

リダイレクトURI取得

```
function getOAuth2RedirectUri() {
    var service = getService();
    Logger.log(service.getRedirectUri());
}

function getService() {
    return OAuth2.createService('YouTube')
        .setAuthorizationBaseUrl('https://accounts.google.com/o/oauth2/auth')
        .setTokenUrl('https://accounts.google.com/o/oauth2/token')
        .setClientId('YOUR_CLIENT_ID') // 後で実際のクライアントIDに置き換える
        .setClientSecret('YOUR_CLIENT_SECRET') // 後で実際のクライアントシークレットに置き換える
        .setCallbackFunction('authCallback')
        .setPropertyStore(PropertiesService.getUserProperties())
        .setScope('https://www.googleapis.com/auth/yt-analytics.readonly')
        .setParam('access_type', 'offline');
}

function authCallback(request) {
    var service = getService();
    var isAuthorized = service.handleCallback(request);
    if (isAuthorized) {
        return HtmlService.createHtmlOutput('認証に成功しました。このタブを閉じてください。');
    } else {
        return HtmlService.createHtmlOutput('認証に失敗しました。');
    }
}
```

データ取得・書き込み

```
function getService() {
  return OAuth2.createService('YouTube')
    .setAuthorizationBaseUrl('https://accounts.google.com/o/oauth2/auth')
    .setTokenUrl('https://accounts.google.com/o/oauth2/token')
    .setClientId('YOUR_CLIENT_ID') // ここに実際のクライアントIDを貼り付け
    .setClientSecret('YOUR_CLIENT_SECRET') // ここに実際のクライアントシークレットを
    貼り付け
    .setCallbackFunction('authCallback')
    .setPropertyStore(PropertiesService.getUserProperties())
    .setScope('https://www.googleapis.com/auth/yt-analytics.readonly')
    .setParam('access_type', 'offline');
}
```

```
function authCallback(request) {
  var service = getService();
  var isAuthorized = service.handleCallback(request);
  if (isAuthorized) {
    return HtmlService.createHtmlOutput('認証に成功しました。このタブを閉じてください。');
  } else {
    return HtmlService.createHtmlOutput('認証に失敗しました。');
  }
}
```

// YouTube Analytics API からデータを取得し、スプレッドシートに書き込む関数

```
function getYouTubeAnalyticsData() {
  var service = getService();
  if (service.hasAccess()) {
    var url = 'https://youtubeanalytics.googleapis.com/v2/reports';
    var params = {
      'ids': 'channel==MINE', // 自分のチャンネル
      'startDate': '2025-01-01', // 取得開始日
      'endDate': '2025-03-23', // 取得終了日
      'metrics': 'views,likes,comments', // 取得する指標
      'dimensions': 'day', // 日別のデータ
      'sort': 'day' // 日付順にソート
    };

    var queryString = Object.keys(params).map(function(key) {
      return key + '=' + encodeURIComponent(params[key]);
    }).join('&');

    var options = {
      'method': 'get',
      'headers': {
        'Authorization': 'Bearer ' + service.getAccessToken()
      },
      'muteHttpExceptions': true
    };
  }
}
```

```
};

var response = UrlFetchApp.fetch(url + '?' + queryString, options);
var result = JSON.parse(response.getContentText());

// スプレッドシートに書き込む処理 (例)
var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
var sheet = spreadsheet.getSheetByName('シート1'); // シート名を指定

// 既存のデータをクリア
sheet.clearContents();

// ヘッダー行を書き込み
sheet.appendRow(result.columnHeaders.map(function(header) { return
header.name; }));

// データ行を書き込み
result.rows.forEach(function(row) {
  sheet.appendRow(row);
});

} else {
  var authorizationUrl = service.getAuthorizationUrl();
  Logger.log('以下のURLを開いて認証してください: %s', authorizationUrl);
}
}
```

分析用データ取得

```

// OAuth2 ライブラリの設定 (YouTube Analytics API v2 と YouTube Data API v3 用)
function getService() {
    return OAuth2.createService('YouTube')
        .setAuthorizationBaseUrl('https://accounts.google.com/o/oauth2/auth')
        .setTokenUrl('https://accounts.google.com/o/oauth2/token')
        .setClientId('YOUR_CLIENT_ID') // クライアントID
        .setClientSecret('YOUR_CLIENT_SECRET') // クライアントシークレット
        .setCallbackFunction('authCallback')
        .setPropertyStore(PropertiesService.getUserProperties())
        // YouTube Analytics API と Data API の両方のスコープを設定
        .setScope([
            'https://www.googleapis.com/auth/yt-analytics.readonly',
            'https://www.googleapis.com/auth/youtube.readonly'
        ])
        .setParam('access_type', 'offline');
}

function authCallback(request) {
    var service = getService();
    var isAuthorized = service.handleCallback(request);
    if (isAuthorized) {
        return HtmlService.createHtmlOutput('認証に成功しました。このタブを閉じてください。');
    } else {
        return HtmlService.createHtmlOutput('認証に失敗しました。');
    }
}

// メイン関数: YouTube Analytics API と Data API からデータを取得し、スプレッドシートに書き込む
function getYouTubeData() {
    var service = getService();
    if (service.hasAccess()) {
        var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
        var today = new Date();
        var formattedDate = Utilities.formatDate(today, 'Asia/Tokyo', 'yyyyMMdd');
        var startDate = new Date(today.getFullYear(), today.getMonth(), today.getDate() - 27);
        var formattedStartDate = Utilities.formatDate(startDate, 'UTC', 'yyyy-MM-dd');
        var formattedEndDate = Utilities.formatDate(today, 'UTC', 'yyyy-MM-dd');

        // 各関数を呼び出してデータを取得・書き込み (エラーが発生しても続行)
        try {
            fetchAndWriteOverviewData(service, spreadsheet, formattedStartDate, formattedEndDate, formattedDate);
        } catch (e) {
            handleError(spreadsheet, '概要データ', formattedDate, e);
        }
    }
}

```

```
    try {
        fetchAndWriteVideoData(service, spreadsheet, formattedStartDate,
formattedEndDate, formattedDate);
    } catch (e) {
        handleError(spreadsheet, '動画別データ', formattedDate, e);
    }

    try {
        fetchAndWriteDemographicsData(service, spreadsheet, formattedStartDate,
formattedEndDate, formattedDate);
    } catch (e) {
        handleError(spreadsheet, '視聴者層データ', formattedDate, e);
    }

    try {
        fetchAndWriteTrafficSourceData(service, spreadsheet, formattedStartDate,
formattedEndDate, formattedDate);
    } catch (e) {
        handleError(spreadsheet, 'トラフィックソース', formattedDate, e);
    }

    try {
        fetchAndWritePlaybackLocationData(service, spreadsheet, formattedStartDate,
formattedEndDate, formattedDate);
    } catch (e) {
        handleError(spreadsheet, '再生場所', formattedDate, e);
    }

    try {
        fetchAndWriteDeviceTypeData(service, spreadsheet, formattedStartDate,
formattedEndDate, formattedDate);
    } catch (e) {
        handleError(spreadsheet, 'デバイスタイプ別', formattedDate, e);
    }

    try {
        fetchAndWriteOperatingSystemData(service, spreadsheet, formattedStartDate,
formattedEndDate, formattedDate);
    } catch (e) {
        handleError(spreadsheet, 'OS別', formattedDate, e);
    }

    try {
        fetchAndWriteCountryData(service, spreadsheet, formattedStartDate,
formattedEndDate, formattedDate);
    } catch (e) {
        handleError(spreadsheet, '国別', formattedDate, e);
    }
```



```

    }

    // 視聴者維持率関連のコードを削除

    } else {
        var authorizationUrl = service.getAuthorizationUrl();
        Logger.log('以下のURLを開いて認証してください: %s', authorizationUrl);
    }
}

// エラーハンドリング用関数
function handleError(spreadsheet, dataType, formattedDate, error) {
    Logger.log('%s 取得エラー: %s', dataType, error);
    var errorSheetName = dataType + '_エラー_' + formattedDate;
    // エラーシートが既に存在するか確認
    var errorSheet = spreadsheet.getSheetByName(errorSheetName);
    if (!errorSheet) {
        errorSheet = spreadsheet.insertSheet(errorSheetName);
    }
    errorSheet.appendRow([dataType + ' 取得エラー', error.toString()]);
}

// 1. 概要データ
function fetchAndWriteOverviewData(service, spreadsheet, startDate, endDate,
formattedDate) {
    fetchAndWriteAnalyticsData(service, spreadsheet, '概要データ_' + formattedDate,
startDate, endDate,

'views,estimatedMinutesWatched,averageViewDuration,averageViewPercentage,likes,dislike
s,comments,shares,subscribersGained,subscribersLost',
    'day',
    'day');
}

// 2. 動画別データ + YouTube Data API v3 で動画情報を取得
function fetchAndWriteVideoData(service, spreadsheet, startDate, endDate, formattedDate) {
    var analyticsData = fetchAnalyticsData(service, startDate, endDate,

'views,estimatedMinutesWatched,averageViewDuration,averageViewPercentage,likes,dislike
s,comments,shares,subscribersGained,subscribersLost',
    'video',
    '-views',
    50); // 上位50件

    if (!analyticsData || !analyticsData.rows) {
        Logger.log('動画別データが見つかりませんでした。');
        return;
    }
}

```

```

var sheet = spreadsheet.insertSheet('動画別データ_' + formattedDate);
var headers = analyticsData.columnHeaders.map(function(header) { return header.name;
});
headers.push('title', 'description', 'tags'); // Data API から取得する項目
sheet.appendRow(headers);

// 各動画のデータを取得・書き込み
analyticsData.rows.forEach(function(row) {
    var videoid = row[0]; // 動画 ID
    var videoInfo = fetchVideoInfo(service, videoid); // YouTube Data API から情報を取得
    if (videoInfo) {
        var videoData = row.concat([
            videoInfo.title,
            videoInfo.description,
            videoInfo.tags ? videoInfo.tags.join(',') : '' // タグはカンマ区切りで結合
        ]);
        sheet.appendRow(videoData);
    } else {
        // 動画情報が取得できなかった場合は、動画IDとエラーメッセージを記録
        var errorData = row.concat(['動画情報取得エラー', '', '']);
        sheet.appendRow(errorData);
    }
});
}

// YouTube Data API v3 を使って動画情報を取得する関数
function fetchVideoInfo(service, videoid) {
    var url = 'https://www.googleapis.com/youtube/v3/videos';
    var params = {
        'part': 'snippet',
        'id': videoid,
        'key': 'YOUR_API_KEY' // ここにAPIキー
    };
    var queryString = Object.keys(params).map(function(key) {
        return key + '=' + encodeURIComponent(params[key]);
    }).join('&');
    var options = {
        'method': 'get',
        'headers': {
            'Authorization': 'Bearer ' + service.getAccessToken()
        },
        'muteHttpExceptions': true
    };
    var response = UrlFetchApp.fetch(url + '?' + queryString, options);
    var result = JSON.parse(response.getContentText());

    if (response.getResponseCode() !== 200 || !result.items || result.items.length === 0) {

```

```

        Logger.log('Error fetching video info for %s: %s', videoid, response.getContentText());
        return null;
    }

    return result.items[0].snippet; // snippet 部分を返す
}

// 3. 視聴者層
function fetchAndWriteDemographicsData(service, spreadsheet, startDate, endDate,
formattedDate) {
    fetchAndWriteAnalyticsData(service, spreadsheet, '年齢層別_' + formattedDate, startDate,
endDate,
        'viewerPercentage',
        'ageGroup,gender',
        'ageGroup,gender');
}

// 4. トラフィックソース
function fetchAndWriteTrafficSourceData(service, spreadsheet, startDate, endDate,
formattedDate) {
    fetchAndWriteAnalyticsData(service, spreadsheet, 'トラフィックソースタイプ別_' +
formattedDate, startDate, endDate,
        'views,estimatedMinutesWatched',
        'insightTrafficSourceType',
        'insightTrafficSourceType'
    );
}

// 5. 再生場所
function fetchAndWritePlaybackLocationData(service, spreadsheet, startDate, endDate,
formattedDate){
    fetchAndWriteAnalyticsData(service, spreadsheet, '再生場所詳細_' + formattedDate,
startDate, endDate,
        'views,estimatedMinutesWatched',
        'insightPlaybackLocationType',
        'insightPlaybackLocationType',
        25
    );
}

// 6. デバイスタイプ
function fetchAndWriteDeviceTypeData(service, spreadsheet, startDate, endDate,
formattedDate) {
    fetchAndWriteAnalyticsData(service, spreadsheet, 'デバイスタイプ別_' + formattedDate,
startDate, endDate,
        'views,estimatedMinutesWatched',
        'deviceType',
        'deviceType');
}

```

```
}
```

```
// 7. OS
```

```
function fetchAndWriteOperatingSystemData(service, spreadsheet, startDate, endDate,
formattedDate) {
    fetchAndWriteAnalyticsData(service, spreadsheet, 'OS別_' + formattedDate, startDate,
endDate,
    'views,estimatedMinutesWatched',
    'operatingSystem',
    'operatingSystem');
}
```

```
// 8. 国別
```

```
function fetchAndWriteCountryData(service, spreadsheet, startDate, endDate,
formattedDate) {
    fetchAndWriteAnalyticsData(service, spreadsheet, '国別_' + formattedDate, startDate,
endDate,

'views,estimatedMinutesWatched,averageViewDuration,averageViewPercentage,likes,dislike
s,comments,shares,subscribersGained,subscribersLost',
    'country',
    '-views',
    250);
}
```

```
// 視聴者維持率関連の関数を削除
```

```
// YouTube Analytics API からデータを取得する共通関数
```

```
function fetchAnalyticsData(service, startDate, endDate, metrics, dimensions, sort,
maxResults, extraParams) {
    var url = 'https://youtubeanalytics.googleapis.com/v2/reports';
    var params = {
        'ids': 'channel==MINE',
        'startDate': startDate,
        'endDate': endDate,
        'metrics': metrics,
        'dimensions': dimensions,
        'sort': sort,
        'maxResults': maxResults,
        'filters': extraParams ? extraParams.filters : undefined // 存在する場合のみ
    };
};
```

```
    // undefined, null, 空文字列 の値を除外し、クエリ文字列を作成
var queryString = Object.keys(params)
    .filter(function(key) {
        var value = params[key];
        return value !== undefined && value !== null && value !== "";
    })
```

```

        .map(function(key) {
            return key + '=' + encodeURIComponent(params[key]);
        })
        .join('&');

var options = {
    'method': 'get',
    'headers': {
        'Authorization': 'Bearer ' + service.getAccessToken()
    },
    'muteHttpExceptions': true
};

var response = UrlFetchApp.fetch(url + '?' + queryString, options);
var result = JSON.parse(response.getContentText());

if (!result.columnHeaders) {
    Logger.log('fetchAnalyticsData Error: %s', response.getContentText()); // より詳細なログ
    return null;
}

return result;
}

// YouTube Analytics API のデータをスプレッドシートに書き込む共通関数
function fetchAndWriteAnalyticsData(service, spreadsheet, sheetName, startDate, endDate,
metrics, dimensions, sort, maxResults, extraParams) {
    var result = fetchAnalyticsData(service, startDate, endDate, metrics, dimensions, sort,
maxResults, extraParams);

    if (result) {
        // シートが既に存在するか確認
        var sheet = spreadsheet.getSheetByName(sheetName);
        if (!sheet) {
            sheet = spreadsheet.insertSheet(sheetName);
            var headers = result.columnHeaders.map(function (header) { return header.name; });
            sheet.appendRow(headers);
        }
        result.rows.forEach(function(row) {
            sheet.appendRow(row);
        });
    }
}

```