

プロジェクト名: B-FORME ○○工場開発システム

機能名: ユーザー情報一覧サービスクラス

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	B-FORME ○○工場開発システム	機能名	ユーザー情報一覧サービス	作成日	2024年8月6日
				更新日	2024年8月6日

No	年月日	内容	更新者
1	2024年8月6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	B-FORME ○○工場開発システム	機能名	ユーザー情報一覧サービスクラス	作成日	2024年8月6日
				更新日	2024年8月6日

No	クラス名（論理名）	クラス名（物理名）	内容	備考
1	サービス	Bfmk02Service	ユーザー情報一覧のサービスクラス	

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	getAuthority	参照・操作権限の有無をチェックする	
2	ユーザー検索	getAllUserInfo	検索フォームとページング情報を基に、該当するユーザー情報を取得し、UserInfoD	
3	全レコード数取得	etAllCount	指定された検索条件に基づき、該当する全レコード数を取得する	
4	ユーザー削除	deleteUser	ユーザーを削除する	
5	日付け変換	isDate	日付けの変換と入力チェック	
6	未来日チェック	futureDateCheck	入力値が未来日でないかのチェック	
7	最も古い日付けを取得	getOldestDate	DBから取得した日付リストの中で、最も古い日付を取得する	
8				
9				
10				

プロジェクト名	B-FORME ○○工場開発システム	機能名	ユーザー情報一覧サービスクラス	作成日	2024年8月6日
				更新日	2024年8月6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	ユーザー情報DTO	権限チェック	IN	Dto	getAuthority
OUT	権限区分有無確認処理		OUT	repository.getAuthDiv	

- 参照・操作権限の有無をチェックする
- 1.呼び出し元ヘリポジトリクラスからパブリックメソッド、権限チェックを呼び出し返す

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	ユーザー情報DTO	ユーザー検索	IN	Dto	getAllUserInfo
OUT	ユーザー情報DTO		OUT	Dto	

- 検索フォームとページング情報を基に、該当するユーザー情報を取得し、UserInfoDto オブジェクトのリストとして返す。
- 1.戻り値の初期化
- (1)空の UserInfoDto のリストを returnDtoList として初期化。

- 2.検索実行
- (1) リポジトリクラスの全ユーザー情報取得メソッドを呼び出し

呼出メソッド名	引数1
全ユーザー情報取得	ユーザー情報DTO

- (2)formとpageDtoを基にユーザー情報のリストを取得

- 3.Dtoに値設定
- (1) 取得したリストから、各ユーザー情報を UserInfoDto オブジェクトに設定していく
- ・工場CD (facCd): 固定値 "bfm1"
 - ・所属ID (affilicateId)
 - ・所属名 (affilicateName)
 - ・ユーザーID (userId)
 - ・ユーザー名 (userName)
 - ・所属人数 (count)
 - ・権限区分 (authDiv)
 - ・参照権限フラグ (watchAuthFlg)
 - ・操作権限フラグ (oprAuthFlg)
 - ・パスワード (pass)
 - ・適用日（FROM）(expireDateFrom)
 - ・適用日（TO）(expireDateTo, 必要時のみ)
 - ・日付のフォーマット設定
 - ・適用日 (expire_date_from, expire_date_to) は "yyyy年MM月dd日" のフォーマットに変換して設定する

- 4.リストへの追加
- (1)各ユーザー情報を設定したUserInfoをreturnDtoListに追加

- 5.終了処理
- (1) 全ユーザー取得メソッドで取得したユーザー情報DTOを呼び出し元へ返却し、処理を終了する。

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	検索フォーム	全レコード数取得	IN	SearchForm	getAllCount
OUT	全件数		OUT	int	

■指定された検索条件に基づき、該当する全レコード数を取得する

1.レコード数の取得

- (1)リポジトリクラスのパブリックメソッド の allCountSql を呼び出す
- ・ form の条件に基づく全レコード数を取得し、allCou に格納

2.終了処理

- (1)取得したレコード数をallCouをそのまま返す

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	ユーザー情報DTO	ユーザー削除	IN	Dto	deleteUser
IN	ユーザーID		IN	ArrayList	
OUT	-		OUT	-	

■ユーザーを削除する

1.削除対象ユーザー確認処理

- (1) 削除対象にチェックがあるか確認

<条件1> 該当データが無い場合

- ・ エラーメッセージを表示する。

<条件2> 1件以上の場合

- ・ 削除処理を行う。

2.削除処理

- (1) リポジトリクラスのパブリックメソッド「削除」を呼び出す。

呼出メソッド名	引数1
削除	ユーザーID

- ・ ループを使用して、IDリスト内の各ユーザーID (idNum) を取り出し削除処理を行う

3.終了処理

- (1) 対象のユーザーIDがすべて削除される。

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	日付け	日付け型変換	IN	Date	isDate
OUT	真偽値		OUT	boolean	

■日付けの変換と入力チェック

1.空文字のチェック

- (1) 日付けの入力値の有無をチェック

<条件1> 値が未入力の場合

- ・ 呼び出し元に真偽値(false)を返す

<条件2> 値が入力されている場合

- ・ 次の処理へ

2.日付けのフォーマット指定

3.フォーマット処理

- (1) フォーマットしtrueを返す成功時、trueを返す
- (2) フォーマット失敗時、falseを返す

3.終了処理

- (1) フォーマットの結果を呼び出し元に真偽値で返す

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	日付け	未来日チェック	IN	fromDate	futureDateCheck
IN	日付け		IN	toDate	
OUT	真偽値		OUT	boolean	

■入力値が未来日でないかのチェック

1.日付フォーマット指定

- (1) yyyy-MM-dd 形式の日付フォーマットで厳密なチェックを行うように設定。

2.日付の解析

- (1) 受け取った fromDate と toDate を指定フォーマットでパースし、日付オブジェクトを生成。

3.日付の解析

- (1) fromDate が toDate より未来であるかを判定し、結果を返す。

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	-	最も古い日付けを取得	IN	-	getOldestDate
OUT	日付け		OUT	boolean	

■DBから取得した日付リストの中で、最も古い日付を取得する。

1.初期値の設定

- (1) oldest_date に最大値 Long.MAX_VALUE を設定し、日付比較の初期値とする。

2.DBから日付リストの取得

- (1) リポジトリクラスのパブリックメソッド から initial_enabled_date() を呼び出し、日付リストを取得。

3.日付けの比較

- (1) リスト内の各レコードから expire_date_from を取得し、oldest_date よりも古い日付が存在する場合、その日付で oldest_date を

4.結果の返却

- ・最も古い日付を oldest_date として返す。

