ORACLE

Oracle Responsys

28-Jul-2022

Automatic Failover for Transactional Messaging (AFTM) Developer's Guide

REST API v1.3

Update 20A

Limited-release documentation for developers who use AFTM API calls to provide high-availability for API-triggered transactional email, SMS, and Mobile Push messages during service disruptions.

Contents

About AFTM functionality and the Responsys API **5**

Standard API References 5

General Usage Guidelines 6

What's new to the AFTM REST API **8**

Changes for 22B 8

Changes for 22A 8

Changes for 21D 8

Changes for 21C 9

Changes for 21B 9

SOAP-based code changes required by AFTM **10**

Use your account-specific URL end point 10

Use only a limited subset of SOAP API functions 11

Use specific functions for merging members and triggering email or SMS messages 12

Handling errors and exceptions 20

REST-based AFTM WS API **21**

Use your account-specific URL end point 21

Use only a limited subset of REST API functions supported for AFTM 22

Use the AFTM-specific Authentication Token for AFTM requests 23

AFTM-specific tasks 24

Retrieve profile list members using query attribute (HA Retrieve List Members) 24

Merge members into a profile list & trigger email messages to them (HA Merge Trigger Email) 27

Merge members into a profile list and trigger email messages with attachments (HA Merge Trigger Email with Attachments) 30

Merge members into a profile list & trigger SMS messages to them (HA Merge Trigger SMS) 38

Trigger mobile push messages (HA Trigger Push Message) 40

Merge members into a profile list (HA Merge List Members) 44

Handling errors and exceptions 47

Asynchronous AFTM WS API 49

Merge members into a profile list (HA Async Merge List Members) 49

Merge members into a profile list & trigger email messages to them (HA Async Merge Trigger Email) 52

Merge members into a profile list & trigger SMS messages to them (HA Async Merge Trigger SMS) 56

Trigger Push Message API (HA Async) 59

Polling API endpoint 64

Common error responses for the Asynchronous API 67

Request payload is not well-formed JSON (INVALID_REQUEST_CONTENT) 67

Recipient limit is exceeded (RECIPIENT_LIMIT_EXCEEDED) 67

Queue is full (UNABLE_TO_SUBMIT_ASYNC_REQUEST) 68

Common error responses for the Polling API 69

Response purged (RESPONSE_PURGED) 69

Async API not enabled (ASYNC_API_DISABLED_FOR_USER) 69

Processing request (PROCESSING_REQUEST) 70

Campaign is invalid (CAMPAIGN_IS_INVALID) 70

Merge fails for one recipient 71

Request ID is not valid (INVALID_REQUEST_ID) 72

About AFTM functionality and the Responsys API

Automatic Failover for Transactional Messaging (AFTM) only provides high-availability for API-triggered transactional email, Push, and SMS messages during service disruptions. As a result, only a limited set of API functions are available at an account-specific end-point URL for AFTM API calls.

AFTM is supported for both SOAP and REST protocols, as described in this document:

- SOAP AFTM Protocols are described in the topic "SOAP-based code changes required by AFTM".
- REST AFTM Protocols are described in the topic "REST-based AFTM WS API".

Standard API References

The standard Oracle Responsys API guides provide some information that is also applicable to the AFTM API, such as the standard API error responses and how to handle them. We provide direct links where applicable in this document, but for your reference, you can find the standard guides in the following locations:

- *Oracle Responsys SOAP API Developer's Guide* – The standard Responsys Web Services SOAP API Guide.
- *Oracle Responsys REST API Developer's Guide* – A PDF version of our standard Responsys Web Services REST API Guide.
- *REST API for Oracle Responsys Marketing Cloud Service* – An HTML version of our standard Responsys Web Services REST API Guide. This guide contains the full request and response parameter descriptions for the API endpoints.

General Usage Guidelines

The following list provides important general guidelines for AFTM software developers. Specific programming requirements are covered in the following sections.

API User management

- It's recommended that only one user ID per account be used for AFTM API login.
- The primary administrator of the account must create the user ID. This is the first Account Admin created in Responsys when it is configured, and this user ID is usually noted in the account user section as admin@accountname. User names are not synced from primary to secondary unless the user ID is created by the primary account administrator. If another account administrator creates the user ID, the authentication request on the failover account will fail.
- Because the Interact account user IDs must be distinct within each POD (that is, interact2 and interact5), you may be asked to create a new user ID if your primary account's existing user ID has already been assigned in the failover POD.
- Responsys user IDs assigned as account administrator should not be used for making AFTM API requests.
- See the article "Creating API users in Responsys" for more information about creating a user ID with the correct roles for an API user.

Code guidelines

- You must never cache the IP address to which you submit your WS API requests.
- Ensure that your client systems use a Transport Layer Security (TLS) version 1.2.
- You must always recompile your software if you have made any changes or reference new URL end points.

- Same throttling policies apply to the AFTM WS API requests. Therefore, you are responsible for queuing the AFTM API requests that were not successfully processed and resubmitting them again.
- You must ensure that there are no programming errors and that the software is fully tested, such that correct data is always passed to the API functions. This will help avoid any potential problems with automatically switching to the failover account.
- You should pass as much personalization data as possible through the WS API requests.
- When the system switches between the primary and failover accounts, you may receive “INVALID_SESSION_ID” (SOAP) or “INVALID_TOKEN” (REST) errors. Ensure that your code contains logic for making multiple re-authentication attempts when an API session becomes invalid (SOAP) or the token becomes invalid (REST).

What's new to the AFTM REST API

This section describes changes and enhancements to the Oracle Responsys AFTM REST APIs in product updates during the past year.

- Update 22B
- Update 22A
- Update 21D
- Update 21C
- Update 21B
- Update 21A

Changes for 22B

- Added support for the HA Async Trigger Push Message API, as described in Trigger Push Message API (HA Async).

Changes for 22A

There are no significant updates to the AFTM API.

Changes for 21D

There are no significant updates to the AFTM API.

Changes for 21C

There are no significant updates to the AFTM API.

Changes for 21B

There are no significant updates to the AFTM API.

SOAP-based code changes required by AFTM

There are four areas that require changes in your SOAP-based code to use AFTM.

These are:

1. Use your account-specific URL end point for AFTM-supported calls.
2. Use only a limited subset of SOAP API functions in your AFTM sessions. (For more information about the object types and error messages, refer to the standard Responsys SOAP API Guide.)
3. Use specific functions for merging members and triggering email or SMS messages.
4. Handling errors and exceptions.

Use your account-specific URL end point

Once the secondary failover account is provisioned, you will receive an account-specific URL endpoint from Oracle that must be used in your AFTM software code.

If your Responsys account is globally routed, your URL endpoint will be a unique global routing endpoint. See Global Routing for more details.

If your account is on a legacy pod, the format of this URL will be similar to the following:

```
https://wsX-companyABC.responsys.net/webservices/services/ResponsysWSService
```

where wsX refers to the pod (for example, ws2 indicates the interact2 pod), and companyABC is the account-specific identifier, such as a company name.

If you have not received your account-specific URL endpoint, please contact your Customer Success Manager (CSM).

Use only a limited subset of SOAP API functions

Only the following standard API functions are processed through AFTM during service disruptions:

- login
- loginWithCertificate
- authenticateServer
- logout
- retrieveListMembers - Use this function only when your code also uses an AFTM-specific call (that is, HAMergeTriggerEmail or HAMergeTriggerSMS). The retrieveListMembers call will work on both primary and secondary accounts, but it cannot detect problems and fail over, as the AFTM-specific calls do.

When using the SOAP session management calls, you must use the account-specific URL endpoint. **Do not use other standard API functions in the same sessions as the AFTM endpoints.** If the same API client code makes AFTM and non-AFTM API calls to the AFTM endpoint, be aware that:

- The entire transaction in the same sessionID may be invalidated during a failover. There is no option of recovery.
- Any non-AFTM API calls that lands on the secondary node causes the system to throw the exception "API_NOT_ALLOWED_IN_SECONDARY" in the HTTP Response. If you mix AFTM and standard API calls in one sessionID, the system defaults to using the lower standard API throttling limits. This may degrade overall performance, because AFTM calls use higher tiered limits.

The following calls are specific to AFTM. Use them as described in the next section:

- HaMergeTriggerEmail
- HaMergeTriggerSMS

Use specific functions for merging members and triggering email or SMS messages

Two specific functions called **HaMergeTriggerEmail** and **HaMergeTriggerSMS** are available specifically in AFTM. They combine into one transaction the merging of members into a profile list and the triggering of messages to those members. Responsys campaigns that already exist can be sent up to 200 members of a profile list. When using AFTM and SOAP, these functions must be used to ensure integrity of data in the primary and secondary failover accounts, as well as automatically switching from the primary to the secondary account during a service disruption. These functions and their input and output parameters are described in the sections that follow.

HaMergeTriggerEmail

This function can be used for triggering both transactional and promotional emails. However, promotional emails will fail on the secondary account during failover. Both merging members and the email trigger will fail.

Syntax

```
TriggerResult[] = service.HaMergeTriggerEmail (RecordData recordData, ListMergeRule
mergeRule,  InteractObject campaign, TriggerData[] triggerData)
```

Request Arguments

Name	Type	Description
recordData	RecordData	Array of RecordData objects that contain field and record data
mergeRule	ListMergeRule	Defines the merge rules for how to handle the record data
campaign	InteractObject	Campaign name and folder
triggerData	TriggerData[]	An array of TriggerData objects that consists of an OptionalData object array The

	TriggerData object has the list of name/value pair parameters needed for triggering messages. OptionalData [] has optional name/value pair parameters associated with List member.

Response

Call returns an array of **TriggerResult** objects. This object has the following properties:

Name	Type	Description
recipientId	Long	Interact internal recipient ID (RIID_) for the individual to whom the message was sent
success	Boolean	Success flag for trigger message request
errorMessage	String	NO_RECIPIENT_FOUND MULTIPLE_RECIPIENTS_FOUND

HaMergeTriggerSMS

Syntax

```
TriggerResult[] = service.HaMergeTriggerSms (RecordData recordData, ListMergeRule mergeRule, InteractObject campaign, TriggerData[] triggerData)
```

Request Arguments

Name	Type	Description
RecordData	RecordData	Array of RecordData objects that contain field and record data
mergeRule	ListMergeRule	Defines the merge rules for how to handle the record data
campaign	InteractObject	Campaign name and folder
triggerData	TriggerData[]	An array of TriggerData objects that consists of an OptionalData object array The TriggerData object has the list of name/value pair parameters needed for triggering messages. OptionalData

	[] has optional name/value pair parameters associated with the List member.

Response

Call returns an array of **TriggerResult** objects. This object has the following properties:

Name	Type	Description
recipientId	Long	Interact internal recipient ID (RIID_) for the individual to whom the message was sent
success	Boolean	Success flag for trigger message request
errorMessage	String	NO_RECIPIENT_FOUND MULTIPLE_RECIPIENTS_FOUND

Sample code for SOAP

Creating TriggerData

```
TriggerData[] triggerData = null;
if (numOfRecipientsInCall > 0) {
    triggerData = new TriggerData[numOfRecipientsInCall];
 }
 for (int i = 0; i < numOfRecipientsInCall; i++) {
      triggerData[i] = new TriggerData();
      //How many campaign variables you want to specify
      OptionalData[] variables = null;
      if (numOfVariables > 0) {
            variables = new OptionalData[numOfVariables];
       }
```

```
      for (int j = 0; j < numOfVariables; j++) {
            variables[j] = new OptionalData();
            variables[j].setName(varName);
            variables[j].setValue(carValue);
         }
         triggerData[i].setOptionalData(variables);
      }
}
```

Making an HaMergeTriggerEmail call

```
try {
   HaMergeTriggerEmail haMergeTriggerEmail = new HaMergeTriggerEmail();
   InteractObject campaignObj = new InteractObject();
   campaignObj.setFolderName(folderNameForTrigger);
   campaignObj.setObjectName(campaignName);
   haMergeTriggerEmail.setRecordData(recordData);
   haMergeTriggerEmail.setMergeRule(listMergeRule);
   haMergeTriggerEmail.setCampaign(campaignObj);
   haMergeTriggerEmail.setTriggerData(triggerData);
   HaMergeTriggerEmailResponse resp = stub.HaMergeTriggerEmail
(haMergeTriggerEmail,
               sessionHeader);
   if (resp == null){
          sop("Failed to create result ");
    }
    else {
          com.rsys.ws.TriggerResult[] trRes =resp.getResult();
          if (trRes == null || trRes.length == 0) {
             sop("Failed to create result ");
          } else {
             int i = 0;
             for (com.rsys.ws.TriggerResult res: trRes){
                if (res == null){
                sop("Record["+i+"] - Failed to create Result");
                }
                else if (res.getSuccess()){
                sop("Record["+i+"] - Success");
               }
```

```
                else {
                sop("Record["+i+"] - Failed:"+ res.getErrorMessage());
               }
             }
       }
   }
}
catch (TriggeredMessageFault trgMsgFaultEx) {
        sop("TriggeredMessageFault HaMergeTriggerEmail");
        sop("Exception Code = " + trgMsgFaultEx.getFaultMessage().getExceptionCode());
        sop("Exception Msg  = " + trgMsgFaultEx.getFaultMessage().getExceptionMessage
());
}
catch (UnexpectedErrorFault unexpectedEx) {
        sop("unexpectedEx HaMergeTriggerEmail");
        sop("Exception Code = " + unexpectedEx.getFaultMessage().getExceptionCode());
        sop("Exception Msg  = " + unexpectedEx.getFaultMessage().getExceptionMessage());
      }
      catch (UnrecoverableErrorFault unexpectedEx) {
        sop("UnrecoverableError HaMergeTriggerEmail");
        sop("Exception Code = " + unexpectedEx.getFaultMessage().getExceptionCode());
        sop("Exception Msg  = " + unexpectedEx.getFaultMessage().getExceptionMessage());
      }
      catch (RemoteException remoteEx) {
        sop("remoteEx HaMergeTriggerEmail");
        sop("Exception Msg = " + remoteEx.getMessage());
      }
```

Making an HaMergeTriggerSms call

```
try {
   HaMergeTriggerSms haMergeTriggerSms = new HaMergeTriggerSms();
   InteractObject campaignObj = new InteractObject();
   campaignObj.setFolderName(folderNameForTrigger);
   campaignObj.setObjectName(campaignName);
   haMergeTriggerSms.setRecordData(recordData);
   haMergeTriggerSms.setMergeRule(listMergeRule);
   haMergeTriggerSms.setCampaign(campaignObj);
   haMergeTriggerSms.setTriggerData(triggerData);
```

```
   HaMergeTriggerSmsResponse resp = stub.HaMergeTriggerSms(haMergeTriggerSms,
sessionHeader);
   if (resp == null){
           sop("Failed to create result ");
    }
    else {
                com.rsys.ws.TriggerResult[] trRes =resp.getResult();
                if (trRes == null || trRes.length == 0) {
                   sop("Failed to create result ");
                } else {
                   int i = 0;
                   for (com.rsys.ws.TriggerResult res: trRes){
                      if (res == null){
                         sop("Record["+i+"] - Failed to create Result");
                      }
                      else if (res.getSuccess()){
                         sop("Record["+i+"] - Success");
              }
                      else {
                         sop("Record["+i+"] - Failed:"+ res.getErrorMessage());
              }
           }
       }
   }
}
catch (TriggeredMessageFault trgMsgFaultEx) {
         sop("TriggeredMessageFault HaMergeTriggerSMS");
         sop("Exception Code = " + trgMsgFaultEx.getFaultMessage().getExceptionCode());
         sop("Exception Msg  = " + trgMsgFaultEx.getFaultMessage().getExceptionMessage
());
}
catch (UnexpectedErrorFault unexpectedEx) {
         sop("unexpectedEx HaMergeTriggerSMS");
         sop("Exception Code = " + unexpectedEx.getFaultMessage().getExceptionCode());
         sop("Exception Msg  = " + unexpectedEx.getFaultMessage().getExceptionMessage());
}
catch (UnrecoverableErrorFault unexpectedEx) {
         sop("UnrecoverableError HaMergeTriggerSMS");
         sop("Exception Code = " + unexpectedEx.getFaultMessage().getExceptionCode());
         sop("Exception Msg  = " + unexpectedEx.getFaultMessage().getExceptionMessage());
```

```
    }


catch (RemoteException remoteEx) {
        sop("remoteEx HaMergeTriggerSMS");
        sop("Exception Msg = " + remoteEx.getMessage());
}
```

Handling errors and exceptions

Once the API requests are routed from one account to the other, a new session must be established. Therefore, your code must monitor responses for an exception for an invalid session, and then re-login with the same credentials used to access your primary Interact account.

- If you receive an unrecoverable exception, your code should wait about 30 seconds and resubmit the request only once.
- If there is no system response, your code can retry up to three times, with two minute intervals. If you still do not receive a response, close the session and re-log in.

REST-based AFTM WS API

There are four areas to consider when developing your REST-based code in order to use AFTM:

1. Use your account-specific URL end point for AFTM-supported calls.
2. Use only a limited subset of REST API functions supported for AFTM.
3. Use the AFTM-specific Authentication Token for AFTM requests.
4. Handling errors and exceptions.

Use your account-specific URL end point

Once the secondary failover account is provisioned, you will receive an account-specific URL endpoint from Oracle that must be used in your AFTM software code.

If your Responsys account is set up for global routing, your URL endpoint will be a unique global routing endpoint. See Global Routing for more details.

If your account is on a legacy pod, the format of this URL will be similar to the following:

`https://wsX-companyABC.responsys.net/webservices/services/ResponsysWSService`

where wsX refers to the pod (for example, ws2 indicates the interact2 pod), and companyABC is the account-specific identifier, such as a company name.

If you have not received your account-specific URL endpoint, please contact your Customer Success Manager (CSM).

Use only a limited subset of REST API functions supported for AFTM

The AFTM WS API supports the following AFTM-specific tasks:

- Retrieve profile list members using query attribute (HA Retrieve List Members)
- Merge members into a profile list & trigger email messages to them (HA Merge Trigger Email)
- Merge members into a profile list and trigger email messages with attachments (HA Merge Trigger Email with Attachments)
- Merge members into a profile list & trigger SMS messages to them (HA Merge Trigger SMS)
- Trigger mobile push messages (HA Trigger Push Message)
- Merge members into a profile list (HA Merge List Members)

For authentication and refreshing your authorization token, AFTM also supports the following standard REST API authentication tasks. For these, use the instructions in the REST API Guide (links provided below) but use your account-specific URL endpoint.

- Log in with username and password
- Log in with username and certificates
- Refreshing tokens

IMPORTANT: When using these endpoints, you must send the request to the account-specific URL endpoint. Using the account-specific endpoint will return an AFTM-specific authorization token, which will also be valid on the AFTM servers, should failover occur. The format of the authentication request URL will be similar to the following (use api, not haApi, in the Service URL for auth requests):

```
POST https://wsX-companyABC.responsys.net/rest/api/v1.3/auth/token
```

Use the AFTM-specific Authentication Token for AFTM requests

After your client application has authenticated successfully, it will receive the authentication response body shown below.

- When logging in with username and password, a successful response contains the values you need.
- Logging in with username and certificates involves two REST API requests, and the response containing the values you need is the second one received from the Responsys system (step 4 in the “Log in with username and certificates” procedure).

The response body for authentication contains the authToken and subsequent endPoint values:

```
{
  "authToken": "<auth_token>",
  "issuedAt": <timestamp_when_token_was_issued>,
  "endPoint": "<endpoint_URI_value>"
}
```

Your client application must use the authToken values and subsequent endPoint values in all requests:

- Use the <endpoint_URI_value> when constructing the URI for your endpoint. The format may vary, depending on the programming language used to code the request. For example, if constructing a URL for an HA Merge Trigger Email request, it looks like this (endpoint URI + Service URL):

```
{endpoint_URI_value}/rest/haApi/v1.3/campaigns/{campaignName}/email
```

- In the Request Header, substitute the response value for <auth_token> in places where we show Authorization: <auth_token>.

AFTM-specific tasks

This section describes the six specific tasks that are available specifically in AFTM.

These tasks perform similarly to their standard API counterparts. However, when you use the haApi endpoints, you ensure data integrity in the primary and secondary failover accounts. You also ensure that your client application automatically switches from the primary to the secondary account during a service disruption.

Retrieve profile list members using query attribute (HA Retrieve List Members)

Merge members into a profile list & trigger email messages to them (HA Merge Trigger Email)

Merge members into a profile list and trigger email messages with attachments (HA Merge Trigger Email with Attachments)

Merge members into a profile list & trigger SMS messages to them (HA Merge Trigger SMS)

Trigger mobile push messages (HA Trigger Push Message)

Merge members into a profile list (HA Merge List Members)

Retrieve profile list members using query attribute (HA Retrieve List Members)

Use the interface to retrieve members of a Profile List, by using query attributes.

Service URL:

/rest/haApi/v1.3/lists/{listName}

> **NOTE:** For this request, ensure that your path uses haApi instead of api. The haApi part of the path is case sensitive.

Required Path Parameters:

`listName` - Name of the profile list from which you want to retrieve members.

Optional Path Parameters:

`qa` - Query attribute. Can be either 'r' (RIID_), 'e' (EMAIL_ADDRESS_), 'c' (CUSTOMER_ID_), or 'm' (MOBILE_NUMBER_).

`fs` - Comma-separated list of fields to retrieve, or use all to retrieve all fields.

`id` - ID corresponding to the query attribute.

Request Method:

GET

Request URL Example:

The following request URL example requests the values of the RIID_ and EMAIL_ADDRESS_ fields for a member of the ExampleList profile list whose EMAIL_ADDRESS_ is test@example.com.

```
/rest/haApi/v1.3/lists/ExampleList?qa=e&fs=EMAIL_ADDRESS_,RIID_
&id=test@example.com
```

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Response Body Properties:

`fieldNames` - Array of field names in the Profile List that were passed in the request using the fs parameter.

`records` - Array of field values returned in the query. Each element in the array represents a single Profile List member.

`fieldValues` - Array of Profile List field values (as specified using the fs parameter) for the recipient found using the query criteria.

`mapTemplateName` - The Map Template in Responsys that can be used to map Field Names of the Profile List to Column Names.

> **NOTE:** Unlike other HA REST APIs, the HA Retrieve List Members response is different from the standard Retrieve List Members request.

Sample Response:

In the following response body example, the query shown in the "Request URL Example" section (above) found one record in the ExampleList profile list for the email address test@example.com. Because the fs parameter requested the EMAIL_ ADDRESS_ and RIID_ for any list members found, those columns are listed in the fieldNames array, and the values for the record found are returned in the fieldValues array (contained in the records array).

```
{
   "fieldNames": [
      "EMAIL_ADDRESS_",
      "RIID_"
   ],
   "records": [
      {
         "fieldValues": [
            "test@example.com",
            "3001221"
         ]
      }
   ],
   "mapTemplateName": null
}
```

Merge members into a profile list & trigger email messages to them (HA Merge Trigger Email)

Use this interface to merge members into a profile list and subsequently send them an email message. This endpoint can be used for triggering both transactional and promotional emails. However, promotional emails will fail on the secondary account during failover. Both merging members and the email trigger will fail.

Service URL:

/rest/haApi/v1.3/campaigns/{campaignName}/email

> **NOTE:** For this request, ensure that your path uses haApi instead of api. The haApi part of the path is case sensitive.

Required Path Parameters:

`campaignName` - Name of the Email campaign to be sent to the recipients.

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body Properties:

Same as the standard Merge Trigger Email request.

NOTE: Using this API requires adhering to several constraints and guidelines, which are described in the standard Responsys REST API reference guide.

Sample Request Body:

```
{
"mergeTriggerRecordData": {
  "mergeTriggerRecords": [
    {
      "fieldValues": ["1", "foo.bar@oracle.com", "c1a_value"],
      "optionalData": [
        { "name": "CUSTOM2", "value": "c2a_value" }
      ]
    },
    {
      "fieldValues": ["2", "my.name@oracle.com ", "c1b_value"],
      "optionalData": [
        { "name": "CUSTOM2", "value": "c2b_value" }
      ]
    }
  ],
  "fieldNames": [
```

```
    "CUSTOMER_ID_",
    "EMAIL_ADDRESS_",
    "CUSTOM1"
  ]
},
"mergeRule": {
  "htmlValue": "H",
  "matchColumnName1": "EMAIL_ADDRESS_",
  "matchColumnName2": null,
  "optoutValue": "O",
  "optinValue": "I",
  "insertOnNoMatch": true,
  "defaultPermissionStatus": "OPTIN",
  "rejectRecordIfChannelEmpty": "E",
  "updateOnMatch": "REPLACE_ALL",
  "textValue": "T",
  "matchOperator": "NONE"
  }
}
```

Response Body Properties

Same as the standard Merge Trigger Email response.

Sample Response:

```
[
  {
    "errorMessage" : null,
    "success" : true,
    "recipientId" : 72067
  },
  {
    "errorMessage" : null,
    "success" : true,
    "recipientId" : 72087
  }
]
```

Merge members into a profile list and trigger email messages with attachments (HA Merge Trigger Email with Attachments)

Use this interface to merge members into a profile list and subsequently send them an email message with dynamic personalized attachments, which are passed through the API payload. This endpoint can be used for triggering both transactional and promotional emails. However, promotional emails will fail on the secondary account during failover. Both merging members and the email trigger will fail.

IMPORTANT: This is a controlled feature. To have this API enabled for your account, please contact your sales representative, account manager, or Customer Success Portal.

Service URL:

/rest/haApi/v1.3/campaigns/{campaignName}/emailAttachments

> **NOTE:** For this request, ensure that your path uses haApi instead of api. The haApi part of the path is case sensitive.

Required Path Parameters:

`campaignName` - Name of the Email campaign to be sent to the recipients.

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body Properties:

Same as the standard Merge Trigger Email with Attachments request.

NOTE: Using this API requires adhering to several constraints and guidelines, which are described in the standard Responsys REST API reference guide.

Sample Request Body:

```
{
    "mergeTriggerRecordDataWithAttachments": {
      "mergeTriggerRecordsWithAttachments": [
        {
          "fieldValues": [
            "1",
            "mdi1124@barcorp.com"
          ],
          "optionalData": [
            {
              "name": "CUSTOM1",
              "value": "custom value"
            },
            {
              "name": "CUSTOM2",
              "value": "custom data"
            }
          ],
          "attachmentData": [
            {
              "name": "Hello World.pdf",
              "value":
"JVBERi0xLjQKJYCAgIAKMSAwIG9iago8PC9QYWdlcyAyIDAgUiAvVmlld2VyUHJlZ
mVyZW5jZXMgOCAwIFIgL1R5cGUgL0NhdGFsb2cgPj4KZW5kb2JqCjIgMCBvYmoKPDw
vQ291bnQgMSAvTWVkaWFCb3ggWzAgMCA1OTYgODQyIF0gL1R5cGUgL1BhZ2VzIC9SZ
XNvdXJjZXMgMyAwIFIgL0tpZHMgWzUgMCBSIF0gPj4KZW5kb2JqCjMgMCBvYmoKPDw
vRm9udCA3IDAgUiA+PgplbmRvYmoKNCAwIG9iago8PC9GaWx0ZXIgL0ZsYXRlRGVjb
2RlIC9MZW5ndGggNjAgPj4Kc3RyZWFtCnic0zdUMDZSCEnjcgrhMlQwAEKwgLm5sZ6
```

FuZGBoUJILpeGR2pOTr5CeH5RToqipkJIFpdrCBcAWMsNBgplbmRzdHJlYW0KZW5kb
2JqCjUgMCBvYmoKPDwvUGFyZW50IDIgMCBSIC9UeXBlIC9QYWdlIC9Db250ZW50cyA
0IDAgUiA+PgplbmRvYmoKNiAwIG9iago8PC9CYXNlRm9udCAvQ291cmllci1Cb2xkI
C9TdWJ0eXBlIC9UeXBlMSAvRW5jb2RpbmcgL1dpbkFuc2lFbmNvZGluZyAvVHlwZSA
vRm9udCA+PgplbmRvYmoKNyAwIG9iago8PC8xIDYgMCBSID4+CmVuZG9iago4IDAgb
2JqCjw8L0Rpc3BsYXlEb2NUaXRsZSB0cnVlID4+CmVuZG9iago5IDAgb2JqCjw8L1N
1YmplY3QgKP7/AFMAYQBtAHAAbABlACAAYQBiAG8AdQB0ACAAYQAgAHMAaQBtAHAAb
ABlACAAJwBoAGUAbABsAG8AIAB3AG8AcgBsAGQAJwAgAHUAcwBpAG4AZwAgAFAARAB
GACAAQwBsAG8AdwBuKSAvQ3JlYXRvciAo/v8AbwByAGcALgBwAGQAZgBjAGwAbwB3A
G4ALgBzAGEAbQBwAGwAZQBzAC4AYwBsAGkALgBIAGUAbABsAG8AVwBvAHIAbABkAFM
AYQBtAHAAbABlKSAvQXV0aG9yICj+/wBTAHQAZQBmAGEAbgBvACAAQwBoAGkAegB6A
G8AbABpAG4AaSkgL0NyZWF0aW9uRGF0ZSAoRDoyMDExMDMwNTIwMDYxNSswMScwMCc
pIC9Qcm9kdWNlciAo/v8AUABEAEYAIABDAGwAbwB3AG4AIABmAG8AcgAgAEoAYQB2A
GEAIAAwAC4AMQAuADApIC9UaXRsZSAo/v8AUABEAEYAIABDAGwAbwB3AG4AIAAtACA
ASABlAGwAbABvACAAdwBvAHIAbABkACAAcwBhAG0AcABsAGUpID4+CmVuZG9iagp4c
mVmCjAgMTAKMDAwMDAwMDAwMCA2NTUzNSBmDQowMDAwMDAwMDE1IDAwMDAwIG4NCjA
wMDAwMDAwODggMDAwMDAgbg0KMDAwMDAwMDE4NyAwMDAwMCBuDQowMDAwMDAwMjE5I
DAwMDAwIG4NCjAwMDAwMDAzNDkgMDAwMDAgbg0KMDAwMDAwMDQxMSAwMDAwMCBuDQo
wMDAwMDAwNTEwIDAwMDAwIG4NCjAwMDAwMDA1MzkgMDAwMDAgbg0KMDAwMDAwMDU4M
SAwMDAwMCBuDQp0cmFpbGVyCjw8L1Jvb3QgMSAwIFIgL0luZm8gOSAwIFIgL1NpemU
gMTAgPj4Kc3RhcnR4cmVmCjEwMzcKJSVFT0Y="
},
{
"name": "weather.jpg",
"value":
"/9j/4AAQSkZJRgABAQEAYABgAAD/4QA6RXhpZgAATU0AKgAAAAgAA1EQAAEAAAABA
QAAAFERAAQAAAABAAAAAFESAAQAAAABAAAAAAAAAAD/2wBDAAIBAQIBAQICAgICAgI
CAwUDAwMDAwYEBAMFBwYHBwcGBwcICQsJCAgKCAcHCg0KCgsMDAwMBwkODw0MDgsMD
Az/2wBDAQICAgMDAwYDAwYMCAcIDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAz/wAARCACtANIDASIAAhEBAxEB/8QAHwAAAQUBA
QEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQ
RBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3O
Dk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5i
ZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8
vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgE
CBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRC
hYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ
3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0
tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDIkXA4I+lM34W
q6NIW+bP4U4wtIfl4r6A8cknnh8n5l+Y1msMH057ir9vo8k0mSP1qxPpOF2sq/UdqC
dWZCzYOM++M1YgvvJPtS3Nh5Jx69DVWaB4m60C1RpxavGV+b86e+pLnaprHJwfypaB
8xrfb17sBUkN/Co+8p9s1i0UBzG1dartT5JAPbNUj4gZThmH51Slfy0z1xVOO2kuD8
q9TihRDmZ01pd/a8tzRMSW5o0C1+w2P7w/Ofxq4bbzl3ClcopC0ZxkLSrbEnGK1bWE
N8vyj1q7Fo6kE7S3qfSlzAcnPpuX3dPrVd4GMm3+H2rrpNFWQH7tU73RVj6df50cwu
Uw7fSGmb2qx/YhTGPpV2GzaKT5e3Wp/KkmbkcLRcGWbeSHqtRyu3932rcktZNu1l6
/nTP7EaY9Pl9xRzDMrDz26c1Na6Jsbq26uitfD/ycL2rSg8OLNgMuMUz0u0y+zj

PeoZolkG3PNdpc6DDbWjMq5YdOOtc5c6aTI21Tkng0lINjE/s//AHaK1DYyZ+8aKq4
GXnaKFuljbletPERePvUTWRPILbvXFMC/aXYY8AVe8mO5Tp+IrGSBomrT0y6ZeGx+V
S0A86CrpkNu/mKzbnQmlkI6j2FdVavG6LuGB6igsu8jr6UrsDjZfD7Kv3WqGXR3Q9D
XcCISt91R68VHNaRlfmVafMLlOFbT5B0FQyQMo+tdgbANG7Kq8dsVVttDF0/zfKvWj
mFynJtbMz/MzYzkdauWVsYiNo4rpL/wtGCu0fN71JZeGti58n73enzBymXbHe22ti2
td8KkAe9OXRFDDcAh71rWGkxxKo4LHj61LZRkrp7QjIx9KuWbyH725WzjPrW9a6aoI
6MT6dK0otGhYfwt9AKjmK5Tk1RnfvnPNNnsfNAyjGuw/sOA4JU59sf4UjaFCw/H07U
cwcpw7acAcKv3u1T2thJG2Ntdh/YMQZj3PTjpUn9mqPxGOe3+eKOYOU5m30ppV9cc9
etX7LQMj7u76dx9a2V01FLe/T0FTkAKf4R37UcxRkw6ZGh6qGXPXk8e1S5WItj5ufv
Y3f57mn3BiCybW6nIA7/5/wA9axtWv5YWO1h83I5pbgQ+JNfWzjbHJPPBzmuO1Lxc0
x2suNvTb2rT1p2ul+asC60rzM/L+NaxSM5NkR8TEHr/AOPUVEfDLOd3ltzz1oqtCdT
Xht/LY/yqUKD2/Srkdvubqo/pTJrFg9Z3KIdilcj73enQW2H/ABpwUDnO1unFC3iRN
yd1AFpI2RdvZulAR1k+9j8Kal2twy9h04qYMsT7t27FAFiKNpOMbj3NTS2LSDkq39a
dYHcrEMCtWEKl9rNtoKSM90wSpqnLbyQS7lbjvWtf239zB68gVlyxzRg8sB0xQSy1B
eoUxI3HHBNWJNUjgh6HGPzrBnZk/l0pqzyE9/pQO5pf2n9om/xrQiljtkDK2T+prnd
5LZBx9ad9uc9TkelAjrrHUhMPmAUdj6VqWurwxJtbI/H+nauBg1JoDwzVYTxLnqefp
S5SuY7ptV3q21e3B3daqzak1sFbdIf1/wDrVyS+ImRdoZtvUjNMl1+R4yob9KXKHMd
hF4kVwP3kYyP4hio5vEuG+V48e3+fr+VcK93JyevfrTP7TZRt3H1p8ocx2d34iaePH
mYXvWe+v+Wfvtj3Ncy+qsW+9UM+ojZ96nyhzM6Y+I4y3zSD06niql1rqN91g34VzL6
nj3qFbmS4bjv6Cq5SeY2bzWVl68/hVeLW9r/LHu+opun6S1y3OcfyroNK8LQy/wAYz
70aAY/9sSH/AJZ0V2qeAI3RWDL8wz0oqeZFcrMFrLbyg+tRPas69DWkJoxGDyx9P/r
1XNwkbfMehoJMue1kB+63NUpYih/xFdPBPE3O3t9f51Jc6PBfR+ZGyKepU0XsOxysE
7Lx8yirkV0ykcbtvar1x4ZBJZfToO9U301oPu02Imh1J4WOON34VYk1Tz8H5QR3z0q
jFEDuLjmoGfa+VHHWkBs2d9IT69xk9adcEz99v0rF+3MI/lp1tqjSH5udvrQBfe1SY
7d3OO3ekbTwicfNTY7wJ3pBq6huduKAEa2wNrD36VDJZjHy1Yl1ONh71Wn1SJFJ3c0
ANEIT71Jsj3Vn3ut4b5fWq66uyty351XKwNhpI0TpUb3gReNvtWXLrygDvzioLnWcp
8vy+57UcormhPdST/1qBpCgwWX86x31Znk+U/8A16ltLWe7P3QfTFVawuYtTXCA8uT
7CoHuQ38J/OtS08E3F0itj8h0q5aeC1jf95J93jGKV0GpzbTt/wDqqxaXTId1dNa+G
LcllbmQdAB+tT2fguFX3MGIHqMfhS5g5WY9lrEh+7+lbOlX7My5jZsH0rSh0Kzgk3S
Ku0HgetbVjNaW6fu44dv1xmpkzTlII7+Qxr/o8h46+tFWj4gYH7zf98iipKPPE1dj1
P60Pq285/yaxt7RtQZ2Na8qMeY24NW8k7lY/wA6sR60R/F19a51blhUiXXNLlHc6iP
W933vmx056VH/AGhvb/e4zXPi5x/FQbtl5Bb8KOUZ0TSxiJvmG49qoXEyqTVAXz4+9
UckzSmjlAsG9YH5e9ENx5U25s7sVTO5R8tCxyEZz+FUA691lt/ysfwqrJqjE7mbAqO
eNhJ83H1qrdQEpkE8c4JoRF2WZ9fbop3VUbUZJHyzflVeirsIuPdCMc/f9KrtdOx64
+lNSF5W4VmqxBpjNy35ClogIFV5z3P1NTx6XM6/7PerC2nlnoFz7VoaXAQ3rnoD3o5
hooW2k7F3Z+pPatbT71LUhcikj0eS7lKnK89qnPh8wScj7vrUt9xm7H4jaKzCq2CV6
jvWVJ4gYyZbjd3pxRo4toXt1zVC8tJJl+VevWpVijWttUVVzn9adN4gZNwUlUb0rMj
0i4aEERsfSopLG4I+7s560WQFy58VeRGF3My+hqOXxkxTZHu57e9Z76Qzs26ShLFbY
cDc2OtPlQtS5/wkVyf7350VUy3/ADzWinYY+SDFVpYDn045NX3l6jH51XugXU/0oAz
5HKHFNWY/X0p0sbBzxUYXH+NBmSeefpTln/Cocc0bS38qANCFxJViOLceKy7eRo5OK
0o5mU/WgtMtJppcD0qSKxW3J3dueaba3zR/hRLe+c3v61OoyvqluH+ZeB2FZLwMzEb
fwrYnk8xdvpRa2SkbjVCsYUGiPdT7V4Hf2q7B4R+f5mP41uWkUcEu7oRUk8oYH+L8K
XMxcpUj0BY48IF+Wli0Xd7AVeiyIt235cY60B9vOanmZRVOirHjPNXrDSlXa23pQsi
gZcj2ol1byeFb2HFGoG1B5OxlMSg4zuHpVS5thcHGMc4AzVO01rL5Y4PqO9Tvq0cnV
ufXikMcnhvLfe5PpUp0BYju3bvr0qAasyNwabJrLSE/xfjQGhba9jsYdqj6c/8A1q5
7WNWaST7v6VZklaSVmx+GantNKku3DKi9epoEYsNtNKu5tyA9jTpNO+XO7PsK6j+wM

```
na7fNnGBWhp3hiFJ183aU75p8w+VnCDS5D/AHqK9SGgaeP7v/fQope0Hynm8tmUPQM
BUEyxr1Vh7Zq4WUj5i2ap3gw36g1SZJTnVT/CfaqrKrHpVyTk1DJb9TVEyREwTZ059
6iZg33R79KkaPccEU0RbfXOc0EhGmH3NVozqqDFVW3dqaXZh92gdy8t8oHP/wCumNq
CjpVTymJpTDnuaB8zJm1DLVKt8/aqaRFW7Vbgb5OlAK49L11bnNTRakSOtQhsmnA/7
IoKLi6mxjK7jtqGTUWJ71COaVY80APN4z/xGo5LsqDmnCHJ4oFluPOfx7UARx3zBsA
cVOt84pgt1h4prpzQBa+2MelSRSSPVWI7StXIZgo/+tSAuWUbFwf0ratbl1QDCrj1N
YkNyFqcX7GoA2hcrjnj9ani1XYeu7noQc1hDUcDn5qeNTUfwgUFcx0a62+0bd23HHz
0Vzv9qn+83/fVFLlQcxhrIy0jkSLgj8qg8xgOtHmMRWpJIluPrTpbFW6cfWoxNQ03p
QBG9jz2pHssr608ynHWk3n1oAg+xKDSi2X+7UuaCMigCFrZVPFOMeRjFSLHxTvIPtQ
BXWHHb9KkWLI+7Unlle2acNoPdaAIxDtGcU1k54p559acsJPXigCNV2inxpuPtTm2o
fu5prS0ASCNY+9BkFQ7qKAHMSxppTNG6jdQA7YoHX605X2Dv+NR7qC9AFhJcNxmpBP
7VTWTJpwmIosBbE+TzT1fjg1TE2KBNU8oFzzv9r9aKqeePeijlAioooqgAHFGc0Zoz
QAoNJRmjNADhSqVHUUwmjNAEnm4HAAppdj+NNooAdvOP/r0FvWm0UALnmkzRRQAUUU
UAIRS4oooACM0YoooAay5oCU6igAAoK8UZooAVV3Uvln/AD2puaM0AFFFFAEeaM0zc
aXeaAHZopm40bjQA+imbjS7zQA6jNN30m6gCTdSh6i3GjcaAJN9LvqPeaTcaAJd9G/
iot5pd5oAk30b6i3mjfQBLvo31FuNG6gCXfRvqLdRuNAEu+jfUW80bjQBLvo31FuNL
voAk30b6i3mjcaAJNxoqPcaKAIPNbFHmtTaKDMcJGAo8xvWm0UAOEjZo81sdabRQA7
zWoMjEU2igLilyaUyMe9NooAcZGNHmNjrTaKAuOEjDvR5rU2igB3mtnrR5jetNooAd
5jUeY3rTaKAHeY3rR5jZptFADhIw70ea3rTaKAHeY3rR5jY602igB3mNR5rU2igB3m
t60U2igLn/9k="
            }
          ]
        },
        {
          "fieldValues": [
            "2",
            "mdi.1234@foobarcorp.com"
          ],
          "optionalData": [
            {
              "name": "FIRST_NAME",
              "value": "jim_1"
            },
            {
              "name": "LAST_NAME",
              "value": "smith_1"
            }
          ],
          "attachmentData": [
            {
              "name": "Hello.pdf",
              "value":
"JVBERi0xLjQKJYCAgIAKMSAwIG9iago8PC9QYWdlcyAyIDAgUiAvVmlld2VyUHJlZ
mVyZW5jZXMgOCAwIFIgL1R5cGUgL0NhdGFsb2cgPj4KZW5kb2JqCjIgMCBvYmoKPDw
vQ291bnQgMSAvTWVkaWFCb3ggWzAgMCA1OTYgODQyIF0gL1R5cGUgL1BhZ2VzIC9SZ
XNvdXJjZXMgMyAwIFIgL0tpZHMgWzUgMCBSIF0gPj4KZW5kb2JqCjMgMCBvYmoKPDw
```

```
vRm9udCA3IDAgUiA+PgplbmRvYmoKNCAwIG9iago8PC9GaWx0ZXIgL0ZsYXRlRGVjb
2RlIC9MZW5ndGggNjAgPj4Kc3RyZWFtCnic0zdUMDZSCEnjcgrhMlQwAEKwgLm5sZ6
FuZGBoUJILpeGR2pOTr5CeH5RToqipkJIFpdrCBcAWMsNBgplbmRzdHJlYW0KZW5kb
2JqCjUgMCBvYmoKPDwvUGFyZW50IDIgMCBSIC9UeXBlIC9QYWdlIC9Db250ZW50cyA
0IDAgUiA+PgplbmRvYmoKNiAwIG9iago8PC9CYXNlRm9udCAvQ291cmllci1Cb2xkI
C9TdWJ0eXBlIC9UeXBlMSAvRW5jb2RpbmcgL1dpbkFuc2lFbmNvZGluZyAvVHlwZSA
vRm9udCA+PgplbmRvYmoKNyAwIG9iago8PC8xIDYgMCBSID4+CmVuZG9iago4IDAgb
2JqCjw8L0Rpc3BsYXlEb2NUaXRsZSB0cnVlID4+CmVuZG9iago5IDAgb2JqCjw8L1N
1YmplY3QgKP7/AFMAYQBtAHAAbABlACAAYQBiAG8AdQB0ACAAYQAgAHMAaQBtAHAAb
ABlACAAJwBoAGUAbABsAG8AIAB3AG8AcgBsAGQAJwAgAHUAcwBpAG4AZwAgAFAARAB
GACAAQwBsAG8AdwBuKSAvQ3JlYXRvciAo/v8AbwByAGcALgBwAGQAZgBjAGwAbwB3A
G4ALgBzAGEAbQBwAGwAZQBzAC4AYwBsAGkALgBIAGUAbABsAG8AVwBvAHIAbABkAFM
AYQBtAHAAbABlKSAvQXV0aG9yICj+/wBTAHQAZQBmAGEAbgBvACAAQwBoAGkAegB6A
G8AbABpAG4AaSkgL0NyZWF0aW9uRGF0ZSAoRDoyMDExMDMwNTIwMDYxNSswMScwMCc
pIC9Qcm9kdWNlciAo/v8AUABEAEYAIABDAGwAbwB3AG4AIABmAG8AcgAgAEoAYQB2A
GEAIAAwAC4AMQAuADApIC9UaXRsZSAo/v8AUABEAEYAIABDAGwAbwB3AG4AIAAtACA
ASABlAGwAbABvACAAdwBvAHIAbABkACAAcwBhAG0AcABsAGUpID4+CmVuZG9iagp4c
mVmCjAgMTAKMDAwMDAwMDAwMCA2NTUzNSBmDQowMDAwMDAwMDE1IDAwMDAwIG4NCjA
wMDAwMDAwODggMDAwMDAgbg0KMDAwMDAwMDE4NyAwMDAwMCBuDQowMDAwMjE5I
DAwMDAwIG4NCjAwMDAwMDAzNDkgMDAwMDAgbg0KMDAwMDAwMDQxMSAwMDAwMCBuDQo
wMDAwMDAwNTEwIDAwMDAwIG4NCjAwMDAwMDA1MzkgMDAwMDAgbg0KMDAwMDAwMDU4M
SAwMDAwMCBuDQp0cmFpbGVyCjw8L1Jvb3QgMSAwIFIgL0luZm8gOSAwIFIgL1NpemU
gMTAgPj4Kc3RhcnR4cmVmCjEwMzcKJSVFT0Y="
            }
          ]
        },
        {
          "fieldValues": [
            "3",
            "mdi.5678@foobarcorp.com"
          ],
          "optionalData": [
            {
              "name": "FIRST_NAME",
              "value": "jim_2"
            },
            {
              "name": "LAST_NAME",
              "value": "smith_2"
            }
          ],
          "attachmentData": [
            {
              "name": "Bye.asd",
              "value":
"JVBERi0xLjQKJYCAgIAKMSAwIG9iago8PC9QYWdlcyAyIDAgUiAvVmlld2VyUHJlZ
```

```
mVyZW5jZXMgOCAwIFIgL1R5cGUgL0NhdGFsb2cgPj4KZW5kb2JqCjIgMCBvYmoKPDw
vQ291bnQgMSAvTWVkaWFCb3ggWzAgMCA1OTYgODQyIF0gL1R5cGUgL1BhZ2VzIC9SZ
XNvdXJjZXMgMyAwIFIgL0tpZHMgWzUgMCBSIF0gPj4KZW5kb2JqCjMgMCBvYmoKPDw
vRm9udCA3IDAgUiA+PgplbmRvYmoKNCAwIG9iago8PC9GaWx0ZXIgL0ZsYXRlRGVjb
2RlIC9MZW5ndGggNjAgPj4Kc3RyZWFtCnic0zdUMDZSCEnjcgrhMlQwAEKwgLm5sZ6
FuZGBoUJILpeGR2pOTr5CeH5RToqipkJIFpdrCBcAWMsNBgplbmRzdHJlYW0KZW5kb
2JqCjUgMCBvYmoKPDwvUGFyZW50IDIgMCBSIC9UeXBlIC9QYWdlIC9Db250ZW50cyA
0IDAgUiA+PgplbmRvYmoKNiAwIG9iago8PC9CYXNlRm9udCAvQ291cmllci1Cb2xkI
C9TdWJ0eXBlIC9UeXBlMSAvRW5jb2RpbmcgL1dpbkFuc2lFbmNvZGluZyAvVHlwZSA
vRm9udCA+PgplbmRvYmoKNyAwIG9iago8PC8xIDYgMCBSID4+CmVuZG9iago4IDAgb
2JqCjw8L0Rpc3BsYXlEb2NUaXRsZSB0cnVlID4+CmVuZG9iago5IDAgb2JqCjw8L1N
1YmplY3QgKP7/AFMAYQBtAHAAbABlACAAYQBiAG8AdQB0ACAAYQAgAHMAaQBtAHAAb
ABlACAAJwBoAGUAbABsAG8AIAB3AG8AcgBsAGQAJwAgAHUAcwBpAG4AZwAgAFAARAB
GACAAQwBsAG8AdwBuKSAvQ3JlYXRvciAo/v8AbwByAGcALgBwAGQAZgBjAGwAbwB3A
G4ALgBzAGEAbQBwAGwAZQBzAC4AYwBsAGkALgBIAGUAbABsAG8AVwBvAHIAbABkAFM
AYQBtAHAAbABlKSAvQXV0aG9yICj+/wBTAHQAZQBmAGEAbgBvACAAQwBoAGkAegB6A
G8AbABpAG4AaSkgL0NyZWF0aW9uRGF0ZSAoRDoyMDExMDMwNTIwMDYxNSswMScwMCc
pIC9Qcm9kdWNlciAo/v8AUABEAEYAIABDAGwAbwB3AG4AIABmAG8AcgAgAEoAYQB2A
GEAIAAwAC4AMQAuADApIC9UaXRsZSAo/v8AUABEAEYAIABDAGwAbwB3AG4AIAAtACA
ASABlAGwAbABvACAAdwBvAHIAbABkACAAcwBhAG0AcABsAGUpID4+CmVuZG9iagp4c
mVmCjAgMTAKMDAwMDAwMDAwMCA2NTUzNSBmDQowMDAwMDAwMDE1IDAwMDAwIG4NCjA
wMDAwMDAwODggMDAwMDAgbg0KMDAwMDAwMDE4NyAwMDAwMCBuDQowMDAwMDAwMjE5I
DAwMDAwIG4NCjAwMDAwMDAzNDkgMDAwMDAgbg0KMDAwMDAwMDQxMSAwMDAwMCBuDQo
wMDAwMDAwNTEwIDAwMDAwIG4NCjAwMDAwMDA1MzkgMDAwMDAgbg0KMDAwMDAwMDU4M
SAwMDAwMCBuDQp0cmFpbGVyCjw8L1Jvb3QgMSAwIFIgL0luZm8gOSAwIFIgL1NpemU
gMTAgPj4Kc3RhcnR4cmVmCjEwMzcKJSVFT0Y="
            }
          ]
        }
      ],
      "fieldNames": [
        "CUSTOMER_ID_",
        "EMAIL_ADDRESS_"
      ]
    },
    "mergeRule": {
      "htmlValue": "H",
      "matchColumnName1": "EMAIL_ADDRESS_",
      "matchColumnName2": null,
      "optoutValue": "O",
      "optinValue": "I",
      "insertOnNoMatch": true,
      "defaultPermissionStatus": "OPTIN",
      "rejectRecordIfChannelEmpty": "E",
      "updateOnMatch": "REPLACE_ALL",
      "textValue": "T",
      "matchOperator": "NONE"
```

```
        }
    }
```

Response Body Properties

Same as the standard Merge Trigger Email with Attachments response.

Sample Response:

The response returns a status for each recipient record sent in the request payload. In this example:

- The first record was successfully merged and the email was sent to the recipient whose RIID is 72067.
- The second record was successfully merged, but the email was undeliverable because the recipient's email deliverability status in his Profile List record was set to "Undeliverable" in Responsys.
- The third record was successfully merged, but the email was not sent because the attachment had an incorrect file type (Bye.asd).

```
[
  {
    "errorMessage": null,
    "success": true,
    "recipientId": 72067
  },
  {
    "errorMessage": "RECIPIENT_STATUS_UNDELIVERABLE: Recipient deliverability status
is undeliverable",
    "success": false,
    "recipientId": -1
  },
  {
    "errorMessage": "FAILURE: Invalid attachment type, allowed types are png, jpg, jpeg,
pdf, ical",
```

```
    "success": false,
    "recipientId": 72068
  }
]
```

Merge members into a profile list & trigger SMS messages to them (HA Merge Trigger SMS)

Use this interface to merge members into a profile list and trigger SMS messages to them.

Service URL:

/rest/haApi/v1.3/campaigns/{campaignName}/sms

> **NOTE:** For this request, ensure that your path uses haApi instead of api. The haApi part of the path is case sensitive.

Required Path Parameters:

`campaignName` - Name of the SMS campaign to be sent to the recipients.

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body Properties:

Same as the standard Merge Trigger SMS request.

NOTE: Using this API requires adhering to several constraints and guidelines, which are described in the standard Responsys REST API reference guide.

Sample Request Body:

```
{
"mergeTriggerRecordData": {
  "mergeTriggerRecords": [
    {
      "fieldValues": ["1001", "foo.bar@oracle.com", "6505551212", "US"],
      "optionalData": [
        { "name": "CITY_", "value": "San Bruno" }
      ]
    },
    {
      "fieldValues": ["1002", "baz.foo@oracle.com", "6505551212", "US"],
      "optionalData": [
        { "name": "CITY_", "value": "San Francisco" }
      ]
    }
  ],
  "fieldNames": [
    "CUSTOMER_ID_",
    "EMAIL_ADDRESS_",
    "MOBILE_NUMBER_",
    "MOBILE_COUNTRY_"
  ]
},
"mergeRule": {
  "htmlValue": "H",
  "matchColumnName1": "CUSTOMER_ID_",
  "matchColumnName2": null,
  "optoutValue": "O",
  "optinValue": "I",
  "insertOnNoMatch": true,
  "defaultPermissionStatus": "OPTIN",
```

```
  "rejectRecordIfChannelEmpty": "E",
  "updateOnMatch": " NO_UPDATE ",
  "textValue": "T",
  "matchOperator": "NONE"
  }
}
```

Response Body Properties

Same as the standard Merge Trigger SMS response.

Sample Response:

```
[
  {
    "errorMessage" : null,
    "success" : true,
    "recipientId" : 72067
  },
  {
    "errorMessage" :" RECIPIENT_STATUS_UNDELIVERABLE: Recipient deliverability status
is undeliverable ",
    "success" : false,
    "recipientId" : -1
  }
]
```

Trigger mobile push messages (HA Trigger Push Message)

Use the following API to trigger push messages to existing members of a Profile List and its corresponding App Channel List.

NOTES:

- The system only replicates the Profile List and App Channel List to the AFTM secondary account. No App Channel Profile Extension Tables (PETs) will be replicated. If the push message needs data from an App Channel List PET, there will be problems triggering it from

a secondary account.

- Trigger Push Message will only work for recipients who are using the mobile app associated with the campaign. For example, if you use listType=Profile and the Profile List member does not have a device registered, the list member will not receive the message. The response payload for the recipient in this situation is NO_RECIPIENT_FOUND.
- See the Trigger Push Message topic in the *REST API for Oracle Responsys Marketing Cloud Service* guide for more important information about using this API.

Service URL:

/rest/haApi/v1.3/campaigns/{campaignName}/push

NOTE: For this request, ensure that your path uses haApi instead of api. The haApi part of the path is case sensitive.

Required Path Parameters:

`campaignName` - Name of the push campaign to be sent to the recipients.

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body Properties:

Same as the standard Trigger Push Message request.

NOTE: Using this API requires adhering to several constraints and guidelines, which are described in the standard Responsys REST API reference guide.

Sample Request Body:

```
{
   "recipientData": [
      {
         "customerId": null,
         "emailAddress": null,
         "recipientId": null,
         "mobileNumber": null,
         "emailSHA256Hash": null,
         "emailMD5Hash": null,
         "deviceId": "device_Id_value",
         "apiKey": "API_Key_value",
         "listType": "PUSH",
         "optionalData": [
            {
               "name": "CUSTOM1",
               "value": "c1a_value_new"
            },
            {
               "name": "CUSTOM2",
               "value": "c2a_value_new"
            }
         ]
      },
      {
         "customerId": null,
         "emailAddress": "foo.bar@oracle.com",
         "recipientId": null,
         "mobileNumber": null,
         "emailSHA256Hash": null,
         "emailMD5Hash": null,
         "deviceId": null,
         "apiKey": "API_Key_value",
         "listType": "PROFILE",
         "optionalData": [
            {
               "name": "CUSTOM1",
```

```
            "value": "c1a_value_new"
          },
          {
            "name": "CUSTOM2",
            "value": "c2a_value_new"
          }
        ]
      }
  ]
}
```

Response Body Properties

Same as the standard Trigger Push Message response. See the Response Notes paragraphs in the Examples section for an explanation of how the response may vary based on the CHANNEL_DELIVERABILITY_STATUS and the CHANNEL_PERMISSION_STATUS settings in the App Channel List.

Sample Response:

The response returns information about each record sent in the request. In this example, the mobile push message was sent successfully to the recipient device whose App Channel List RIID is 72067. (Unlike the responses for Email and SMS, the recipientId in a Trigger Push Message response is always the App Channel List RIID.) For the second record, the trigger push message action was not successful, because listType was set to PROFILE, a record was present in the Profile List, CHANNEL_DELIVERABILITY_STATUS_ is D (Deliverable) in the App Channel List, but the recipient's CHANNEL_PERMISSION_STATUS_ in the App Channel List was set to "O" (OptOut).

```
[
  {
```

```
    "errorMessage" : null,
    "success" : true,
    "recipientId" : 72067
  },
  {
    "errorMessage" :" RECIPIENT_STATUS_UNDELIVERABLE: Recipient deliverability status
is undeliverable ",
    "success" : false,
    "recipientId" : -1
  }
]
```


Merge members into a profile list (HA Merge List Members)

Use this interface to merge members into a profile list. Your client application can add new members to an existing profile list, and it can also update existing members of the profile list.

Service URL:

/rest/haApi/v1.3/lists/{listName}/members

> **NOTE:** For this request, ensure that your path uses haApi instead of api. The haApi part of the path is case sensitive.

Required Path Parameters:

`listName` - Name of the Profile List to which Responsys will merged the recipients in the request body.

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body Properties:

Same as the standard Merge List Recipients request.

NOTE: Using this API requires adhering to several constraints and guidelines, which are described in the standard Responsys REST API reference guide.

Sample Request Body:

NOTE: In the following example, the fourth record (record 3, because the array sequence begins with zero) contains an invalid value for the RIID_, so that we can demonstrate the error in the sample response.

```
{
  "recordData" : {
    "fieldNames" : ["riid_", "mobile_number_", "email_address_"],
    "records" : [
      ["4094326", "9845349498","ab.cd@example.com"],
      ["4094327", "9844444444","unknown@example.com"],
      ["4094328", "9844444666","abc@example.com"],
      ["ssdcf", "9844444444","xyz@example.com"]
    ],
    "mapTemplateName" : null
  },
  "mergeRule" : {
    "htmlValue" : "H",
    "optinValue" : "I",
    "textValue" : "T",
```

```
    "insertOnNoMatch" : true,
    "updateOnMatch" : "REPLACE_ALL",
    "matchColumnName1" : "RIID_",
    "matchColumnName2" : null,
    "matchOperator" : "NONE",
    "optoutValue" : "O",
    "rejectRecordIfChannelEmpty" : null,
    "defaultPermissionStatus" : "OPTIN"
  }
}
```

Response Body Properties

Same as the standard Merge List Recipients response, except that the HA Merge List Members API only returns the “self” link.

Sample Response:

NOTE: For the first three records (0, 1, and 2 of the array), the merge occurred successfully. The fourth record (record 3 of the array) was not merged, because “ssdcf” is not a valid value for the RIID_ field.

```
{
  "recordData": {
    "fieldNames": ["RIID_"],
    "records": [
      ["4094326"],
      ["4094327"],
      ["4094328"],
      ["MERGEFAILED: Record 3 = INVALID_PARAMETER: The value ssdcf is not
       valid for an integer field\n\n\r\n"]
    ],
    "mapTemplateName": null
  },
  "mergeRule": {
    "textValue": "T",
    "insertOnNoMatch": true,
```

```
    "updateOnMatch": "REPLACE_ALL",
    "matchOperator": "NONE",
    "matchColumnName3": null,
    "matchColumnName1": "RIID_",
    "matchColumnName2": null,
    "optinValue": "I",
    "optoutValue": "O",
    "rejectRecordIfChannelEmpty": null,
    "htmlValue": "H",
    "defaultPermissionStatus": "OPTIN"
  },
  "links": [
    {
      "rel": "self",
      "href": "/rest/api/v1.3/lists/DemoNewsLetterList/members",
      "method": "POST"
    }
  ]
}
```


Handling errors and exceptions

Detecting failovers

When a failover occurs, AFTM-supported requests are routed to the failover account. Assuming that your code used your account-specific endpoint, the authorization token should still be valid for the failover account.

However, we still recommend that your code watch for exceptions or lack of response.

- One way your client code can watch for exceptions or lack of response is to periodically make a standard read-only request, such as "Fetch All Profile Lists", and watch for responses other than the standard HTTPS 200 Success response. If you receive an "API_NOT_ALLOWED_IN_SECONDARY" (HttpStatus.UNAUTHORIZED, HTTPS status code 401), a failover has occurred and your client code may only use AFTM-supported endpoints.

- If you receive an “INVALID_TOKEN” error (HttpStatus.UNAUTHORIZED, HTTPS status code 401), your code must re-login to your account-specific endpoint using the same credentials that you used to access your primary Interact account.
- If you receive an “UNRECOVERABLE_EXCEPTION” error (HttpStatus.INTERNAL_SERVER_ERROR, HTTPS status code 500), your code should wait about 30 seconds and then resubmit the request only once.
- If there is no system response, your code can retry up to three times, with two minute intervals, to prevent the API login being disabled. If you still do not receive a response, contact Oracle Support.

For more information, see “Handling System Outages” in the standard REST API guide.

Other errors

AFTM only supports promotional campaigns in primary: You can use APIs such as HaMergeTriggerEmail for both promotional and transactional campaigns in primary. However, merge and trigger for promotional campaigns will fail in the secondary pod during failover.

When this occurs, the API will return the below sample error.

```
{
  "type": "",
  "title": "Invalid request parameters",
  "errorCode": "INVALID_PARAMETER",
  "detail": " <Campaign Name> must be Transactional campaign",
  "errorDetails": []
}
```

Asynchronous AFTM WS API

AFTM provides asynchronous endpoints for the Merge Members into a Profile List, Merge Trigger Email, and Merge Trigger SMS endpoints. These endpoints are asynchronous alternatives to the REST-based endpoints.

Merge members into a profile list (HA Async Merge List Members)

Merge members into a profile list & trigger email messages to them (HA Async Merge Trigger Email)

Merge members into a profile list & trigger SMS messages to them (HA Async Merge Trigger SMS)

Polling API endpoint

Merge members into a profile list (HA Async Merge List Members)

Provides an asynchronous interface to merge members into a profile list. Your client application can add new members to an existing profile list, and it can also update existing members of the profile list.

Service URL:

/rest/haAsyncApi/v1.3/lists/{listName}/members

NOTE: For this request, ensure that your path uses haAsyncApi instead of haApi. The haAsyncApi part of the path is case sensitive.

Required Path Parameters:

`listName` - Name of the list.

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body Properties:

Same as the standard Merge List Recipients request.

NOTE: Using this API requires adhering to several constraints and guidelines, which are described in the standard Responsys REST API reference guide.

Sample Request Body:

```
{
  "recordData" : {
    "fieldNames" : ["riid_", "mobile_number_", "email_address_"],
    "records" : [
      ["4094326", "9845349498","ab.cd@example.com"],
```

```
    ["4094327", "9844444444","unknown@example.com"],
    ["4094328", "9844444666","abc@example.com"]
   ],
   "mapTemplateName" : null
  },
  "mergeRule" : {
   "htmlValue" : "H",
   "optinValue" : "I",
   "textValue" : "T",
   "insertOnNoMatch" : true,
   "updateOnMatch" : "REPLACE_ALL",
   "matchColumnName1" : "RIID_",
   "matchColumnName2" : null,
   "matchOperator" : "NONE",
   "optoutValue" : "O",
   "rejectRecordIfChannelEmpty" : null,
   "defaultPermissionStatus" : "OPTIN"
  }
}
```

Sample Response:

NOTES:

- A successful response for the asynchronous API returns an HTTPS status code of 202 Accepted.
- A successful response returns a requestId, to be used in the Polling API endpoint to retrieve the status of your request.
- See Common error responses for the Asynchronous API for details about the error responses.

```
{
  "requestId": "TVVjVTY2OjoxNTc2MDk0NjQ5",
  "apiName": "HaAsyncMergeListMembers",
  "status": "REQUEST_ACCEPTED",
  "links": [
```

```
  {
   "rel": "self",
   "href": "/rest/haAsyncApi/v1.3/lists/listName/members",
   "method": "POST"
  },
  {
   "rel": "asyncResponse",
   "href": "/rest/haAsyncApi/v1.3/requests/TVVjVTY2OjoxNTc2MDk0NjQ5",
   "method": "GET"
  }
 ]
}
```


Merge members into a profile list & trigger email messages to them (HA Async Merge Trigger Email)

Use this interface to merge members into a profile list and subsequently send them an email message. This endpoint can be used for triggering both transactional and promotional emails. However, promotional emails will fail on the secondary account during failover. Both merging members and the email trigger will fail.

Service URL:

/rest/haAsyncApi/v1.3/campaigns/{campaignName}/email

> **NOTE:** For this request, ensure that your path uses haAsyncApi instead of haApi. The haAsyncApi part of the path is case sensitive.

Required Path Parameters:

`campaignName` - Name of the Email campaign to be sent to the recipients.

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body Properties:

`mergeTriggerRecordData` – Record data that represents Field Names and Values for the recipient. For more information, see the standard Merge Trigger Email request example.

`mergeRule` – Merge rule used to merge recipient records in a Profile List. Only used during the merge operation. For more information, see the standard Merge Trigger Email request example.

Sample Request Body:

```
{
   "mergeTriggerRecordData": {
      "mergeTriggerRecords": [{
            "fieldValues": [
               "mdi1234@foobar.com",
               "martiness",
               "6505551212",
               "US"
            ],
            "optionalData": [{
                  "name": "FIRST_NAME",
                  "value": "jim_1"
               },
               {
```

```
                "name": "LAST_NAME",
                "value": "smith_1"
            }
        ]

    },
    {
        "fieldValues": [
            "mdi.1234@foobarcorp.com",
            "concord",
            "6505551212",
            "US"
        ],
        "optionalData": [{
                "name": "FIRST_NAME",
                "value": "jim_2"
            },
            {
                "name": "LAST_NAME",
                "value": "smith_2"
            }
        ]

    }
  ],
  "fieldNames": [
    "EMAIL_ADDRESS_",
    "CITY_",
    "MOBILE_NUMBER_",
    "MOBILE_COUNTRY_"
  ]
},
"mergeRule": {
  "htmlValue": "H",
  "matchColumnName1": "EMAIL_ADDRESS_",
  "matchColumnName2": null,
  "optoutValue": "O",
  "insertOnNoMatch": true,
  "defaultPermissionStatus": "OPTIN",
  "rejectRecordIfChannelEmpty": "E",
```

```
      "optinValue": "I",
      "updateOnMatch": "REPLACE_ALL",
      "textValue": "T",
      "matchOperator": "NONE"
    }
}
```

Sample Response:

NOTES:

- A successful response for the asynchronous API returns an HTTPS status code of 202 Accepted.
- A successful response returns a requestId, to be used in the Polling API endpoint to retrieve the status of your request.
- See Common error responses for the Asynchronous API for details about the error responses.

```
{
 "requestId": "aWpDUHozOjoxNTU4NTQyMTI1",
 "apiName": "HaAsyncMergeTriggerEmail",
 "status": "REQUEST_ACCEPTED",
 "links": [
  {
   "rel": "self",
   "href": "/rest/haAsyncApi/v1.3/campaigns/campTest/email",
   "method": "POST"
  },
  {
   "rel": "asyncResponse",
   "href": "/rest/haAsyncApi/v1.3/requests/aWpDUHozOjoxNTU4NTQyMTI1",
   "method": "GET"
  }
 ]
}
```

Merge members into a profile list & trigger SMS messages to them (HA Async Merge Trigger SMS)

Use this interface to merge members into a profile list and trigger SMS messages to them.

Service URL:

/rest/haAsyncApi/v1.3/campaigns/{campaignName}/sms

> **NOTE:** For this request, ensure that your path uses haAsyncApi instead of haApi. The haAsyncApi part of the path is case sensitive.

Required Path Parameters:

`campaignName` - Name of the SMS campaign to be sent to the recipients.

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body Properties:

`mergeTriggerRecordData` – Record data that represents Field Names and Values for the recipient. For more information, see the standard Merge Trigger SMS request

example.

`mergeRule` – Merge rule used to merge recipient records in a Profile List. Only used during the merge operation. For more information, see the standard Merge Trigger SMS request example.

Sample Request Body:

```
{
   "mergeTriggerRecordData": {
      "mergeTriggerRecords": [{
            "fieldValues": [
               "mdi1234@foobar.com",
               "martiness",
               "6505551212",
               "US"
            ],
            "optionalData": [{
                  "name": "FIRST_NAME",
                  "value": "jim_1"
               },
               {
                  "name": "LAST_NAME",
                  "value": "smith_1"
               }
            ]

         },
         {
            "fieldValues": [
               "mdi.1234@foobarcorp.com",
               "concord",
               "6505551212",
               "US"
            ],
            "optionalData": [{
                  "name": "FIRST_NAME",
                  "value": "jim_2"
```

```
            },
            {
               "name": "LAST_NAME",
               "value": "smith_2"
            }
         ]

      }
   ],
   "fieldNames": [
      "EMAIL_ADDRESS_",
      "CITY_",
      "MOBILE_NUMBER_",
      "MOBILE_COUNTRY_"
   ]
  },
  "mergeRule": {
     "htmlValue": "H",
     "matchColumnName1": "EMAIL_ADDRESS_",
     "matchColumnName2": null,
     "optoutValue": "O",
     "insertOnNoMatch": true,
     "defaultPermissionStatus": "OPTIN",
     "rejectRecordIfChannelEmpty": "E",
     "optinValue": "I",
     "updateOnMatch": "REPLACE_ALL",
     "textValue": "T",
     "matchOperator": "NONE"
  }
}
```

Sample Response:

NOTES:

- A successful response for the asynchronous API returns an HTTPS status code of 202 Accepted.

- A successful response returns a requestId, to be used in the Polling API endpoint to retrieve the status of your request.
- See Common error responses for the Asynchronous API for details about the error responses.


```
{
 "requestId": "eUYycDdoOjoxNTU2MDEwODEz",
 "apiName": "HaAsyncMergeTriggerEmail",
 "status": "REQUEST_ACCEPTED",
 "links": [
  {
   "rel": "self",
   "href": "/rest/haAsyncApi/v1.3/campaigns/campTest/email",
   "method": "POST"
  },
  {
   "rel": "asyncResponse",
   "href": "/rest/haAsyncApi/v1.3/requests/eUYycDdoOjoxNTU2MDEwODEz",
   "method": "GET"
  }
 ]
}
```


Trigger Push Message API (HA Async)

Use the following interface to trigger Push campaign messages to existing members of a Profile List and its corresponding App Channel List. This endpoint is the AFTM alternative to Trigger Push Message API (Asynchronous)

Service URL:

/rest/haAsyncApi/v1.3/campaigns/{campaignName}/push

NOTE: For this request, ensure that your path uses haAsyncApi instead of haApi. The haAsyncApi part of the path is case sensitive.

Required Path Parameters:

`campaignName` - Name of the SMS campaign to be sent to the recipients.

Request Method:

POST

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Request Body Properties:

See the Trigger Push Message Endpoints page in the Oracle Responsys REST API Guide. View the Request body section of the topic.

Request Notes:

- You can send push messages to existing Responsys Push campaigns for to up to 200 members of a Profile List and its corresponding App Channel List.
- The request payload allows specifying one of the available Profile List attributes, so that you may uniquely identify the recipients of the push message.
- The Profile List attributes that can be specified are one of `recipientId`, `customerId`, `emailAddress`, `mobileNumber`, `emailSHA256Hash`, or `emailMD5Hash`.

Note: In the request body, all `recipientData` must be present, but the Profile List attributes you are not using, must be set to `null`.

- Use the `"listType": "PROFILE"` when recipients need to be matched from Profile List (that is, known recipients).
    - The Profile List members selected using these attributes are matched against the App Channel list to find the device IDs of all devices to trigger push messages.
    - You can send to all devices and apps that a mobile app user has installed by setting `"apiKey": null`.
    - You can send to a specific mobile app that a user has installed by specifying the `apiKey` value. Note that you may only use one apiKey value per recipient. This means that if a mobile app end user (recipient) has a single entry in a Profile List but has multiple entries in the App Channel List for different apps, the `apiKey` attribute restricts the push message to trigger for one of those apps.
- Use the `"listType": "PUSH"` when recipients need to be matched from App Channel List (that is, unknown recipients).
    - In the request body, all `recipientData` properties must be present, but the Profile List attributes you are not using must be set to `null`.
    - When using `"listType": "PUSH"`, the App Channel List attributes that you must specify are both `deviceId` and `apiKey`. Including both `deviceId` and `apiKey` ensures that push messages will only be triggered for devices that match the `apiKey`.
- You can use the `optionalData` attribute as part of the payload is used for customizing the push message.

- Do not use system-defined fields (for example, `MOBILE_NUMBER_`) for your optional data.
- To pass extended/accented characters in `optionalData` payload, they must be escaped as Unicode characters. For example, the euro currency symbol is escaped as \u20AC, the yen currency symbol is escaped as \u00A5, an umlauted u is escaped as \u00FC, an accented e is escaped as \u00E9, and the like. Otherwise, you may receive an `INVALID_REQUEST_CONTENT` error.

- This endpoint supports sending reference data in requests. To learn how you can retrieve this data when polling, refer to Returning reference data for asynchronous requests.

Sample Request Body:

```
{
 "recipientData": [
  {
   "customerId": null,
   "emailAddress": null,
   "recipientId": 131006807,
   "mobileNumber": null,
   "emailSHA256Hash": null,
   "emailMD5Hash": null,
   "deviceId": "",
   "apiKey": "",
   "listType": "PROFILE",
   "optionalData": [
    {
     "name": "CUSTOM1",
     "value": "WELCOME10"
    },
    {
     "name": "CUSTOM2",
     "value": "COUPON"
    }
   ]
  },
  {
   "customerId": null,
   "emailAddress": null,
```

```
    "recipientId": 100007327,
    "mobileNumber": null,
    "emailSHA256Hash": null,
    "emailMD5Hash": null,
    "deviceId": "",
    "apiKey": "",
    "listType": "PROFILE",
    "optionalData": [
     {
      "name": "CUSTOM1",
      "value": "DISCOUNT10"
     },
     {
      "name": "CUSTOM2",
      "value": "COUPON"
     }
    ]
   }
  ]
}
```


Sample Response:

NOTES:

- A successful response for the asynchronous API returns an HTTPS status code of 202 Accepted.
- A successful response returns a requestId, to be used in the Polling API endpoint to retrieve the status of your request.
- See Common error responses for the Asynchronous API for details about the error responses.


```
{
 "requestId": "YjN1aWowOjoxNjQzOTUyMjI0",
 "apiName": "HaAsyncTriggerPush",
 "status": "REQUEST_ACCEPTED",
```

```
 "links": [
  {
   "rel": "self",
   "href": "/rest/haAsyncApi/v1.3/campaigns/Summer_Push/push",
   "method": "POST"
  },
  {
   "rel": "asyncResponse",
   "href": "/rest/haAsyncApi/v1.3/requests/YjN1aWowOjoxNjQzOTUyMjI0",
   "method": "GET"
  }
 ]
}
```

Polling API endpoint

Use the Polling API to find out the status of your async request.

Note: You must send one request for each requestId you want to query. This request does not support sending multiple requestIds per request.

Service URL:

/rest/haAsyncApi/v1.3/requests/{requestId}

NOTE: For this request, ensure that your path uses haAsyncApi instead of haApi. The haAsyncApi part of the path is case sensitive.

Required Path Parameters:

`requestId` - Unique identifier for the API request. Obtain this value from the successful response to your async request.

Request Method:

GET

Request Header:

Authorization=<AUTH_TOKEN>

Content-Type=application/json

Sample Response:

NOTES:

- A successful response for the asynchronous API returns an HTTPS status code of 202 Accepted.
- See Common error responses for the Polling API for details about the error responses.
- Note that the date and time values consider the Responsys account's time zone.


```
{
  "response": [
    {
      "recipientId": 39001321,
      "success": true,
      "errorMessage": null
    },
    {
      "recipientId": 39001501,
      "success": true,
      "errorMessage": null
    }
```

```
    ],
    "requestId": "UGpCVnBCOjoxNTQwNTQ2MzQ5",
    "apiName": "HaAsyncMergeTriggerEmail",
    "timeZone": "India Standard Time",
    "requestTime": "2018-10-26T15:02:29.344",
    "requestProcessedTime": "2018-10-26T09:48:18.000",
    "status": "SUCCESS"
}
```

Common error responses for the Asynchronous API

These are common error responses for the Asynchronous API.

Request payload is not well-formed JSON (INVALID_REQUEST_CONTENT)

The system validates whether or not the request payload is well-formed JSON. If it is not, the system returns the following response with HTTPS Status Code 400 (Bad Request). Verify that your payload is in proper JSON format, and that it does not have missing or incorrectly used punctuation.

```
{
   "type": "",
   "title": "Invalid request parameters",
   "errorCode": "INVALID_PARAMETER",
   "detail": "Invalid parameter 'mergeTriggerRecordData'",
   "errorDetails": []
}
```

Recipient limit is exceeded (RECIPIENT_LIMIT_EXCEEDED)

If your client application submits an async request with more than 200 recipients, it is rejected immediately and is not put on the queue. Limit your requests to 200 or fewer recipients.

```
{
   "type": "",
   "title": "Recipient limit exceeded",
   "errorCode": "RECIPIENT_LIMIT_EXCEEDED",
```

```
  "detail": "Recipient limit exceeded, maximum of 200 recipients are allowed per each api
call",
  "errorDetails": []
}
```

Queue is full (UNABLE_TO_SUBMIT_ASYNC_REQUEST)

If the request cannot be accepted due to queue being full, the following response is returned with HTTPS Status Code 500 (Internal Server Error). Try the request again later.

```
{
  "type": "",
  "title": "Unable to submit Async request",
  "errorCode": "UNABLE_TO_SUBMIT_ASYNC_REQUEST",
  "detail": "Error in submitting async request",
  "errorDetails": []
}
```

Common error responses for the Polling API

Unless otherwise noted, each of the following has an HTTPS status code of 200 OK, because the polling request itself was considered successful, even if it contains an error response from the system.

Response purged (RESPONSE_PURGED)

The system returns this response when the async job status response is removed from the queue. Responses are retained in the system for 24 hours. Client applications must poll for responses within 24 hours after the initial API request was submitted.


```
{
   "response": {
      "type": "",
      "title": "Response purged",
      "errorCode": "RESPONSE_PURGED",
      "detail": "Response Message Purged",
      "errorDetails": "[]"
   },
   "requestId": "WGlJSWhsOjoxNTIxMzU5OTI1",
   "apiName": "",
   "timeZone": null,
   "requestTime": "2018-03-18 13:28:45.0",
   "requestProcessedTime": "",
   "status": "UNKNOWN"
}
```


Async API not enabled (ASYNC_API_DISABLED_FOR_USER)

The system returns this response when the async API is not enabled for the user performing the request. Contact your CSM to ensure your user and account meet the

prerequisites.


```
{
   "type": "",
   "title": "ASYNC API disabled for user",
   "errorCode": "ASYNC_API_DISABLED_FOR_USER",
   "detail": "ASYNC API disabled for user",
   "errorDetails": []
}
```


Processing request (PROCESSING_REQUEST)

The system returns this response when your request is still in progress.


```
{
   "response": {
      "type": "",
      "title": "Still processing request",
      "errorCode": "PROCESSING_REQUEST",
      "detail": "Processing Request",
      "errorDetails": "[]"
   },
   "requestId": "Q3hvOUZtOjoxNTU3NDgxMjY3",
   "apiName": "",
   "timeZone": null,
   "requestTime": "2019-05-10 15:11:07.0",
   "requestProcessedTime": "",
   "status": "IN_PROGRESS"
}
```


Campaign is invalid (CAMPAIGN_IS_INVALID)

The system returns this response when your request is still in progress.


```
{
   "response": {
      "errorCode": "CAMPAIGN_IS_INVALID",
      "detail": "An error has occurred and the system is unable to process your request at this
```

time. Please contact Responsys Support for further assistance.",
      "type": "",
      "title": "Not a valid campaign",
      "errorDetails": "[]"
   },
   "requestId": "UGpCVnBCOjoxNTQwNTQ2MzQ5",
   "apiName": "HaAsyncMergeTriggerEmail",
   "timeZone": "India Standard Time",
   "requestTime": "2018-10-26T15:02:29.344",
   "requestProcessedTime": "2018-10-26T09:48:18.000",
   "status": "ERROR"
}

Merge fails for one recipient

The system returns this response when the merge fails for a recipient.

{
   "response": [
      {
         "recipientId": 0,
         "success": false,
         "errorMessage": null
      },
      {
         "recipientId": 0,
         "success": false,
         "errorMessage": "FAILURE: Record 1 = INVALID_PARAMETER:matchColumnName1 is not found in fieldNames or corresponding value is null or empty\r\n"
      }
   ],
   "requestId": "UGpCVnBCOjoxNTQwNTQ2MzQ5",
   "apiName": "HaAsyncMergeTriggerEmail",
   "timeZone": "India Standard Time",
   "requestTime": "2018-10-26T15:02:29.344",
   "requestProcessedTime": "2018-10-26T09:48:18.000",
   "status": "SUCCESS"
}

Request ID is not valid (INVALID_REQUEST_ID)

The system returns this response when the requestId provided is not valid. Verify the requestId is correct.


```
{
   "response": {
      "type": "",
      "title": "Invalid Request Id",
      "errorCode": "INVALID_REQUEST_ID",
      "detail": "Request Id is not valid ",
      "errorDetails": "[]"
   },
   "requestId": "UG",
   "apiName": null,
   "timeZone": null,
   "requestTime": null,
   "requestProcessedTime": null,
   "status": "ERROR"
}
```