```
typedef struct  push_notification_channel
{
   int channel_uri_length;
   char channel_id[CHANNEL_ID_MAX_LEN];
   char* channel_uri;
   char expiry_time[TIME_MAX_LEN];
}PUSH_NOTIFICATION_CHANNEL_T;

typedef struct  wns_push_notification
{
   char channel_id[CHANNEL_ID_MAX_LEN];
   char time[TIME_MAX_LEN];
   MSG_WNS_NOTIFICATION_TYPE_E notification_type;
   int content_length;
   char* contents;
}WNS_PUSH_NOTIFICATION_T;

typedef enum
{
   WNS_STATUS_SUCCESS,
   WNS_STATUS_NOMEMORY,
   WNS_STATUS_BADPARAM,
   WNS_STATUS_NOTREGISTERED,
   WNS_STATUS_NOTCONNECTED,
   WNS_STATUS_ALREADYCONNECTED,
   WNS_STATUS_INPROGRESS,
   WNS_STATUS_FAILED,
   WNS_STATUS_COMMON_ERROR
}MSG_WNS_STATUS_E;


/*create channel api,
**input paramters: appkey, Microsoft will provide it for every application. We need apply it.`
**appname, application name used by bind appkey and application.
** func_ptr, get channel api callback.
**return paramters: status, the status of WNS application manager*/
a.      MSG_WNS_STATUS_E
         create_push_notification_channel_for_applicationasync( char* appname,  char * appkey ,
        get_push_notification_channel_cb *func_ptr)


/* get channel api callback,
**input paramters: appkey, Microsoft will provide it for every application. We need apply it.`
**appname, application name used by bind appkey and application.
** channel, struct, include uri and expiry_time
**return paramters: void*/
b.    typedef void (*get_push_notification_channel_cb)( char* appname,  char *
appkey ,push_notification_channel * channel);
```

```
 /*close channel api,
**input paramters: appkey, Microsoft will provide it for every application. We need apply it.*/
c.     BOOL  close_push_notification_channel_for_applicationasync(char * appkey)

 /*get global settings api,  will return true or false. True is push notification service is on,
otherwise is off.
**it can get push notification service state at the current phone */
d.      BOOL get_push_notification_setting()

/*get notification message detail api,
**input paramters: appkey, index
**return paramters: struct, include message detail information*/
e.     wns_push_notification * get_notifcation_message_information (char * appkey, int index)

/*get notification message  counts
**input paramters: appkey
**return:  application notification message counts.*/
f.     int get_notification_message_count(char * appkey)
```