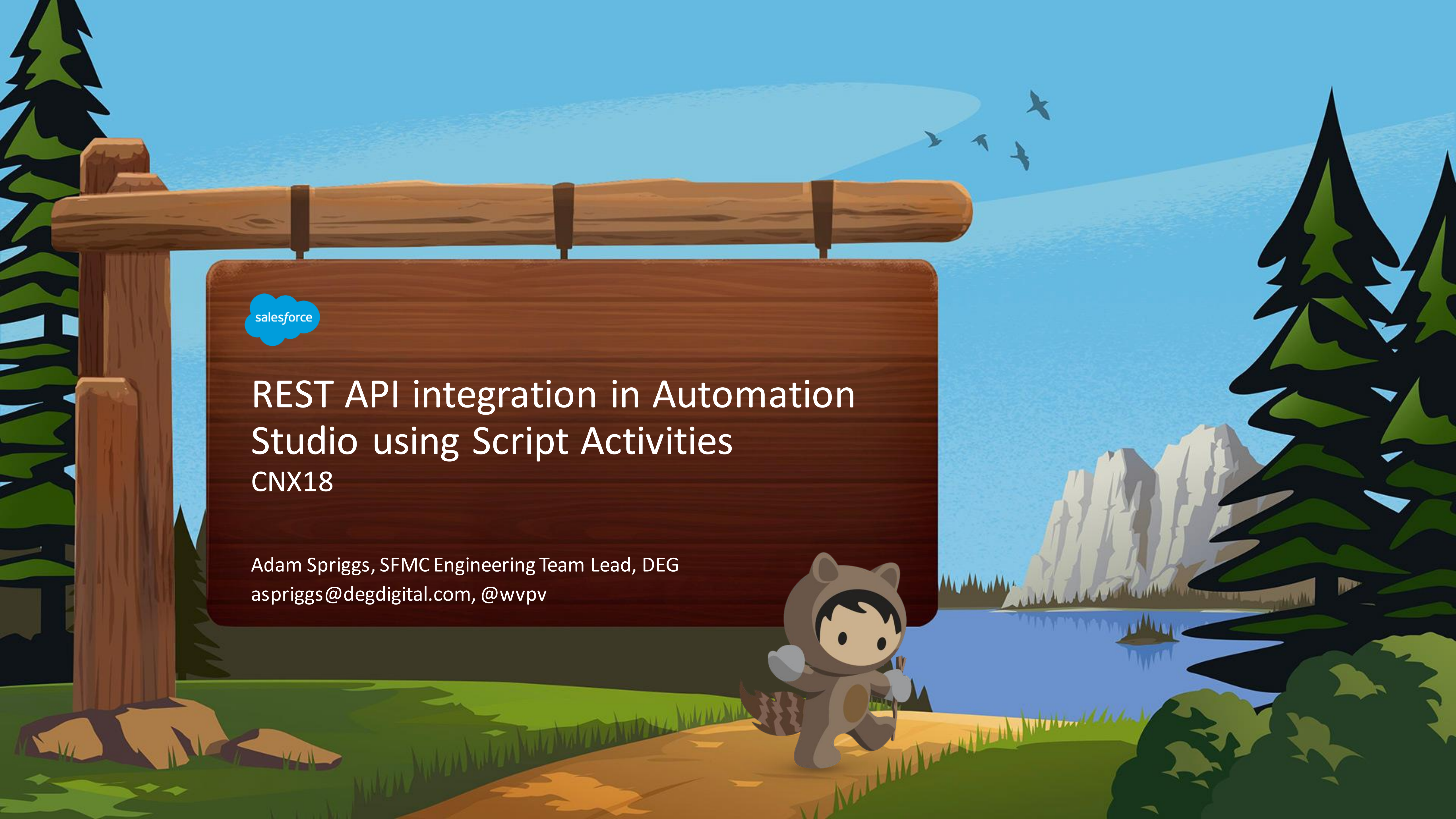


REST API integration in Automation Studio using Script Activities

CNX18

Adam Spriggs, SFMC Engineering Team Lead, DEG
aspriggs@degdigital.com, @wvpv

Introduction

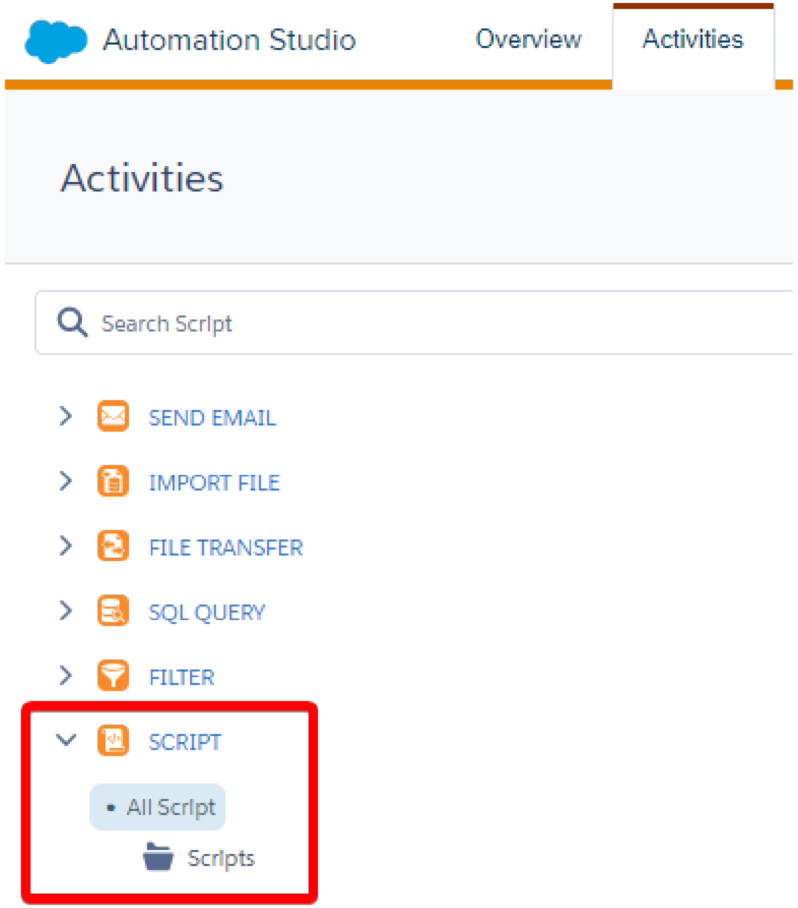
Script Activities are one of my favorite parts of Marketing Cloud. They open the door for all kinds of automated interactions and integrations with the platform. Not only can you leverage the built-in Server-Side JavaScript libraries, but it works nicely with the Marketing Cloud REST API for functions that aren't possible outside of the API.

Agenda

- Provisioning
- REST API Setup
- Contact Delete

Provisioning

- Script Activities have to be enabled by Marketing Cloud Support.
- Once enabled, they'll show as an Activity Type in Automation Studio
- Server-Side JavaScript in general, does not need to be provisioned. You can it in CloudPages and -- with caution -- in emails.



REST API Setup

- Log into Marketing Cloud
- Hover over your user name in the upper right corner
- Click Administration
- In the Account drop-down, click Installed Packages
- Click the New button in the upper right corner
- Give your package a name and click Save
- Click the Add Component button at the bottom
- Choose API Integration
- Click Next
- Check the permissions appropriate for your use-case.
- Click Save
- The Client ID and Client Secret are now configured and ready to use.

Endpints

Marketing Cloud REST API endpoint:

https://www.exacttargetapis.com/

For requesting a token:

https://auth.exacttargetapis.com

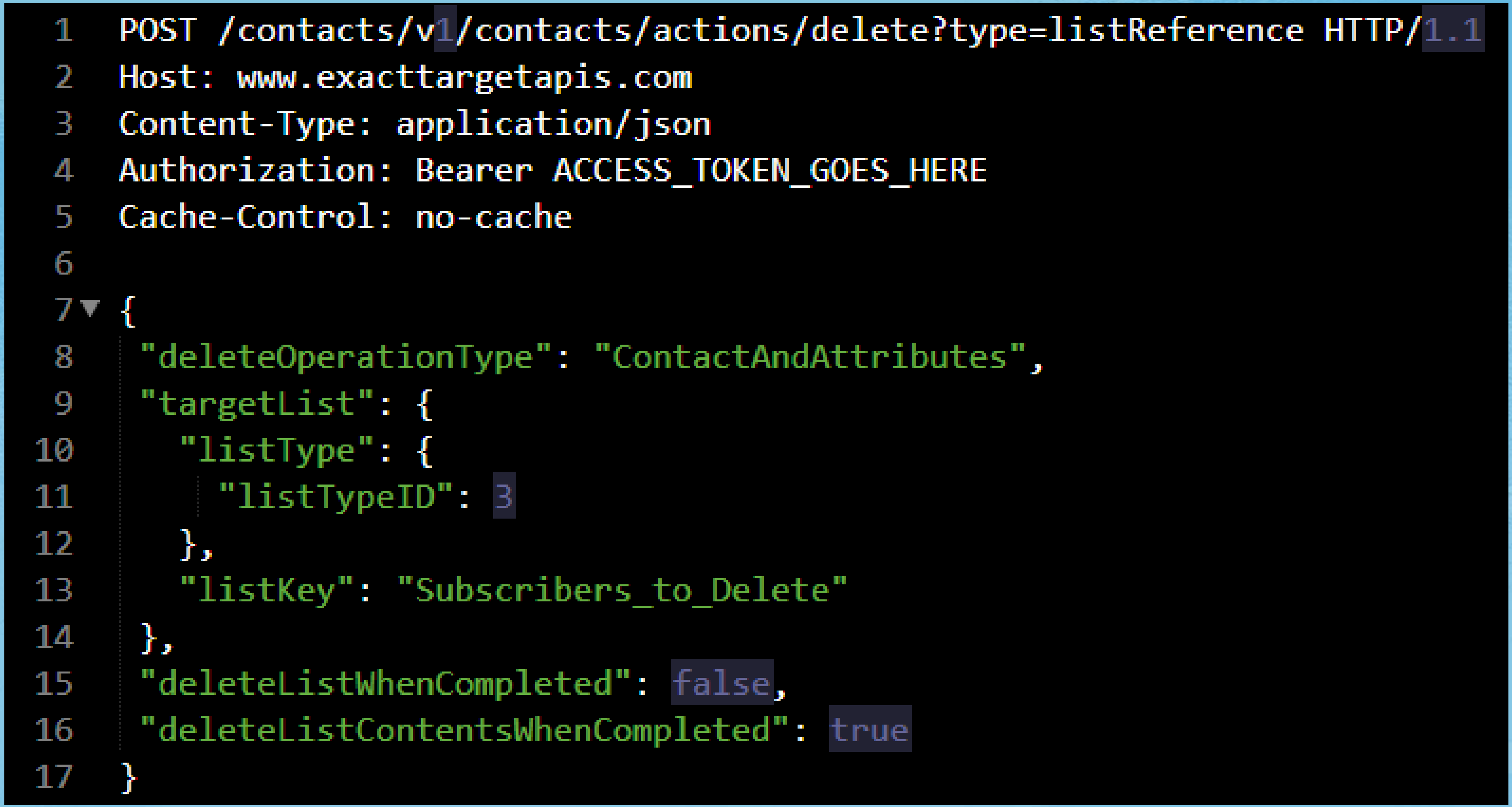
REST API Testing & Debugging

- Use Postman to write your initial API calls
- Paste your script into a CloudPage, load it and verify the results

Contact Delete

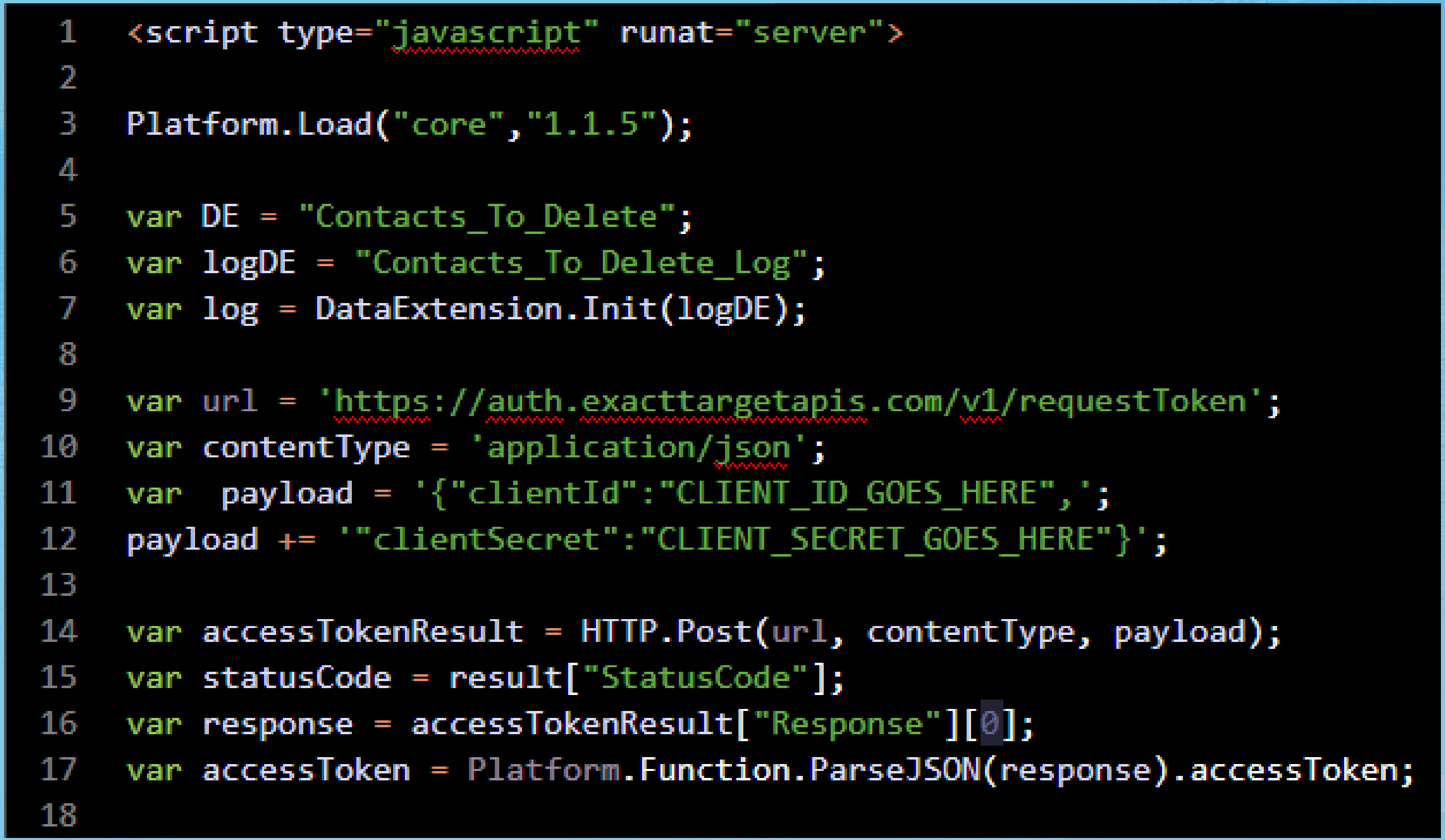
- Must be provisioned by Marketing Cloud Support and then enabled in Contact Builder
- Only for use in the parent business unit
- Deletion also be initiated manually by list, sendable data extension or individually
- Does not remove tracking data, just anonomizes it.
- Suppression period defaults to 14 days, but can be set to zero days.
- Non-sendable data extensions are excluded from this deletion process.
- Need to consider the possibility of re-introduction of contacts from external sources and integrations.

REST API Call to Delete



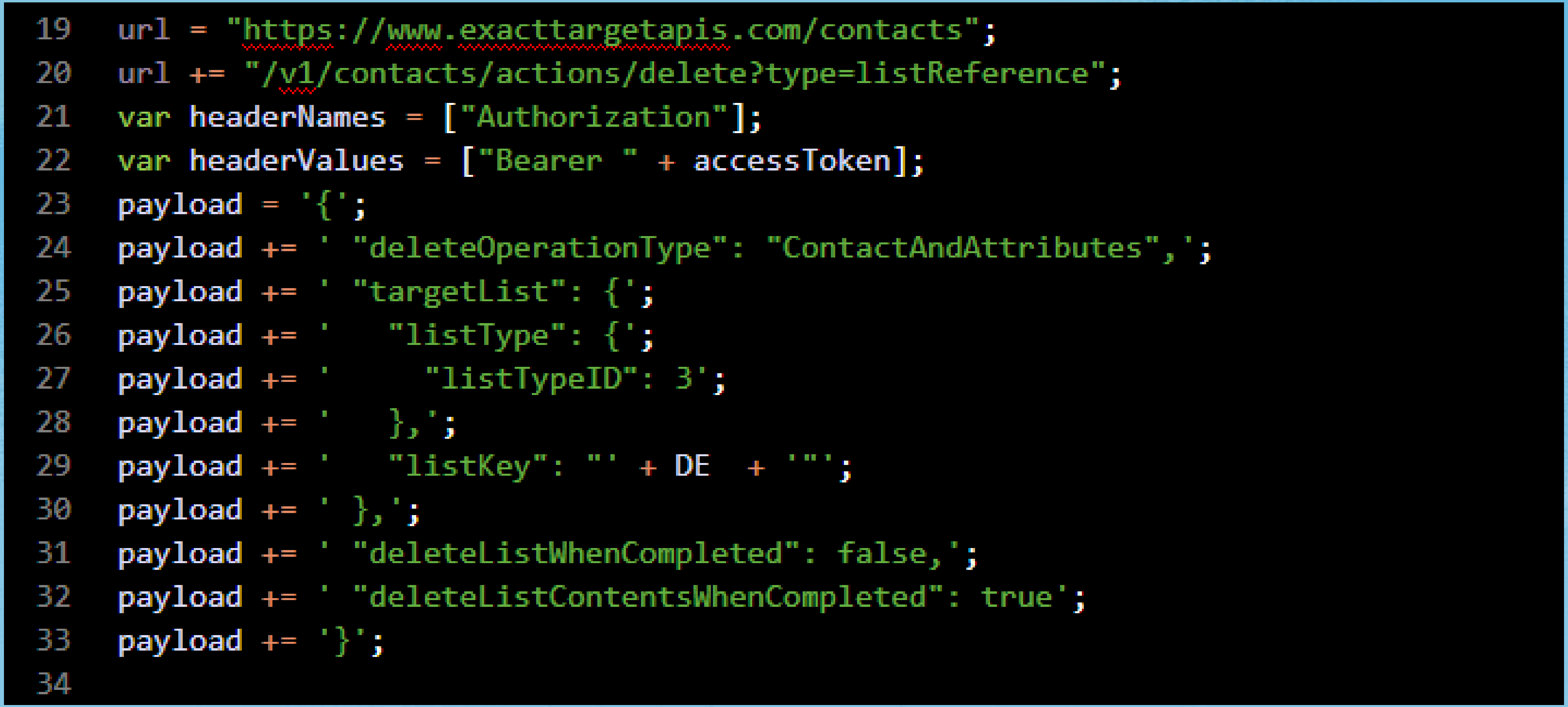
```
POST /contacts/v1/contacts/actions/delete?type=listReference HTTP/1.1
Host: www.exacttargetapis.com
Content-Type: application/json
Authorization: Bearer ACCESS_TOKEN_GOES_HERE
Cache-Control: no-cache

{
 "deleteOperationType": "ContactAndAttributes",
 "targetList": {
   "listType": {
     "listTypeID": 3
   },
   "listKey": "Subscribers_to_Delete"
 },
 "deleteListWhenCompleted": false,
 "deleteListContentsWhenCompleted": true
}
```

SSJS Contact Delete – Part 1



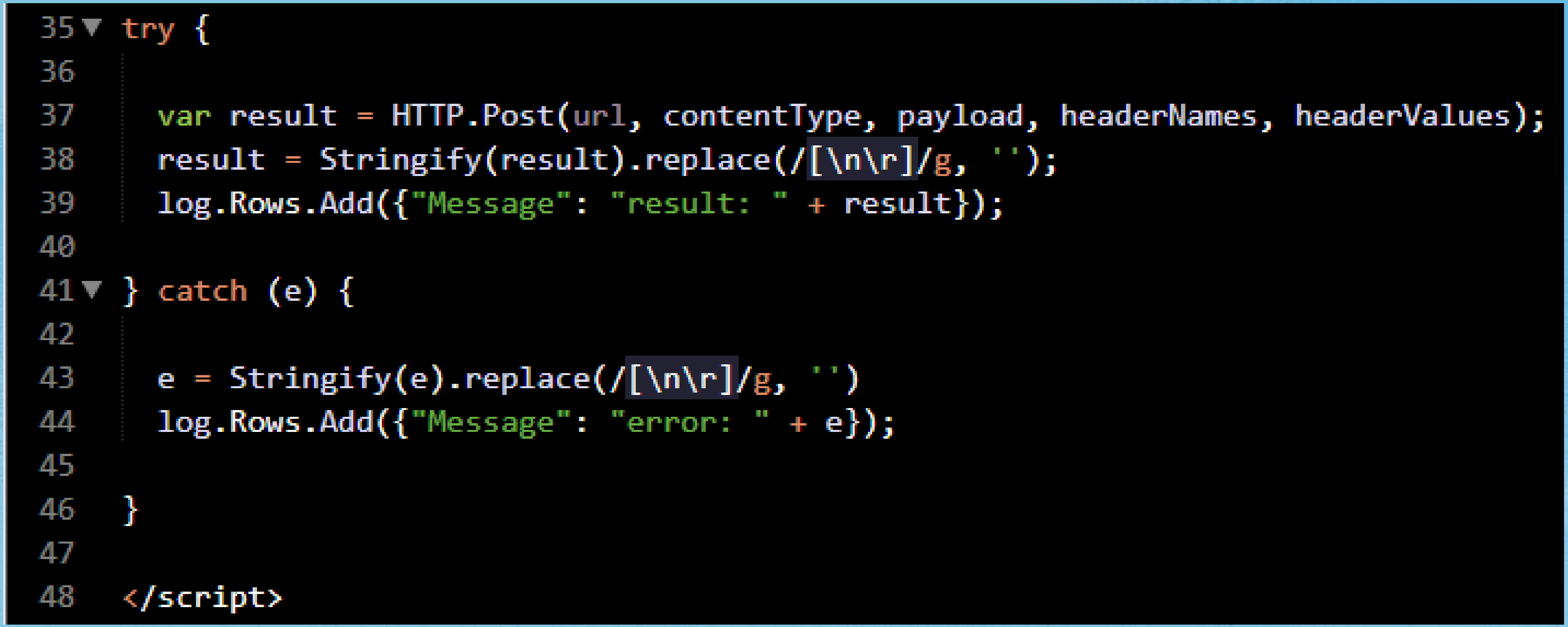
```
<script type="javascript" runat="server">

Platform.Load("core","1.1.5");

var DE = "Contacts_To_Delete";
var logDE = "Contacts_To_Delete_Log";
var log = DataExtension.Init(logDE);

var url = 'https://auth.exacttargetapis.com/v1/requestToken';
var contentType = 'application/json';
var  payload = '{"clientId":"CLIENT_ID_GOES_HERE",';
payload += '"clientSecret":"CLIENT_SECRET_GOES_HERE"}';

var accessTokenResult = HTTP.Post(url, contentType, payload);
var statusCode = result["StatusCode"];
var response = accessTokenResult["Response"][0];
var accessToken = Platform.Function.ParseJSON(response).accessToken;

```

SSJS Contact Delete – Part 2



```
url = "https://www.exacttargetapis.com/contacts";
url += "/v1/contacts/actions/delete?type=listReference";
var headerNames = ["Authorization"];
var headerValues = ["Bearer " + accessToken];
payload = '{';
payload += ' "deleteOperationType": "ContactAndAttributes",';
payload += ' "targetList": {';
payload += '   "listType": {';
payload += '     "listTypeID": 3';
payload += '   },';
payload += '   "listKey": "' + DE  + '"';
payload += ' },';
payload += ' "deleteListWhenCompleted": false,';
payload += ' "deleteListContentsWhenCompleted": true';
payload += '}';

```

SSJS Contact Delete – Part 3



```
try {

  var result = HTTP.Post(url, contentType, payload, headerNames, headerValues);
  result = Stringify(result).replace(/[\n\r]/g, '');
  log.Rows.Add({"Message": "result: " + result});

} catch (e) {

  e = Stringify(e).replace(/[\n\r]/g, '')
  log.Rows.Add({"Message": "error: " + e});

}

</script>
```

Questions?

Links

- This Slide Deck - http://bit.ly/CNX18-RASA-PDF
- Google Doc - http://bit.ly/CNX18-RASA

THANK YOU