

Link Checker

/**

- Configuration to be used for the Link Checker.

*/

CONFIG = {

// URL of the spreadsheet template.

// This should be a copy of

<https://docs.google.com/spreadsheets/d/1iO1iEGwIbe510qo3Li-j4KgyCeVSmodxU6J7M756ppk/copy>.

'spreadsheet_url':

<https://docs.google.com/spreadsheets/d/1Tv-PVLRawUetqmkAN2s23JQMHGfvzICtOH2BN0GUokE/edit?usp=sharing>,

// Array of addresses to be alerted via email if issues are found.

'recipient_emails': [

'youremail'

],

// Label to use when a link has been checked. Label will be created if it doesn't exist.

'label': 'LinkChecker_Done',

// Number of seconds to sleep after each URL request. If your URLs are

// all on one or a few domains, use this throttle to reduce the load that the

// script imposes on your web server(s).

'throttle': 0,

// Number of seconds before timeout that the script should stop checking URLs

// to make sure it has time to output its findings.

```
'timeout_buffer': 120,
```

```
'advanced_options': {
```

```
/**
```

- Parameters controlling the script's behavior after hitting a UrlFetchApp
- QPS quota limit.

```
*/
```

```
'quota_config': {
```

```
'INIT_SLEEP_TIME': 250,
```

```
'BACKOFF_FACTOR': 2,
```

```
'MAX_TRIES': 5
```

```
}
```

```
}
```

```
};
```

Negative Keyword Checker

```
function main() {
```

```
var campaigns = AdsApp.campaigns().get();
```

```
while (campaigns.hasNext()) {
```

```
var campaign = campaigns.next();
```

```
var adGroups = campaign.adGroups().get();
```

```
while (adGroups.hasNext()) {
```

```
var adGroup = adGroups.next();
```

```
var keywords = adGroup.keywords().withCondition("Status = ENABLED").get();
```

```

var negativeKeywords = adGroup.negativeKeywords().withCondition("Status =
ENABLED").get();

var negativeKeywordTexts = [];

while (negativeKeywords.hasNext()) {

var negativeKeyword = negativeKeywords.next();

negativeKeywordTexts.push(negativeKeyword.getText());

}

while (keywords.hasNext()) {

var keyword = keywords.next();

var keywordText = keyword.getText();

for (var i = 0; i < negativeKeywordTexts.length; i++) {

var negativeKeywordText = negativeKeywordTexts[i];

if (negativeKeywordText == keywordText) {

var newNegativeKeyword = adGroup.createNegativeKeyword(keywordText);

Logger.log("Conflict found: " + keywordText + " added to negative keywords");

}

}

}

}

}

}

}

```

Bug in Ads

```

function main() {

var emailAddress = 'youremail'; // Your email address

```

```

var timeFrame = 1; // Check for changes in the last 1 day

var accountInfo = getAccountInfo();

var campaignChanges = getRecentCampaignChanges(timeFrame);

if (campaignChanges.length > 0) {

    sendEmail(accountInfo, campaignChanges, emailAddress, timeFrame);

} else {

    Logger.log('No changes detected in the last 24 hours. No email sent.');
```

```

}
```

```

}
```

```

function getAccountInfo() {
```

```

    var account = AdsApp.currentAccount();
```

```

    return {
```

```

        name: account.getName(),
```

```

        customerId: account.getCustomerId()
```

```

    };
```

```

}
```

```

function getRecentCampaignChanges(days) {
```

```

    var changes = [];
```

```

    var today = formatDate(new Date());
```

```

    var yesterday = formatDate(new Date(new Date().getTime() - 24 * 60 * 60 * 1000));
```

```

    var campaignIterator = AdsApp.campaigns()
```

```

        .forDateRange('TODAY')
```

```

        .get();
```

```

    while (campaignIterator.hasNext()) {
```

```

var campaign = campaignIterator.next();

var todayStats = campaign.getStatsFor('TODAY');

var statusChanged = hasStatusChanged(campaign, yesterday);

if (todayStats.getImpressions() > 0 || todayStats.getClicks() > 0 || todayStats.getCost() >
0 || statusChanged) {

changes.push({

'name': campaign.getName(),

'status': campaign.isEnabled() ? 'Enabled' : 'Paused',

'statusChanged': statusChanged,

'impressions': todayStats.getImpressions(),

'clicks': todayStats.getClicks(),

'cost': todayStats.getCost()

});

}

}

return changes;

}

function hasStatusChanged(campaign, yesterday) {

var currentStatus = campaign.isEnabled() ? 'Enabled' : 'Paused';

var yesterdayStats = campaign.getStatsFor(yesterday, yesterday);

var hadActivityYesterday = yesterdayStats.getImpressions() > 0 ||
yesterdayStats.getClicks() > 0 || yesterdayStats.getCost() > 0;

// If the campaign is currently enabled but had no activity yesterday, or

// if it's currently paused but had activity yesterday, we consider it changed

```

```

return (currentStatus === 'Enabled' && !hadActivityYesterday) || (currentStatus ===
'Paused' && hadActivityYesterday);

}

function sendEmail(accountInfo, changes, emailAddress, timeFrame) {

var accountUrl = 'https://ads.google.com/aw/overview?ocid=' +
accountInfo.customerId.replace('-', '');

var subject = 'Alert: Changes detected in Google Ads account "' + accountInfo.name +
'''';

var body = 'The following changes were detected in the last ' + timeFrame + '
day(s):\n\n';

body += 'Account: ' + accountInfo.name + ' (Customer ID: ' + accountInfo.customerId +
')\n';

body += 'Account Link: ' + accountUrl + '\n\n';

body += 'Campaign Changes:\n\n';

for (var i = 0; i < changes.length; i++) {

var change = changes[i];

body += 'Campaign: ' + change.name + '\n';

body += 'Status: ' + change.status + (change.statusChanged ? ' (Changed)' : '') + '\n';

body += 'Impressions: ' + change.impressions + '\n';

body += 'Clicks: ' + change.clicks + '\n';

body += 'Cost: ' + change.cost + '\n\n';

}

MailApp.sendEmail(emailAddress, subject, body);

}

function formatDate(date) {

var year = date.getFullYear();

```

```
var month = ('0' + (date.getMonth() + 1)).slice(-2);  
var day = ('0' + date.getDate()).slice(-2);  
return year + month + day;  
}
```

For the full resources guide [join our newsletter](#) and send us a message at will@forgeiq.ca.