2.SQL文の実行

(1)jdbcTemplate.queryForList メソッドを使用して、SQL文を実行し、expire_date_from のデータを取得。結果を days というリストに格納

3.結果の返却

(1)取得した expire_date_from のリスト (days) を呼び出し元に返却