

How to Remove Contacts from an Active Salesforce Marketing Cloud Journey Using REST API

A step-by-step technical guide using Journey Builder Contact Exit API

Salesforce Marketing Cloud • Journey Builder • REST API

1. Introduction

In Salesforce Marketing Cloud Journey Builder, there are situations where a Journey is already active, but a specific set of contacts should no longer continue through that Journey.

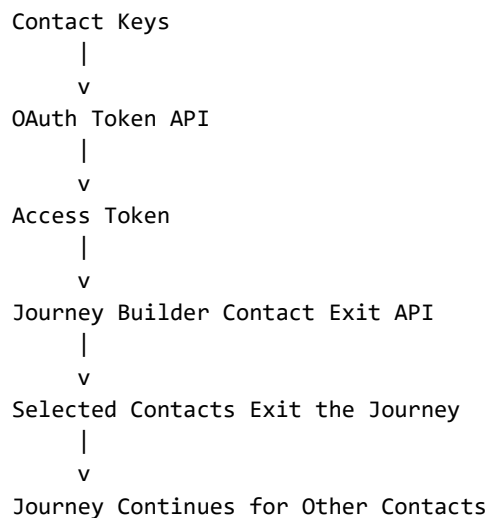
Typical examples include:

- A customer has opted out of communication.
- A lead is no longer eligible for the Journey.
- A customer has completed the required business process.
- A business rule has changed and selected contacts must exit.
- A large list of contacts needs to be removed programmatically.

Stopping the Journey is not appropriate when only selected contacts should be removed. Salesforce Marketing Cloud provides a Journey Builder REST API that can remove specific contacts while allowing the Journey to remain active for other contacts.

2. What We Will Build

The approach covered in this guide is REST API only. The flow is:



3. The API We Need

The Journey Builder Contact Exit API is:

POST /interaction/v1/interactions/contactexit

This endpoint is used to request that specified contacts exit a Journey version. The Journey itself does not need to be stopped just because selected contacts are being removed.

4. Prerequisites

Before making the API call, collect the following information:

Requirement	Purpose
Client ID	OAuth authentication

Client Secret	OAuth authentication
Authentication Base URI	Used to request an access token
REST Base URI	Used to call the Journey Builder REST API
Journey Definition Key	Identifies the Journey
Journey Version	Identifies the version to target
Contact Keys	Identifies the contacts to remove

The Client ID and Client Secret are normally obtained from an Installed Package in Salesforce Marketing Cloud. The package must have the API permissions required for the operations being performed.

5. Step 1 — Identify the Journey Definition Key

The DefinitionKey identifies the Journey definition. Before sending the Contact Exit request, make sure you have the correct Journey definition key and the version that you intend to target.

For example:

```
DefinitionKey = "VTP_Direct_Program_Journey"
```

6. Step 2 — Identify the Journey Version

A Journey can have multiple versions. The Contact Exit request includes a Versions array, so use the appropriate version for the Journey you are targeting.

```
"Versions": [
  3
]
```

Do not assume that version 1 is always the correct version. Verify the Journey version before making a production request.

7. Step 3 — Identify the Contact Keys

The ContactKey values must correspond to the contact identity used by the Journey. For example, if the Journey uses a Lead ID as ContactKey, send those Lead IDs—not an unrelated email address.

```
"ContactKey": [
  "10001",
  "10002",
  "10003"
]
```

8. Step 4 — Generate an OAuth Access Token

The first API operation is to authenticate with Marketing Cloud.

Endpoint:

```
POST https://YOUR_SUBDOMAIN.auth.marketingcloudapis.com/v2/token
```

Headers:

```
Content-Type: application/json
```

Request body:

```
{
  "grant_type": "client_credentials",
  "client_id": "YOUR_CLIENT_ID",
  "client_secret": "YOUR_CLIENT_SECRET"
}
```

A successful response contains an access_token:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIs... ",
  "expires_in": 1080,
  "token_type": "Bearer",
  "rest_instance_url": "https://YOUR_SUBDOMAIN.rest.marketingcloudapis.com/"
}
```

Store the access token securely and use it as a Bearer token in the next API request. Do not expose Client Secret or access tokens in public documentation, screenshots, source control, or client-side code.

9. Step 5 — Call the Contact Exit API

Once the access token is available, call:

```
POST
https://YOUR_SUBDOMAIN.rest.marketingcloudapis.com/interaction/v1/interactions/contactexit
```

Headers:

```
Authorization: Bearer YOUR_ACCESS_TOKEN
Content-Type: application/json
```

10. Step 6 — Build the Request Body

Example request:

```
{
  "ContactKey": [
    "10001",
    "10002",
    "10003"
  ],
  "DefinitionKey": "YOUR_JOURNEY_DEFINITION_KEY",
  "Versions": [
    3
  ]
}
```

The three main properties are:

Property	Description
ContactKey	Array of contact identities that should exit
DefinitionKey	Journey definition identifier
Versions	Journey version(s) being targeted

11. Step 7 — Test the API in Postman

Postman is useful for validating the API before implementing it in an application or integration.

11.1 Generate the Access Token

1. Create a new POST request.
2. Use the authentication endpoint.
3. Set Content-Type to application/json.
4. Send the Client ID and Client Secret in the request body.
5. Copy the access_token from the response.

POST
https://YOUR_SUBDOMAIN.auth.marketingcloudapis.com/v2/token

Body:

```
{
  "grant_type": "client_credentials",
  "client_id": "YOUR_CLIENT_ID",
  "client_secret": "YOUR_CLIENT_SECRET"
}
```

11.2 Call the Contact Exit Endpoint

6. Create a second POST request.
7. Use the REST endpoint.
8. Add the Authorization header with the Bearer token.
9. Set Content-Type to application/json.
10. Add the ContactKey, DefinitionKey, and Versions in the request body.
11. Click Send.

POST
https://YOUR_SUBDOMAIN.rest.marketingcloudapis.com/interaction/v1/interactions/contactexit

Headers:

Authorization: Bearer YOUR_ACCESS_TOKEN

Content-Type: application/json

Body:

```
{
  "ContactKey": [
    "10001",
    "10002",
    "10003"
  ],
  "DefinitionKey": "YOUR_JOURNEY_DEFINITION_KEY",
  "Versions": [
    3
  ]
}
```

12. Example: Remove One Contact

If only one contact must be removed:

```
{
  "ContactKey": [
    "LEAD_10001"
  ],
  "DefinitionKey": "Lead_Nurturing_Journey",
}
```

```

    "Versions": [
      3
    ]
  }

```

Only the specified contact is targeted by the request. Other contacts in the active Journey are not included in this request.

13. Example: Remove Multiple Contacts

For a selected group:

```

{
  "ContactKey": [
    "LEAD_10001",
    "LEAD_10002",
    "LEAD_10003",
    "LEAD_10004"
  ],
  "DefinitionKey": "Lead_Nurturing_Journey",
  "Versions": [
    3
  ]
}

```

14. Removing a Large Number of Contacts

For a large contact population, do not design the integration around one oversized API request. The Contact Exit API has request limits, so contacts should be processed in batches. A common implementation pattern is to split the population into batches and make multiple requests.

```

1000 Contacts
|
+---- Batch 1
+---- Batch 2
+---- Batch 3
+---- ...
+---- Batch N
|
v
Contact Exit API

```

Always verify the current Salesforce API documentation and your tenant's applicable limits before selecting a batch size for production.

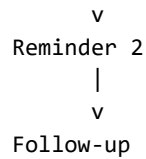
15. Real-World Use Case — Lead Nurturing

Imagine a Journey that sends:

```

Welcome Message
|
v
Reminder 1
|

```



A lead completes a site visit. The business rule says that the lead should no longer receive the remaining nurturing messages.

The integration can call:

```
{
  "ContactKey": [
    "LEAD_10001"
  ],
  "DefinitionKey": "Lead_Nurturing_Journey",
  "Versions": [
    3
  ]
}
```

The Journey continues for other contacts, while the selected lead is targeted for exit.

16. Real-World Use Case — Opt-Out

Suppose a customer opts out while they are already moving through a Journey. The business integration can identify the customer and request that the customer exit the Journey.

```
{
  "ContactKey": [
    "Customer_C"
  ],
  "DefinitionKey": "Marketing_Communication_Journey",
  "Versions": [
    3
  ]
}
```

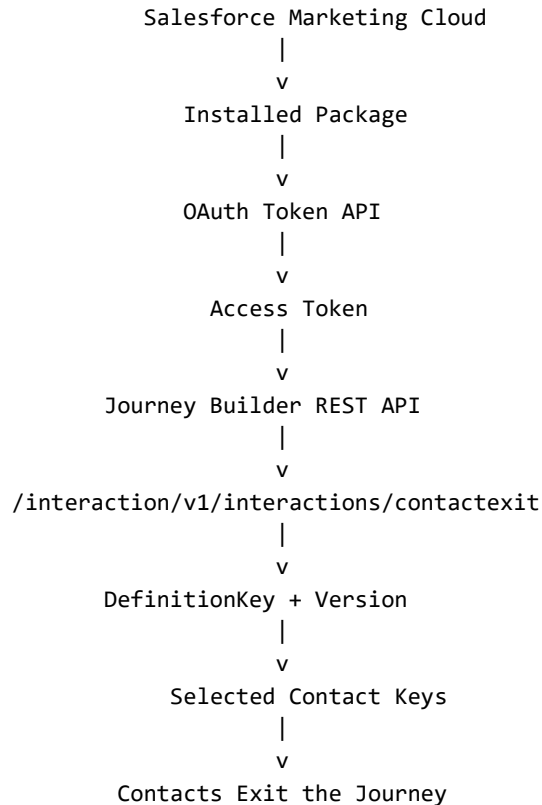
For actual consent and compliance implementations, the Journey exit should be part of a broader consent-management design; removing a contact from one Journey does not by itself constitute a universal unsubscribe.

17. Why Not Stop the Journey?

Stopping a Journey is a Journey-level operation. If the requirement is only to remove selected contacts, stopping the Journey is too broad.

Requirement	Recommended approach
Remove selected contacts	Contact Exit REST API
Remove a large selected population	Contact Exit REST API with batching
Stop all processing for the Journey	Journey-level stop operation
Conditionally prevent future contacts	Consider Journey entry/exit design and business rules

18. End-to-End REST API Architecture



19. Security Considerations

- Never hard-code Client Secret in public repositories.
- Do not expose access tokens in screenshots or blog posts.
- Use an appropriate Installed Package and minimum required API permissions.
- Store credentials in a secure secret-management mechanism in production.
- Do not log Client Secrets or full access tokens.
- Validate that the DefinitionKey and Journey version are correct before production execution.

20. Troubleshooting Checklist

Problem	What to check
401 Unauthorized	Access token, Client ID/Secret, authentication endpoint, token expiration
403 Forbidden	Installed Package/API permissions
400 Bad Request	JSON syntax, required properties, ContactKey format, DefinitionKey, Version
Contacts do not exit	Confirm ContactKey matches Journey contact identity and the targeted Journey version
Wrong Journey targeted	Verify DefinitionKey and version
Large request fails	Reduce batch size and follow current API limits

21. Complete cURL Example

```
curl --request POST \
  --url
  "https://YOUR_SUBDOMAIN.rest.marketingcloudapis.com/interaction/v1/interactions/contactexit" \
  --header "Authorization: Bearer YOUR_ACCESS_TOKEN" \
  --header "Content-Type: application/json" \
  --data '{
    "ContactKey": [
      "10001",
      "10002",
      "10003"
    ],
    "DefinitionKey": "YOUR_JOURNEY_DEFINITION_KEY",
    "Versions": [
      3
    ]
  }'
```

22. End-to-End Summary

12. Create or use an Installed Package with the required API permissions.
13. Collect the Client ID and Client Secret securely.
14. Call the OAuth token endpoint.
15. Retrieve the access token.
16. Identify the correct Journey DefinitionKey.
17. Identify the correct Journey version.
18. Collect the ContactKeys that should exit.
19. Call the Contact Exit endpoint.
20. Process large populations in batches.
21. Validate the result and monitor errors.

23. Key Takeaway

When a Salesforce Marketing Cloud Journey is active and only a selected set of contacts needs to be removed, the Journey Builder Contact Exit REST API provides a programmatic solution.

POST /interaction/v1/interactions/contactexit

The important distinction is that the API targets specified ContactKeys rather than requiring the entire Journey to be stopped. This makes it useful for operational scenarios such as lead status changes, completed business processes, eligibility changes, and other controlled contact-exit requirements.

24. Final Reference

Item	Value / Pattern
Authentication	OAuth 2.0 Client Credentials
Token Endpoint	https://YOUR_SUBDOMAIN.auth.marketingcloudapis.com/v2/token
Contact Exit Method	POST
Contact Exit Endpoint	/interaction/v1/interactions/contactexit
Authorization	Bearer access token
Required identity	Journey ContactKey

Journey identifier	DefinitionKey
Journey version	Versions

25. Official Salesforce References

Use the following official Salesforce documentation as the primary reference when implementing or validating this solution.

- [Salesforce Developers — Remove Contact from a Journey](#) — Primary reference for POST `/interaction/v1/interactions/contactexit`, `ContactKey`, `DefinitionKey`, `Versions`, and the 50-contact request limit.
- [Salesforce Developers — Contact Exit Request Status](#) — Reference for checking the status of a contact-exit request.
- [Salesforce Developers — REST API Permission IDs and Scopes](#) — Reference for REST API permission and scope information.
- [Salesforce Help — Journey Health](#) — Reference for Journey Health and Contact Path behavior.
- [Salesforce Help — View a Contact Path in a Journey](#) — Reference for viewing a contact path and manually removing a contact.

Important: The Contact Exit API documentation confirms that `ContactKey` can contain a single contact or an array of up to 50 contacts. For larger populations, process contacts in batches and verify the current Salesforce limits before production use.

26. Official API Reference for Verifying Exit Status

After submitting a contact-exit request, Salesforce also provides an endpoint for retrieving the status of the request.

Endpoint: POST `/interaction/v1/interactions/contactexit/status`

Official documentation: [Salesforce Developers — Contact Exit Request Status](#)

This can be useful when implementing an automated integration where you need to confirm whether each `ContactKey` was successfully removed from the targeted Journey version.