loggerについて

Controllerの場合

独自のLoggerWapperを使用（*resources¦i18n¦event.properties*から定型文を取得できる）

```
private static final LoggerWapper logger = LoggerWapperFactory.getLogger(クラス名);
```

使い方

以下のMethodを用意する。クラスの初期化で呼ばれるので、他の処理を入れてもよい。

```
@PostConstruct
protected void init() {
    logger.setMessageSource(messageSource);
}
```

使う場所

Methodの始まり

```
logger.infoCode("I0001"); // I0001=メソッド開始:{0}
```

Methodの終り(return直前)

```
logger.infoCode("I0002", xxxx); // I0002=メソッド終了:{0}
return xxxx;
```

正常の場合

```
logger.infoCode("Ixxxx", yyyy);
```

I1001=ログイン成功。{0}
I1002=ログアウトしました。{0}
I1003=削除しました。{0}
I1004=更新しました。{0}
I1005=パスワードを更新しました。{0}
I1006=削除ユーザ: {0}
I1007=処理中です。しばらくお待ちください。{0}

警告の場合

```
〜〜 //処理結果
logger.warnCode("Wxxxx", yyyy);
```

W1001=該当するデータはありません。{0}
W1002={0}は{1}の為、更新できません。
W1003={0}は別のユーザーによって更新されている為、削除できませんでした。
W1004={0}は{1}の為、削除できませんでした。
W1005=旧パスワードが正しくありません。{0}
W1006=データが存在しないか指定のデータは別のユーザーによって更新されている為、更新できませんでした。{0}
W1007={0}を確認してください。
W1008={0}から再度、お試しください。

ERRORの場合

throwする場合＝ERRORの場合。※ただしexceptionをcatchして処理する場合はInfo,Warnを使用してください。

```
〜〜 //処理結果
logger.errorCode("E0014", ex.getMessage()); // E0014=メソッド異常終了:{0}
throw ex;
または
throw new ApplicationException(xxxxx);
```

exceptionを埋め込み

```
〜〜 //処理結果
logger.errorCode("Exxxx", ex);
```

E1001={0}は別のユーザーによって更新されている為、更新できませんでした。
E1002=選択した{0}は対応しておりません。
E1003=データが正常でない為、実行できません。
E1004=正しくアクセスが行われなかった為、エラーが発生しました。
E1005=ページのデータ保持期限が切れています。
E1006=権限がありません。{0}
E1007=登録に失敗しました。{0}
E1008=更新に失敗しました。{0}
E1009=削除に失敗しました。{0}
E1010=処理回数をオーバーです。{0}
E1011=ダウンロードに失敗しました。{0}
E1012=ファイルのアップロードに失敗しました。{0}
E1013=エラーとなりました。{0}

DEBUG、TRACEの場合

isDebugEnabled()、isTraceEnabled()で囲むこと。

```
if (logger.isDebugEnabled()) {
   logger.debug(〜〜〜〜);
   logger.debug(〜〜〜〜);
}
```

メッセージを増やしたい場合

研川に連絡する。event.propertiesに追記します。

ServiceImplの場合

Controllerにbooleanなどで、ERROR内容を返さない場合に記載する。
例：
Update、Delete失敗。Controllerにはfalseをreturnする場合など。
Insertの場合は戻り値は、成功=シーケンスKey、失敗＝nullなど。

Repositoryの場合

不要

その他

CoreやUtilに関わるClassの場合は
通常のLoggerを使用　※propertiesからメッセージを取得しないTYPE。

```
private static final Logger logger = LoggerFactory.getLogger(クラス名);
```

以上です。