



# CHAPTER 10

## Notification Handling Functions

---

This chapter provides information about the following notification handling functions:

- [deregisterNotificationListener](#), page 10-1
- [disableNotificationByEmail](#), page 10-2
- [enableNotificationByEmail](#), page 10-2
- [registerNotificationListener](#), page 10-3

### deregisterNotificationListener

#### Synopsis

```
boolean deregisterNotificationListener(UserToken token, NotificationListener listener)  
throws RemoteException;
```

#### Description

This function removes the notification listener from the client program.

#### Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
listener	Notification Listener interface, mandatory	—	An object that was registered previously by <code>addNotificationListener()</code> .

#### Return

This function returns true if the de-registration is successful and false if it is unsuccessful.

#### Error and Exception

If a system error prevents the operation from completing, a `RemoteException` is thrown.

When an error occurs, this function returns false.

## disableNotificationByEmail

### Synopsis

```
boolean disableNotificationByEmail(UserToken token) throws RemoteException;
```

### Description

This function disables sending e-mail notifications to the user.

### Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.

### Return

This function returns true if it is successful and false if it is unsuccessful.

### Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, this function returns false.

## enableNotificationByEmail

### Synopsis

```
boolean enableNotificationByEmail(UserToken token) throws RemoteException;
```

### Description

This function enables sending e-mail notifications to the user.

### Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.

### Return

This function returns true if it is successful and false if it is unsuccessful.

**Error and Exception**

If a system error prevents the operation from completing, a `RemoteException` is thrown.

When an error occurs, this function returns false.

## registerNotificationListener

**Synopsis**

```
boolean registerNotificationListener(UserToken token, NotificationListener listener)
throws RemoteException;
```

**Description**

This function allows a client program to register a listener for notifications. When a notification has occurred, the `onNotification()` method in the listener object is invoked.

The client must use the `registerNotification()` application programming interface (API) in order to be able to receive notification. The client should implement the `NotificationListener` interface in order to be able to take action upon notification.

```
package com.cisco.nm.clm.common;
public interface NotificationListener {
    public void onNotification(Notification notification);
}
```

Clients should implement the `NotificationListener` interface as follows.

```
public class MyClient implements NotificationListener {
    //other fields and functions to do client's job

    //the action that will be taken on notification.
    public void onNotification(Notification notify) {
        System.out.println(notify.toString());
    }
}
```

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
listener	Notification Listener interface, mandatory	—	An object that implements the <code>NotificationListener</code> interface.

**Return**

This function returns true if the registration is successful and false if it is unsuccessful.

**Error and Exception**

If a system error prevents the operation from completing, a `RemoteException` is thrown.

When an error occurs, this function returns false.