Nimbusコードマスタ仕様書

2016年9月20日

目次

[1. 概要 1](#_Toc462410354)

[2. インタフェース設計 3](#_Toc462410355)

[2.1. CodeMasterFinder 3](#_Toc462410356)

[2.2. PartUpdate 4](#_Toc462410357)

[3. 実装設計 6](#_Toc462410358)

[3.1. CodeMasterFinder 6](#_Toc462410359)

[3.1.1. jp.ossc.nimbus.service.codemaster.CodeMasterService 6](#_Toc462410360)

[3.1.2. jp.ossc.nimbus.service.codemaster.CodeMasterFinderGroupService 10](#_Toc462410361)

# 概要

コードマスタとは、ある時点でのメモリ外のマスタ情報を、メモリ中に読み取って保持しておき、アプリケーションに高速に提供する機能である。また、メモリ外のマスタ情報を変更した場合は、更新通知を送る事で、再読み込みを行うことができる。その際、読み取ったマスタ情報を、いつメモリ中に反映させるかの時間指定を行うこともできる。この機能を提供するインタフェースがCodeMasterFinderである。

CodeMasterFinder

マスタ取得

マスタ情報

マスタ情報

マスタ読み込み

データベース

など

更新通知

マスタ更新

マスタの更新においては、３種類の方法がある。

対象のマスタ名を通知する事で、マスタの全件再読み込みを促す方法。マスタの全体件数が多く、更新件数が少ない場合、全てのデータを読み直すのは非効率なため、差分情報のみを通知して、更新処理を効率的に行う方法。その差分更新において、差分レコードのキー情報と更新種別（追加、変更、削除）のみを含めて、マスタの差分再読み込みを促す方法と、同時に変更された値も通知し、マスタを再読み込みする事なく差分を当てる方法がある。但し、差分更新を利用するためには、メモリ中のマスタ情報となるオブジェクトがPartUpdateインタフェースを実装する必要があり、差分情報としてPartUpdateRecordsを通知する必要がある。

また、マスタを参照する際に、一連の処理の流れの中で、マスタを複数回参照する場合、マスタが途中で更新されて値が変わってしまう読み取り一貫性の問題が発生する。Nimbus AOPのThreadContextInitializeInterceptorService、NimbusコンテキストのThreadContextServiceとコードマスタを組み合わせる事で、処理開始時点のコードマスタをスレッド単位のコンテキスト情報として保持しておき、そこからアプリケーションがマスタ情報を参照する事で、読み取り一貫性を保証する事ができる。

CodeMasterFinder

アプリケーションＡ

ThreadContextInitialize

InterceptorService

ThreadContextService

マスタ情報

マスタ情報

アプリケーションＢ

JavaDocは、以下を参照。

<http://nimbus.osdn.jp/reports/apidocs/jp/ossc/nimbus/service/codemaster/package-summary.html>

# インタフェース設計

## CodeMasterFinder

マスタ情報をマスタ名と紐づけて管理し、アプリケーションに提供するインタフェースである。また、マスタの更新を通知して、マスタ情報の再読み込みを行う機能も提供する。

getCodeMasters()

**CodeMasterFinder**

Map

マスタ情報

マスタ名

updateCodeMaster(マスタ名)

以下に、ソースコードの例を示す。

import jp.ossc.nimbus.core.\*;

import jp.ossc.nimbus.service.codemaster.\*;

CodeMasterFinder finder = (CodeMasterFinder)ServiceManagerFactory.getServiceObject(“CodeMasterFinder”);

Map masters = finder.getCodeMasters();

Object masterA = masters.get(“マスタA”);

以下に、CodeMasterFinderインタフェースの実装クラスの一覧を示す。

|  |  |  |
| --- | --- | --- |
| # | 実装クラス名 | 概要 |
| 1 | jp.ossc.nimbus.service.codemaster.CodeMasterService | デフォルト実装クラスである。  マスタの更新通知を、ローカルからのメソッド呼び出しに加え、JMSまたはNimbusパブリッシャで、リモートから通知を受ける事ができる。 |
| 2 | jp.ossc.nimbus.service.codemaster.CodeMasterFinderGroupService | CodeMasterFinderサービスをグルーピングする実装クラスである。 |

## PartUpdate

コードマスタで管理するマスタ情報が、差分更新をサポートする場合に実装すべきインタフェースである。

PartUpdate

CodeMasterFinder

cloneAndUpdate(records)

PartUpdateRecords

updateCodeMaster(マスタ名、records)

PartUpdateRecords

差分情報

**PartUpdate**

**マスタ情報**

addRecord(key)

CodeMasterUpdateKey

差分情報であるPartUpdateRecordsは、差分のキー情報と更新種別（追加、変更、削除）を表すCodeMasterUpdateKeyをaddRecord(key)メソッドで追加する事で、差分を伝える。この方法でマスタを更新する場合は、更新種別が追加や変更の時、追加または変更された値を、CodeMasterUpdateKeyのキー情報を元に、データベースなどのメモリ外のマスタから検索し読み込む必要がある。

また、PartUpdateRecordsのaddRecord(key, value)メソッドで、CodeMasterUpdateKeyと、それに該当する値を対にして登録する方法もある。この方法で、マスタ更新を行う場合は、PartUpdateRecordsを受け取ったCodeMasterFinderで、メモリ外のマスタからデータを取得する必要なく、更新する事ができる。

マスタ情報となるオブジェクトがPartUpdateインタフェースを実装している場合、マスタ更新メソッドupdateCodeMaster()の第２引数で、差分情報を表すPartUpdateRecordsを渡す事で、差分更新が可能となる。差分情報であるPartUpdateRecordsを受け取ったCodeMasterFinderは、現在のマスタ情報であるPartUpdateに対して、cloneAndUpdate()メソッドで差分情報であるPartUpdateRecordsを渡して、その差分情報を取り込んだ後のPartUpdateを取得し、次のマスタ情報として使用する。このメソッドの実装は、読み取り一貫性の担保のために、現在のマスタ情報を壊さないように、差分以外のデータを複製し、差分部分を当て込んだ別のインスタンスを返す必要がある。

以下に、PartUpdateインタフェースの実装クラスの一覧を示す。

|  |  |  |
| --- | --- | --- |
| # | 実装クラス名 | 概要 |
| 1 | jp.ossc.nimbus.beans.dataset.RecordList | 動的構造化データを表すクラス群を持つdatasetパッケージの、２次元構造データを表すクラスである。 |
| 2 | jp.ossc.nimbus.recset.RecordSet | 簡易なO/Rマッピング機能を持つ２次元構造データを表すクラスである。 |

# 実装設計

インタフェース設計で示した各インタフェースのうち、機能性のあるインタフェースの実装クラスの機能を説明する。

## CodeMasterFinder

### jp.ossc.nimbus.service.codemaster.CodeMasterService

CodeMasterFinderインタフェースのデフォルト実装クラスである。

マスタ取得

業務フロー

XML定義ファイル

**CodeMasterService**

invokeFlow(input)

マスタ情報

マスタ情報

BeanFlowInvokerFactory

createFlow(マスタ名)

BeanFlowInvoker

BeanFlowInvoker

データベース

など

マスタ読み込み

コードマスタ上に管理するマスタ情報の取得は、データベースから読み取ったり、ファイルから読み取ったり、様々な実装方法が想定されるので、その実装をNimbus業務フローに委譲し、業務フローの戻り値をマスタ情報とする。サービスの開始時に、マスタ名と同名の業務フローを呼び出し、その戻り値を初期状態のマスタ情報として、メモリ上に管理する。

ローカルJavaVM

**CodeMasterService**

MessageReceiver

または

JMSMessageConsumerFactory

updateCodeMaster(マスタ名…)

マスタ情報

任意のマスタ更新

アプリケーション

invokeFlow(input)

BeanFlowInvoker

データベース

など

リモートJavaVM

任意のマスタ更新

アプリケーション

setMasterFlowKey(マスタ名)

send()

ServerConnection

または

JMSSessionFactory

CodeMasterNotifyBean

マスタ情報の更新に関しては、ローカルから更新する場合は、updateCodeMaster()メソッドを直接呼び出す。リモートから更新する場合は、Nimbusパブリッシャ、またはJava EE規格のJMS（**J**ava **M**essage **S**ervice）を使って、リモートから更新通知を行うことでupdateCodeMaster()メソッドを間接的に呼び出す。その場合は、CodeMasterNotifyBeanを使って、更新通知を行う事ができ、更新情報としてマスタ名はもちろん、反映日時、差分情報等も指定できる。

また、マスタの取得元であるデータベースなどの外部リソースとの切り離しのために、メモリ上のマスタ情報をファイルに保存しておき、サービスの起動時に、ファイルからマスタ情報を復元する機能もある。但し、マスタ情報が、直列化可能である必要性がある。

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | MasterNames | String[] | マスタ名の配列を設定する。  ここで指定するマスタ名は、getCodeMasters()で取得できるマスタ情報のMapのキー名として使用される。また、そのマスタを取得する業務フロー名も兼ねる。 |
| 2 | BeanFlowInvokerFactoryServiceName | ServiceName | 業務フローを生成するjp.ossc.nimbus.service.beancontrol.interfaces.BeanFlowInvokerFactoryサービスのサービス名を設定する。 |
| 3 | MessageReceiverServiceName | ServiceName | マスタの更新通知をNimbusパブリッシャから受信する場合に使用するjp.ossc.nimbus.service.publish.MessageReceiverサービスのサービス名を設定する。 |
| 4 | Subjects | String[] | マスタの更新通知をNimbusパブリッシャから受信する場合に使用する、受信サブジェクトを設定する。 |
| 5 | JMSTopicSubscriberFactoryServiceName | ServiceName | マスタの更新通知をJMSから受信する場合に使用するjp.ossc.nimbus.service.jms.JMSMessageConsumerFactoryサービスのサービス名を設定する。 |
| 6 | NotifyMasterNameMapping | Properties | マスタの更新通知で指定されたマスタ名とMasterNames属性で指定するマスタ名が異なる場合に、通知マスタ名とマスタ名のマッピングを設定する。 |
| 7 | StartMasterNames | String[] | サービスの開始時に取得するマスタのマスタ名配列を設定する。  指定しない場合は、MasterNames属性で指定した全てのマスタがサービスの開始時に取得される。 |
| 8 | NotStartMasterNames | String[] | サービスの開始時に取得しないマスタのマスタ名配列を設定する。  指定しない場合は、MasterNames属性で指定した全てのマスタがサービスの開始時に取得される。 |
| 9 | StartMasterInputMap | Map<String,Object> | サービスの開始時に取得するマスタのマスタ名と入力オブジェクトのマッピングを設定する。  サービス開始時のマスタ取得時に、入力が必要ないマスタの場合は、マッピングを設定する必要はない。 |
| 10 | PersistDir | String | マスタをファイルとして永続化するディレクトリを設定する。 |
| 11 | LoadOnStart | boolean | サービスの開始時にファイルに永続化されているマスタを読み込むかどうかを設定する。  ファイルから読み込まれた場合は、マスタ取得業務フローは実行しない。  デフォルトは、falseで読み込まない。 |
| 12 | SaveOnStop | boolean | サービスの停止時にマスタをファイルに永続化するかどうかを設定する。  デフォルトは、falseで永続化しない。 |

### jp.ossc.nimbus.service.codemaster.CodeMasterFinderGroupService

複数のCodeMasterFinder をグルーピングするCodeMasterFinderインタフェースの実装クラスである。

getCodeMasters()

**CodeMasterFinderGroupService**

CodeMasterFinder

CodeMasterFinder

getCodeMasters()を呼び出されると、グルーピングしているCodeMasterFinderに対して、それぞれgetCodeMasters()を呼び出し、結果をマージして返す。

updateCodeMaster()を呼び出されると、指定されたマスタ名のマスタを保持しているCodeMasterFinderに対して、それぞれupdateCodeMaster()を呼び出す。

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | CodeMasterFinderServiceNames | ServiceName[] | グルーピングするCodeMasterFinderサービスのサービス名を配列で設定する。 |