

WNS Java API design (For Microsoft)

Author: Kula Shi
Distribution: Public
Version: 1.3

Revision Chart

Revision	Version	Comments	Date	Author	Reviewer
01	1.0	Initial	2015-06-29	Kula Shi	Scar Ai
02	1.1	Microsoft reviewed changes.	2015-07-09	Kula Shi	Torres
03	1.2	1. Add WNS message type pre-definition 2. Change close(), getMessage(), getMessageCount() to WNSManager	2015-07-10	Kula Shi	Alan Li
04	1.3	1. Add parameters for APIs close(), getMessage() and get MessageCount() 2. Add annotation.	2015-07-22	Kula Shi	Torres

Content

<i>Revision Chart</i>	2
1 <i>Java WNS APIs</i>	3
1.1 <i>WNSManager.java</i>	3
1.2 <i>WNSChannel.java</i>	4
1.3 <i>WNSNotificationMessage.java</i>	4
1.4 <i>WNSException.java</i>	5

1 Java WNS APIs

Java WNS (Microsoft Notification System) APIs design follows platform WNS APIs. Here four classes are introduced to fulfil WNS requirement.

1.1 WNSManager.java

This class is defined to create WNS channels, get push server status, etc. This class is a final class, cannot be extended, and cannot be instantiation.

Package:

com.ms.service.wns;

Class:

*public final class **WNSManager** extends java.lang.Object;*

Fields:

*public static final int **SERVER_STATUS_ON** = 1;*

*public static final int **SERVER_STATUS_OFF** = 0;*

Methods:

```
/**
 * Create a channel for specified appName and appKey;
 * @param appName, application name, it's specified by application; As agreement, appName
 * needs to be parsed out from application property "MIDlet-WNS-AppName" in JAD
 * file or MANIFEST file. Usually, applicatoin should use
 * getAppProperty("MIDlet-WNS-AppName") to get appName as default.
 * @param appKey, application key, it's specified by application;
 * @return , return an instance of WNSChannel;
 * @throws, create channel failure;
 */
public static WNSChannel synchronized createChannel(String appName, String appKey) throws
WNSException;

/**
 * Get push server status;
 * @return SERVER_STATUS_ON means push sever is available, SERVER_STATUS_OFF means
 * push server is not available;
 */
public static int getPushServerStatus();

/**
 * Close WNS channel by specified appKey;
 * @param appKey, application key, it's specified by application;
 * @return true means close success, otherwise failure;
 */
public static boolean close(String appKey);
```

```

/**
 * Get a WNS push message by specified appKey and index;
 * @param appKey, application key, it's specified by application;
 * @param index, the index in WNS message cache;
 * @return a WNS push message;
 */
public static WNSNotificationMessage getMessage(String appKey, int index);

/**
 * Get WNS push message count by specified appKey;
 * @param appKey, application key, it's specified by application;
 * @return WNS push message count;
 */
public static int getMessageCount(String appKey);

```

1.2 WNSChannel.java

This interface is defined an API set for WNS channel operations. It's only can be constructed by **createChannel** of **WNSManager**;

Package:

com.ms.service.wns;

Class:

*public interface **WNSChannel**;*

Fields:

None.

Methods:

*public String **getChannelUri**();*
*public String **getChannelExpireTime**();*

1.3 WNSNotificationMessage.java

This interface is defined a WNS notification message. It's only can be constructed by **getMessage** of **WNSChannel**. It defines an API set to get message information.

Package:

com.ms.service.wns;

Class:

*public interface **WNSNotificationMessage**;*

Fields:

/ Pre-define message types */*
public static final int WNS_TYPE_TOAST = 0;

```
public static final int WNS_TYPE_TILE = 1;
public static final int WNS_TYPE_BADGE = 2;
public static final int WNS_TYPE_RAW = 3;
```

Methods:

```
/**
 * Get message time;
 */
public String getTime();

/**
 * Return the notification message type. It can be WNS_TYPE_TOAST,
 * WNS_TPE_FILE, WNS_TYPE_BADGE, WNS_TYPE_RAW.
 */
public int getNotificationType();

/**
 * Get message content as a byte buffer.
 */
public byte[] getContents();
```

1.4 WNSException.java

This class is defined as a WNS exception. It will be thrown while failing to create a **WNSChannel** in **createChannel** of **WNSManager**;

Package:

```
com.ms.service.wns;
```

Class:

```
public class WNSException extends java.io.IOException;
```

Fields:

```
None
```

Methods:

```
public WNSException();
public WNSException(String msg);
```