REL-IDverify-API Documentation

Introduction

REL-IDverify is a feature provided by REL-ID. Using this feature, enterprises can initiate a push notification based approval request to any of their transactional (login / financial transaction or any other approval) operations to the user. REL-ID server ensures to send the notification to user's registered device. Enterprises just need to invoke REST Web service call to REL-ID backend to initiate the notification request, and poll for its status.

REL-IDverify is available for both Android and iOS platforms.

Most importantly, REL-ID uses Google / Apple notification services just as a doorbell (to indicate, there is some notification). This means, the actual transactional information (what resource is accessed, amount for transaction, etc.) is never shared over the public notification services OR with the ISPs. Once the REL-ID app receives a notification, it pulls the actual transactional information though REL-ID's MITM proof secure channel, and shows the prompt to the user to act on the transaction.

The REL-IDverify APIs that enterprise can invoke are authenticated REST web service APIs. REL-ID admin will create a unique "Enterprise ID" along with the authentication credentials and provide these authentication credentials to enterprise developers. Enterprise developers would use these credentials to invoke the REL-IDverify web services. Enterprises can choose to create multiple verify enterprise IDs, so that notifications from different applications / departments etc. can be tracked distinctly.

REL-IDverify exposed these authenticated web service APIs using a web service component, which accepts the http web requests on a configurable port (default 8007). In this document, we have used "REL-ID-VERIFY-SERVER" to indicate host name / IP address of REL-ID server which has the REL-IDverify component, and "VERIFY-PORT" as the port on which REL-IDverify accepts these requests.

Generate REL-ID Notification Request

Enterprise can use this API to generate the notification (approval) request for their transactional operation. Enterprise must provide unique "message ID" for individual notification request. REL-ID system shall generate a notification, and send it to user's device (apple / android - depending on type of device user has registered with REL-ID).

In case user is not available in REL-ID OR is not in the valid state, this API will return error.

In case of success, it will generate a unique notification-id and will send the notification-id as response. This notification-ID can be used to check status of notification using the next API.

REL-ID provides an option to the enterprise, to define its own messages that needs to be displayed to the user. There are two types of messages. "msg" attribute contains the actual transactional information and "notification_msg" contains the text that flows through public Google / Apple notification servers. We recommend not to send any transactional information using "notification_msg" attribute.

Enterprise can also choose the number of options and text of the actions that needs to be shown to the user. For example, "Approve/Reject" OR "Accept/Decline/Fraud" etc.

Against every action (Accept/Decline), enterprise can also choose the authentication level (none/password/biometric etc) by providing auth_level number. This feature helps to enforce additional authentication on the action.

REQUEST PARAMETERS

- **Request URL**:

  http://REL-ID-VERIFY-SERVER:VERIFY-PORT/authorize.htm

- **Request Method:**

  POST

- **Request Headers:**

  Add the Authentication Header with credentials

Authorization: "Basic Base64.encodeBase64("ENTERPRISE-USER:ENTERPRISE-PASSWORD")"|

- **HTTP Request Body (JSON)**

```json
{
    msg_id:"<message-id>",
    enterprise_id:"<enterprise-id>",
    user_id:"<user-id>",
    msg:{
        subject:"<message-subject>",
        body:"<message-body>"
    },
    notification_msg:{
        subject:"<notification-message-subject>",
        body:"<notification-message-body>"
    },
    expiry_time:<expiry-time-in-seconds>,
    actions:[{
        label:"<label>",
        action:"<action>",
        authlevel:"<0/1/2>"
      },{
        label:"<label>",
        action:"<action>",
        authlevel:"<0/1/2>"
    }]
}
```

test	asdfasdfasdf

Details of the JSON parameters in Request Body

Field	Description
msg_id	Unique message id generated by enterprise. Enterprise must provide unique msg_id for every notification. msg_id value can be as simple as the transaction_id generated for the transaction for which enterprise is seeking

	REL-ID approval. REL-ID system can provide logs OR notification status against the msg_id for forensic purposes or during audits.
enterprise_id	Enterprise ID provided by REL-ID admin
user_id	user_id of user in REL-ID system
msg	Actual Transactional Message to be displayed to the user. Contents in the Body will be shown as details to the user. Contents in subject will be shown as subject line for the notification.
notification_msg	Text that enterprise would like to display in Push notification (For example "you have a login notification"). Do not send transactional information here.
expiry_time	Number of seconds for which REL-ID shall consider lifetime for this notification request. If user does not act on the notification within the given number of seconds, notification will be marked as EXPIRED by REL-ID
actions	Array of JSON objects with label & action text along with authentication level. Each array element is presented as a clickable button. Label value will be used as TEXT to show to the user and Action Text will be send back to enterprise as notification status. On clicking any of the displayed buttons, authlevel option works as follows: • if authlevel is 0 - user is not prompted for any additional authentication • if authlevel is 1 - user is prompted for Password - on successful password authentication notification status is posted to server • if authlevel is 2 - user is prompted for TouchID/Pattern on iOS/Android - on successful TouchID/Pattern verification notification status is posted to server

HTTP Response Body

TYPE	Description
Success Response	`{ response_code:"0", notification_uuid:"" }` The notification_uuid received in response has to be used to fetch notification status (using next api)
Failure Response	`{ response_code:"1", error_code:"" error_message:"" }`

Error Types

CODE	Message	Description
2600	INVALID URI	If URI is not correct. (Possibly a spelling mistake in "authorize.htm")
2601	Authorization header not found	Empty Authorization Header
2602	Authorization method not found	Empty Authorization Method (If you miss to put "Basic")
2603	Authorization payload not found	Incorrect Authorization Payload (If you miss to put Base64 converted value, and send a blank after "Basic")
2604	Authorization Failed	Incorrect Authorization Header
3521	Notification Save Request null or empty	If Blank request body is provided
3522	Invalid Notification Save Request	Structure of http request body is not according to specifications
3525	Null or Empty parameter -> MSG_ID	If Blank msg_id is provided

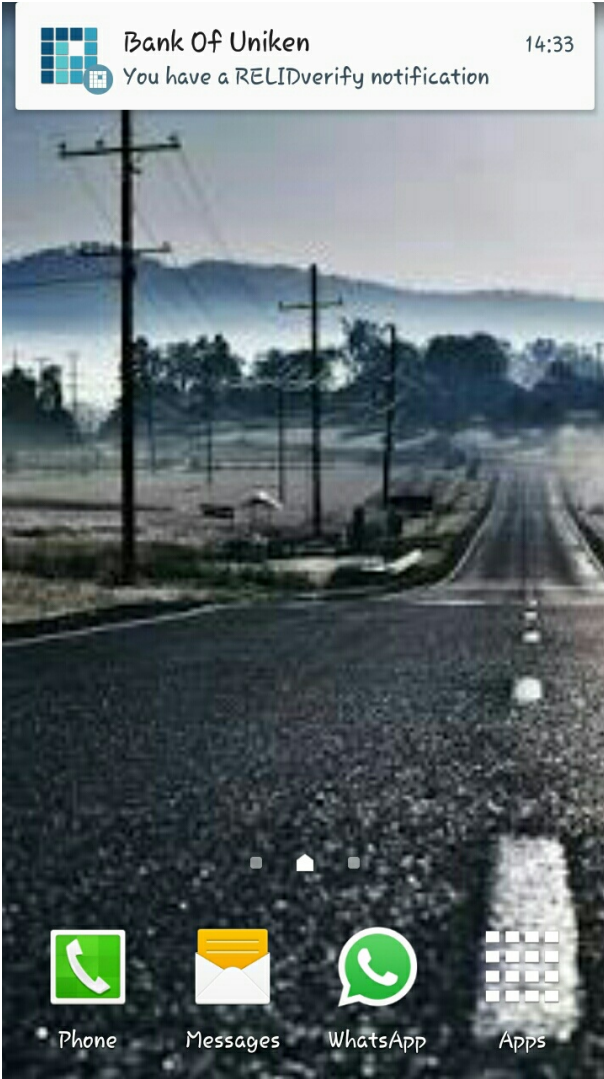| 3526	Null or Empty parameter -> ENTERPRISE_ID	If Blank enterprise_id is provided
3527	Invalid ENTERPRISE ID	If incorrect enterprise_id is provided
3528	Null or Empty parameter -> USER_ID	If Blank user_id is provided
3529	Invalid User State	If user is present, but in a state (BLOCKED/SUSPENDED) which is not legitimate state to send notifications
3530	User not active or present	User is not active in REL-ID OR not present
3531	Null or Empty parameter -> MSG	If msg value is null or blank
3532	Null or Empty parameter -> MSG_SUBJECT	If msg_subject value is null or blank
3533	Null or Empty parameter -> MSG_BODY	If msg_body value is null or blank
3534	Null or Empty parameter -> NOTIFICATION_MSG	If notification_msg value is null or blank
3535	Null or Empty parameter -> NOTIFICATIONMSG_SUBJECT	If notification_msg_subject value is null or blank
3536	Null or Empty parameter -> NOTIFICATIONMSG_BODY	If notification_msg_body value is null or blank
3537	Null or Empty parameter -> EXPIRY_TIME	If expiry_time value is null or blank
3538	Invalid Expiry Time	If incorrect value for expiry_time is provided

3541	Null or Empty parameter -> ACTIONS	If actions value is null or blank
3542	Invalid Actions Provided	If actions value is not as per the structure
3543	Null or Empty parameter -> ACTIONS_LABEL	If actions_label value is null or blank
3544	Null or Empty parameter -> ACTIONS_ACTION	If actions_action value is null or blank
3545	Internal Server Error. Please retry.	In case of unknown server error

Example

TYPE	Description
Example URL	http://relidverify.uniken.com:8007/authorize.htm
Authorization Header	- Assume you have been provided a username as "reliduser" with its password as "password123" by REL-ID admin. - First, calculate Base64 of "reliduser:password123". Refer https://www.base64encode.org/, which shows base64 encode "reliduser:password123" is "cmVsaWR1c2VyOnBhc3N3b3JkMTIz" - Hence your authorization header will be as shown below **Authorization: "Basic cmVsaWR1c2VyOnBhc3N3b3JkMTI** **Example: Setting HTTP Authorization Header in Java:** - String encoding = Base64Encoder.encode ("reliduser:password123"); - HttpPost httppost = new HttpPost("http://relidverify.uniken.com:8007/authorize.htm - httppost.setHeader("Authorization", "Basic " + encoding); - HttpResponse response = httpclient.execute(httppost);

| HTTP Request Body | ```
{
msg_id:"12345678877",
enterprise_id:"CBS",
user_id:"testuser",
notification_msg:{
    body:"You have a REL-IDverify notification",
    subject:"REL-IDverify notification"
},
msg:{
    body:"Windows NT,10.0;WOW64\n,
(49.248.126.42)\nSite:Netbanking Retail",
    subject:"Login Attempt"
},
expiry_time:180,
actions:[{
    label:"Accept",
    action:"Accept"
  },{
    label:"Reject",
    action:"Reject"
  }]
}
``` |
| HTTP Response Body | ```
{
response_code:0,
notification_uuid:"03f67f9e-ec4f-11e6-b006-92361f002671"
}
``` |

Example Notification on App

Bank Of Uniken   14:33
You have a RELIDverify notification

Phone   Messages   WhatsApp   Apps

Back   My Notifications

Login Attempt   02/03/17
- Windows NT 10.0; WOW64
- , (49.248.126.42)
- Site: NetBanking Retail

Approve   Reject

Check Notification Status

Enterprise will get a unique notifcaiton-uuid upon successful invocation of the first API to generate the notification. Using this notifcaiton-uuid enterprise can keep checking the status of the notification as per desired time intervals.

This request-and-poll architecture ensures that REL-ID servers and enterprise server both does not need to keep the TCP connections open for high number of concurrent notifications.

NAME	Description
Request URL	http://REL-ID-VERIFY-SERVER:VERIFY-PORT/notificationStatus.htm
Request Method	GET
Request Headers	Add the Authentication Header with credentials **Authorization:**"Basic Base64.encodeBase64("ENTERPRISE-USER:ENTERPRISE-PASSWORD")"
Parameters	None

HTTP Response Body

TYPE	Description

Success response	`{`
	`notification_uuid:"<notification_uuid>"`
	`response_code:"0",`
	`status:<notification_status>,`
	`delivery_status:<delivery_status>,`
	`action_response:"<RESPONSE_TEXT>"`
	`}`

| Success response | |

```
{notification_uuid:"<notification_uuid>", response_code:"0", status:<no
tification_status>, delivery_status:<delivery_status>,a ction_response:
"<RESPONSE_TEXT>"}
```

| status |

- **ACTIVE** : Notification is Active (User has not acted on it yet)
- **UPDATED** : User has taken action on the notification on it yet)
- **EXPIRED** : Notification is expired

| delivery_status |
- **NONE** : REL-ID is attempting to deliver notification on Users Device(s)
- **PARTIALLY_NOTIFIED** : User has recived notification on at-least in device (in case he has multiple devices)
- **NOTIFIED** : User has recived notification on ALL the registered devices

| | action_response |
- **NONE** : User has not taken action on notification yet
- **RESPONSE_TEXT** : Please refer the examples below
  |

Action Response Example

For example, if the actions value in the notification request was

```
actions:[{label:"Accept",action:"Accept"}, {label:"Reject",action:"
Reject"}]
```

If the User has clicked / tapped on "Accept" label, then RESPONSE_TEXT will be "Accept"

If the User has clicked / tapped on "Reject" label, then RESPONSE_TEXT will be

"Reject"

....

Another example, if the actions value in the notification request was **actions: [{label:"Accept",action:"YES"}, {label:"Reject",action:"NO"}]**

If the User has clicked / tapped on "Accept" label, then RESPONSE_TEXT will be "YES"

If the User has clicked / tapped on "Reject" label, then RESPONSE_TEXT will be "NO"

....

| | Failure response |

{

response_code:"1",

error_code:""

error_message:""

}

| | Error Codes |

| 3556 | Notification Identifier is null or empty | If notification_uuid value is null or blank | | 3557 | Internal Server Error. Please retry. | unknown error | | 3558 | Notification Identifier is not found | If no record found for given notifcaiton-uuid |

Error Code	Error Message	scenario

|

Example | Example URL |

http://relidverify.uniken.com:8007/notificationStatus.htm/03f67f9e-ec4f-11e6-b006-92361f002671 | | HTTP Response Body | { response_code:0, action_response:"Accept", delivery_status: "NOTIFIED", notification_uuid:"03f67f9e-ec4f-11e6-b006-92361f002671" } |

Example Section