



ServerMessengerResolver

IServerMessenger



Broadcasts distributed cache notifications to servers, and manages the local server (notifications must be processed immediately)
`PerformRefresh(...)`, `PerformRefreshAll(...)`, `PerformRemove(...)`

ServerMessengerBase {abstract}



Base class for all messengers. Notifies cache refreshers immediately on the local server, and then messages remote servers

Core/WebServiceServerMessenger {legacy}



Sends remote notifications via webservice to all servers
Via `ServerSyncWebServiceClient`
To `CacheRefresher.asmx` - which notifies `ICacheRefreshers`

Core/BatchedWebServiceServerMessenger {abstract}



Queues remote notifications in a batch and sends them to all servers as one message per server, when flushed

Web/BatchedWebServiceServerMessenger {current default}

Flushes the batch at the end of the current http request

Core/DatabaseServerMessenger {abstract}



Stores remote notifications in the database
Manages database notifications to notify `ICacheRefreshers`

Core/BatchedDatabaseServerMessenger {abstract}



Queues remote notifications in a batch and store them as one message when flushed

Web/BatchedDatabaseServerMessenger {new}

Flushes the batch at the end of the current http request
Syncs the server at the start of each request (throttled)
Boots the server at the start of the application

TODO

`WebServiceServerMessenger` should not message the local server - see `WebServerUtility`

There should be a `Notify` method accepting a payload that would not be serialized on the local server, would replace `RefreshByJson`



Distributed Cache

v001 - 2/3 - Servers Registration

ServerRegistrarResolver

IServerRegistrar

Returns registrations as
IEnumerable<IServerAddress>

ConfigServerRegistrar

Uses settings (IServer) to return
ConfigServerAddress

DatabaseServerRegistrar

Uses ServerRegistrationService
to return ServerRegistration

IServer

Represents a server
in the settings, used by ConfigServerRegistrar
to initialize ConfigServerAddress

ServerElement

IServerAddress

Provides the address of
a server as a string

ConfigServerAddress

ServerRegistration

By default, servers must be registered manually in the config and then the ConfigServerRegistrar will list them

The new system have them registered automatically in the database for info only (because we just guess the address) though registration's not used anymore really

ServerRegistrationEventHandler

Auto-register servers for information
purposes (cannot be used for http)

ServerRegistrationService

ServerRegistrationRepository

ServerRegistrationMapper

ServerRegistrationFactory

ServerRegistrationDto



Distributed Cache

v001 - 3/3 - Refreshers

The envelope is what's used by the messenger to wrap the instructions with infos about the target servers. ServerMessengerBase takes care of notifying the ICacheRefresher instances immediately on the local server, then the actual messenger makes sure the notifications are sent to other servers. Either CacheRefresher.asmx (for the webService thing) or DatabaseServerMessenger (for the new DB thing) will then notify the ICacheRefresher instances on the remote servers.

RefreshInstructionEnvelope

Contains a list of target servers (IServerAddress), and a list of RefreshInstructions

RefreshInstruction RefreshMethodType

Describes what needs to be done, processed by CacheRefresher.asmx or DatabaseServerMessenger to notify the ICacheRefresher

CacheInstructionDto

Stored in the database, contains a timestamp, the origin identity, and RefreshInstructions

MessageType CacheUpdated

end-users subscribe to the CacheUpdated event of the proper refresher, to get notified of changes

ICacheRefresher

ICacheRefresher<TEntity>
IJsonCacheRefresher

CacheRefresherBase<TRefresher>

JsonCacheRefresherBase<TRefresher>
TypedCacheRefresherBase<TRefresher, TEntity>

(all our refreshers)

