Newman Run Report

仕様

動作

実行すると `newman` フォルダが作成され、結果が保存されていきます
`newman` ディレクトリにはhtml、json形式の結果

Newman Report

Collection	Potato_Run
Time	Fri Feb 01 2019 15:44:34 GMT+0900 (GMT+09:00)
Exported with	Newman v4.3.1

	Total	Failed
Iterations	1	0
Requests	63	0
Prerequest Scripts	63	0
Test Scripts	108	0
Assertions	142	0

Total run duration	6.5s
Total data received	56KB (approx)
Average response time	70ms

Total Failures **0**

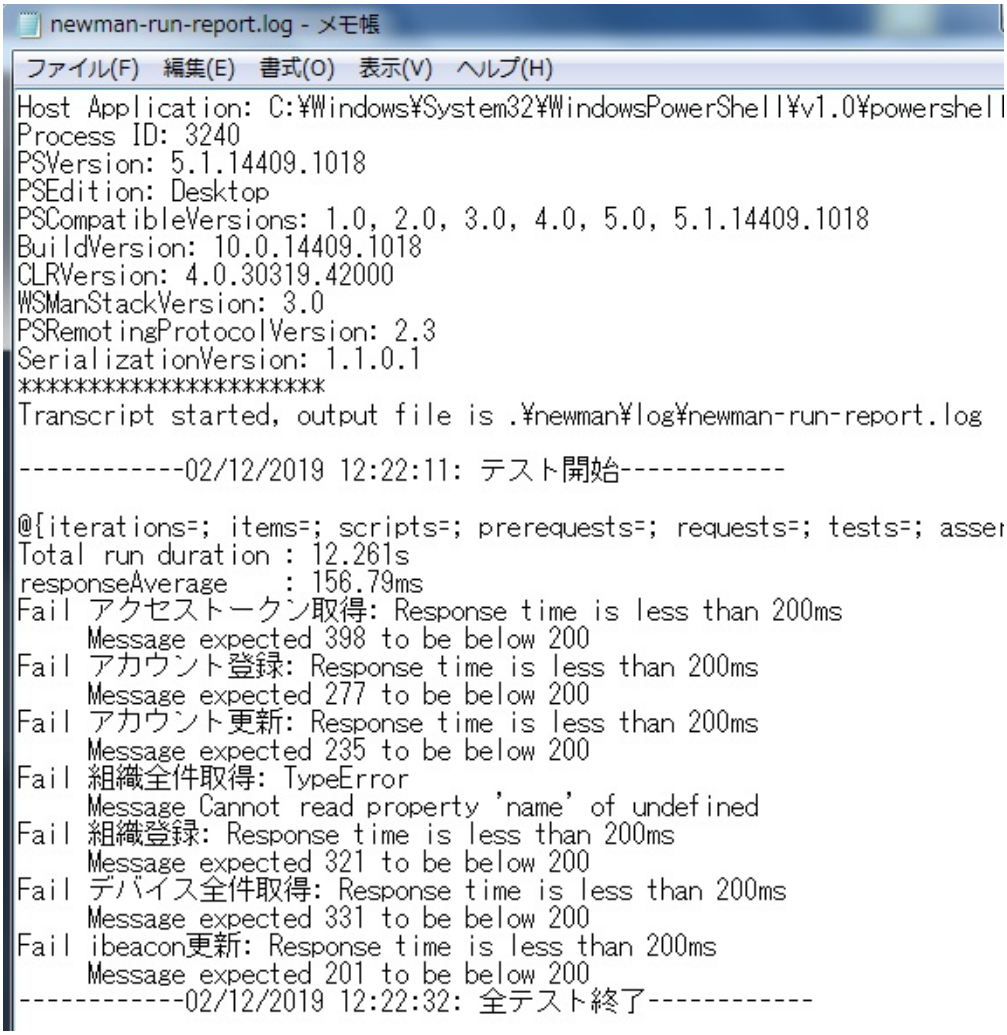
Requests

アクセストークン取得

Method	POST
URL	https://admin-api-dev.[redacted].net/access_tokens
Mean time per request	261ms
Mean size per request	269B

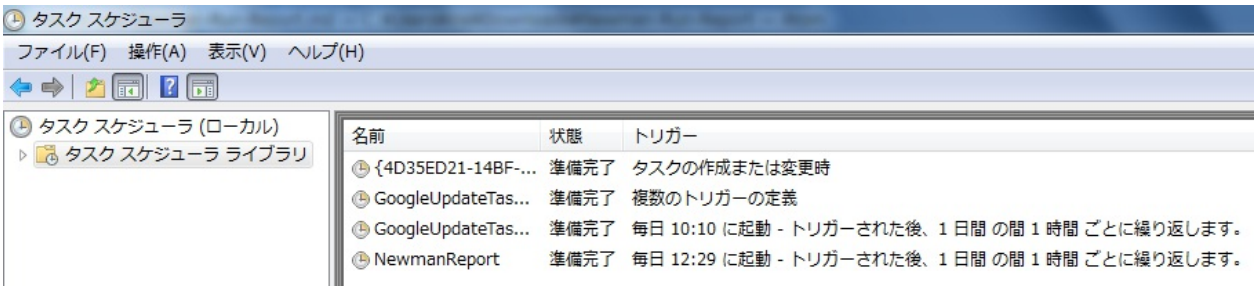
`newman\log` ディレクトリにはcsv形式の結果とlog形式のスクリプトログ

Time	TotalRunDurarion	ResponseAverage	PassTests	FailTests	PassTestScripts	FailTestScripts
2019/2/1 15:36	6.499	69.38	63	0	108	0
2019/2/1 15:36	6.363	69.48	63	0	108	0
2019/2/1 15:37	6.247	68.51	63	0	108	0
2019/2/1 15:37	6.436	71.7	63	0	108	0
2019/2/1 15:37	6.717	73.56	63	0	108	0
2019/2/1 15:38	6.489	71.6	63	0	108	0
2019/2/1 15:38	6.37	69.49	63	0	108	0
2019/2/1 15:39	6.667	73.21	63	0	108	0
2019/2/1 15:39	6.46	69.52	63	0	108	0
2019/2/1 15:39	6.29	67.54	63	0	108	0
2019/2/1 15:40	6.323	69.35	63	0	108	0
2019/2/1 15:40	6.653	71.21	63	0	108	0



さらに、任意のSlackチャンネルに結果の通知が送られます

また、Windowsタスクスケジューラに New man Run Reportを登録するとテストを好みのタイミングかつ、バックグラウンドで行うことができます



必要条件

- Windows Powershell 3.0以上
- Node.js via package manager v6以上
- Newman v4

使用方法

流れ

1. Postmanをインストール
2. Postmanにて `Collections` を作成し、json形式で出力
3. Newmanをインストール
4. Slack Incoming Webhook URL を取得
5. `Newman-Run-Report` に入っている `Report.ps1` を編集し、同じディレクトリに#2で作成したjsonファイルを配置
6. （オプション）WindowsタスクスケジューラにNew man Run Reportを登録する

1.Postmanをインストール

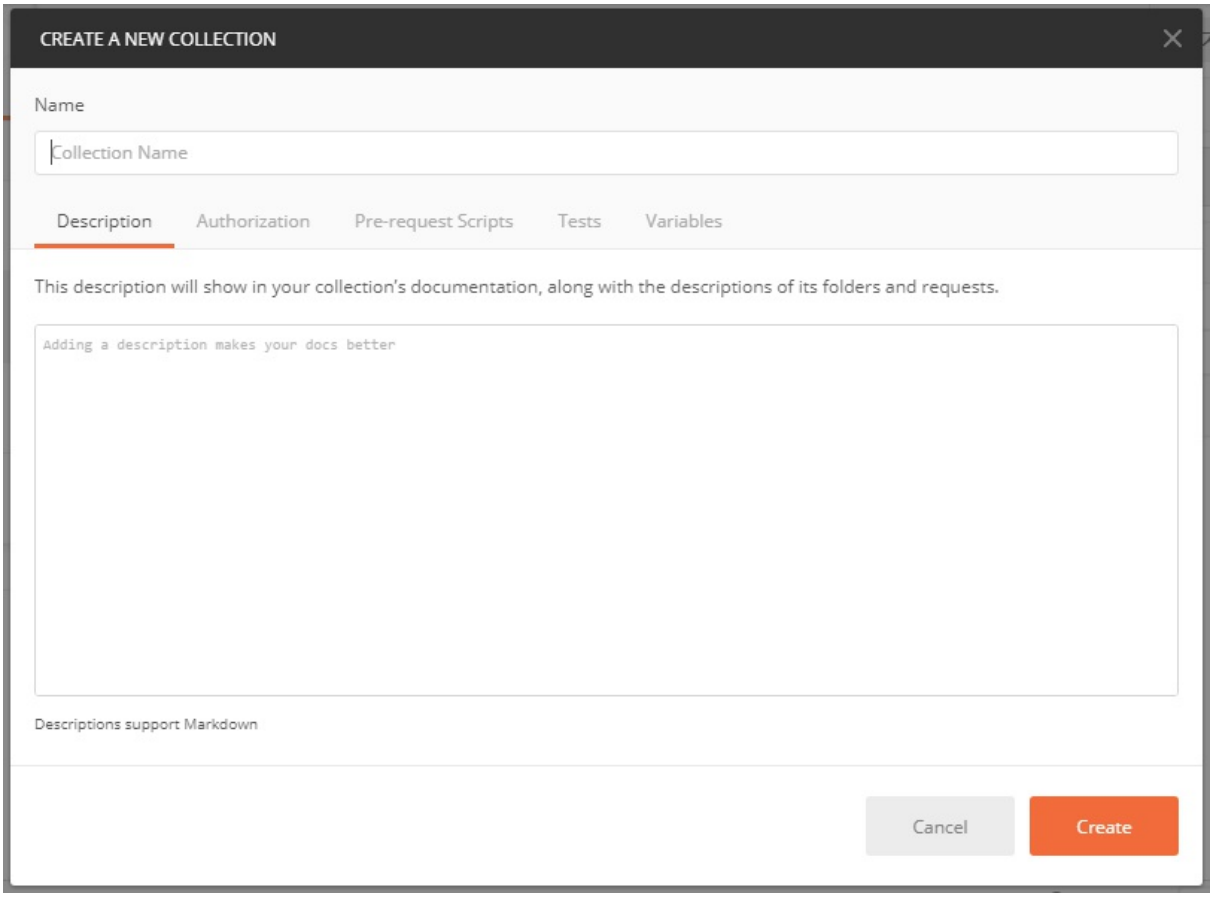
まだインストールしていない場合はこちらからインストールしてください

2.Postman

Collectionsを作成

1. 画面左側の `+` を押す

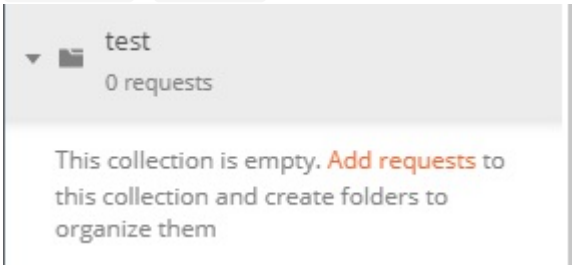
2. 名前を適当につけたら右下の `Create` で作成

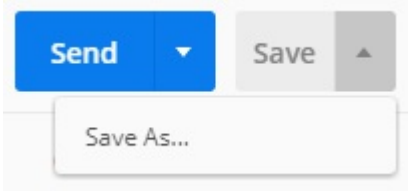


他の項目は後から変更できます

Requestsを作成

`Add requests` や `Save As...` などからCollectionsにRequestsを追加

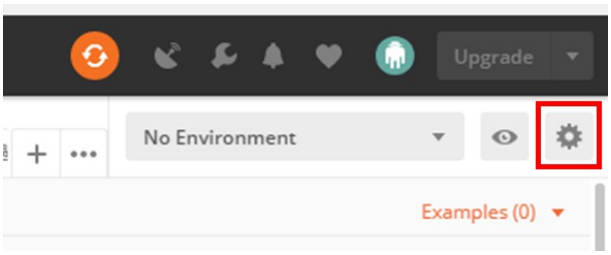




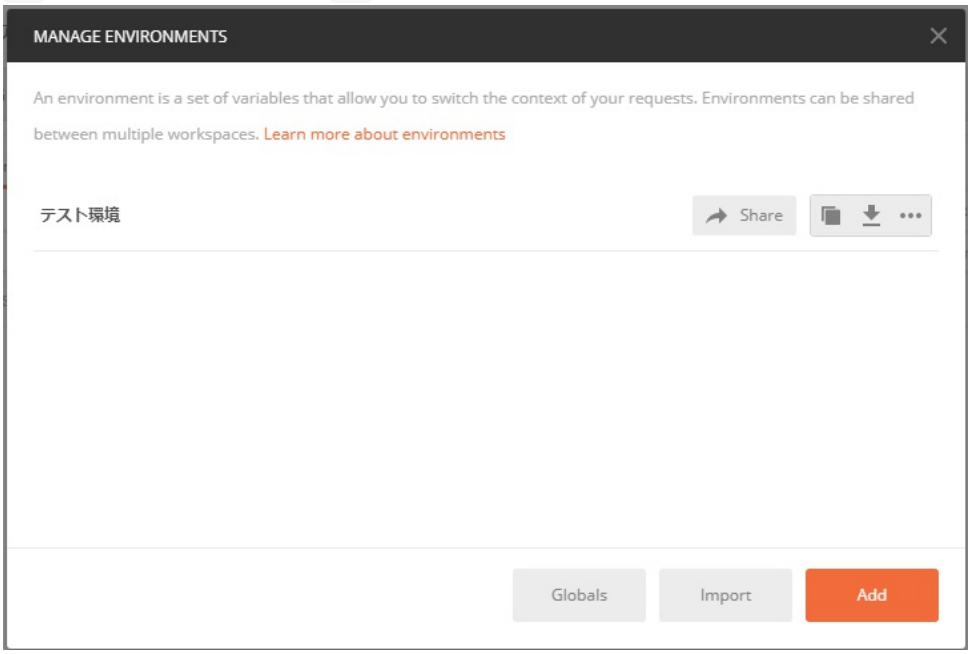
お好みでRequests内容を編集してください

トークン設定

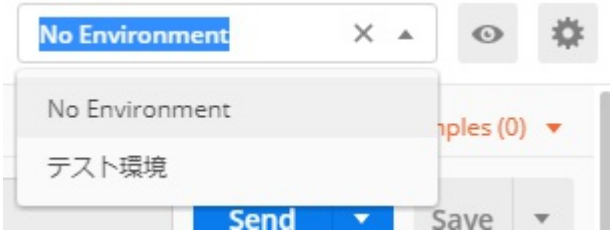
1. 画面右上の `歯車` を押す



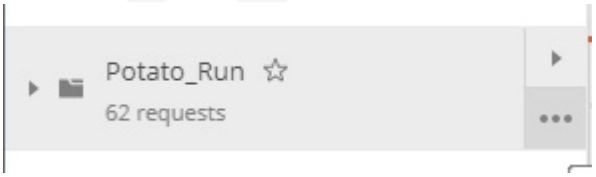
2. `Add` を押し、適当な名前をつけて再び `Add`



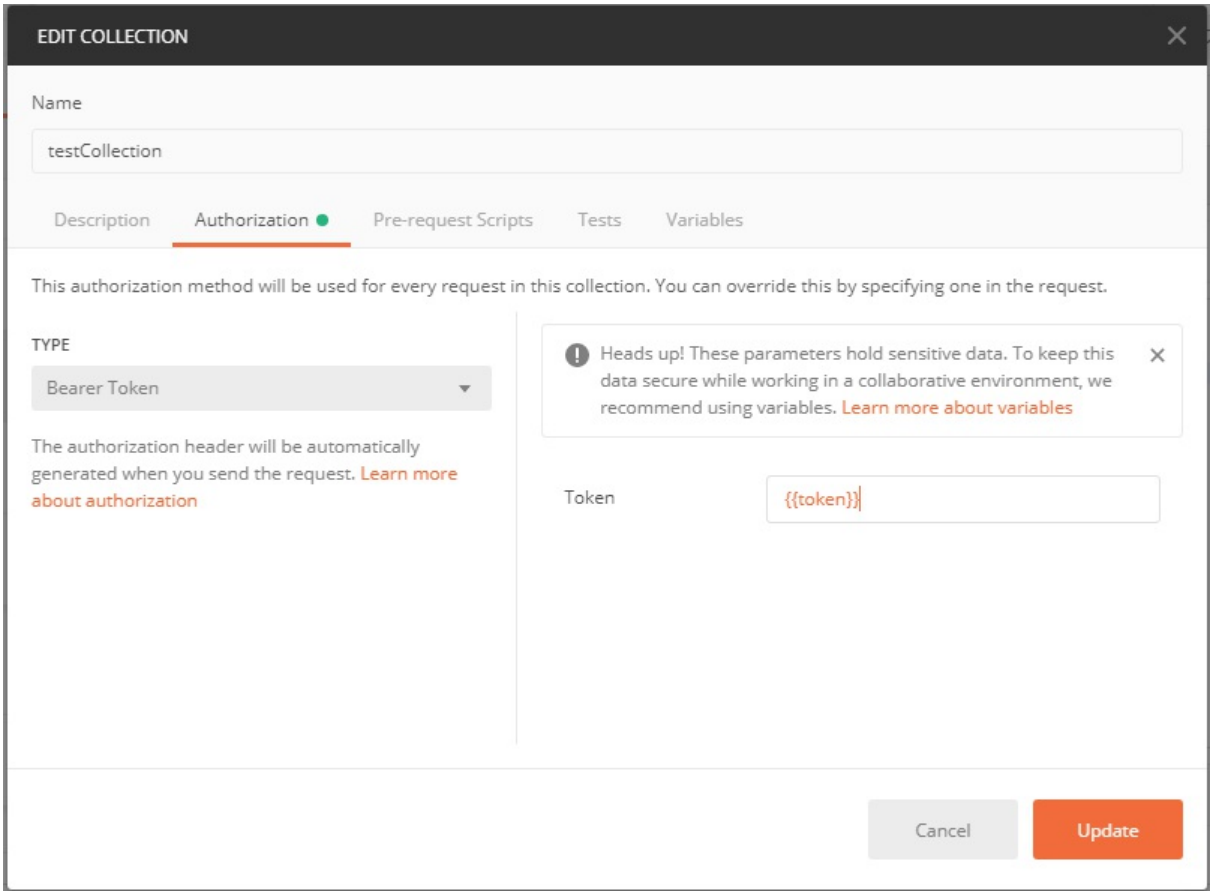
3. 画面右上の `No Environment` を押し、作成した環境を選択



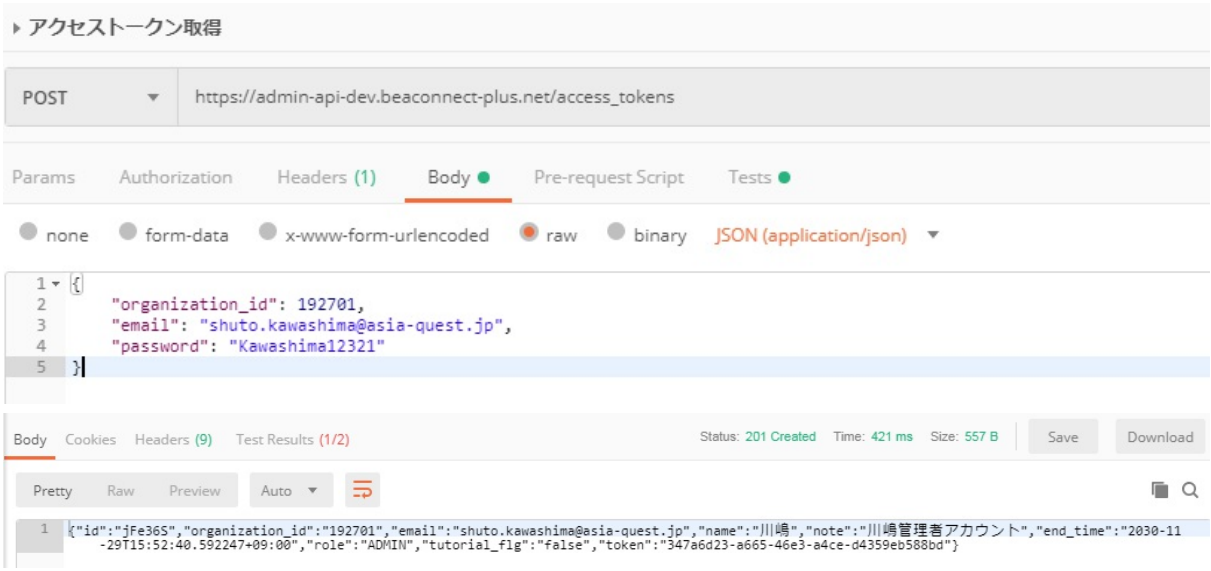
4. Collectionsの `...` を押し、`Edit` に進む



5. `Authorization` タブのTYPEを任意のものに変更し、Tokenを `{{token}}` とする

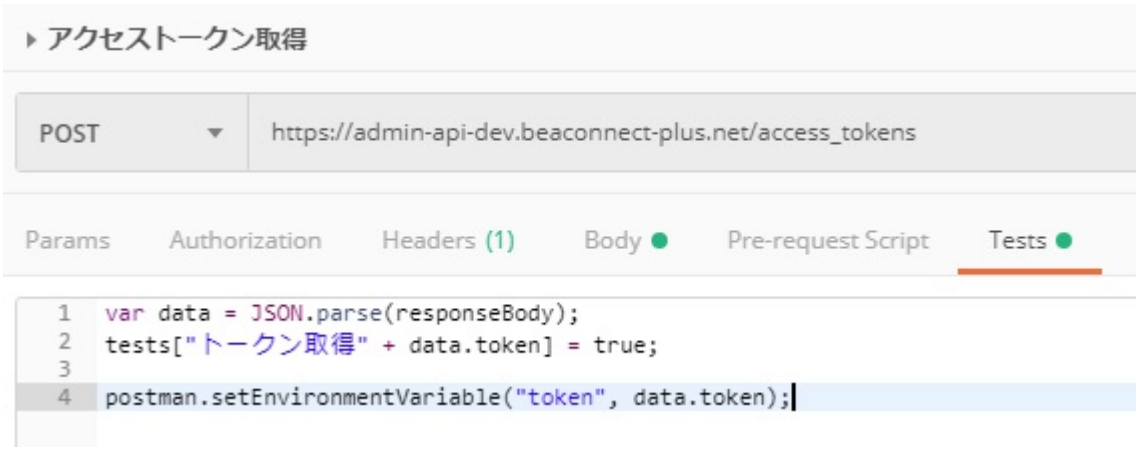


6. BodyにtokenがReturnするRequestsの場合



`Tests` に以下の記述をするとPostman内の環境変数 `token` にBody内のtokenが代入されます

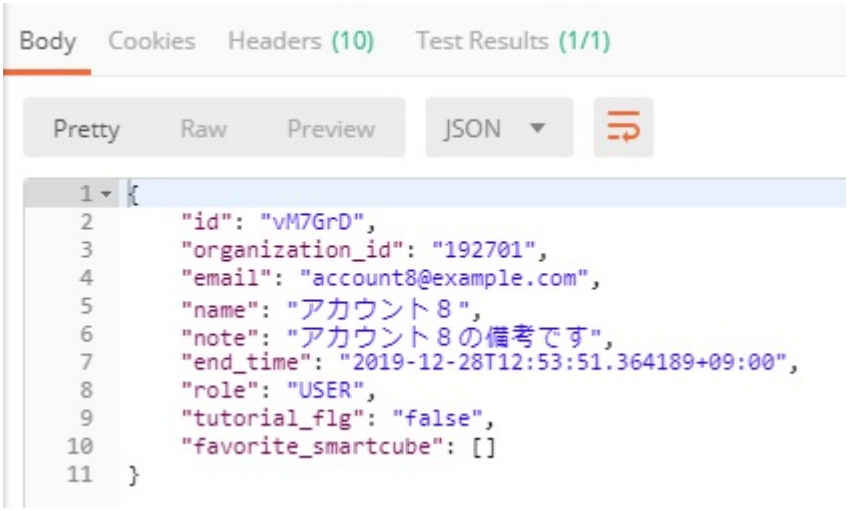
```
var data = JSON.parse(responseBody);
postman.setEnvironmentVariable("token", data.token);
```



Tests設定

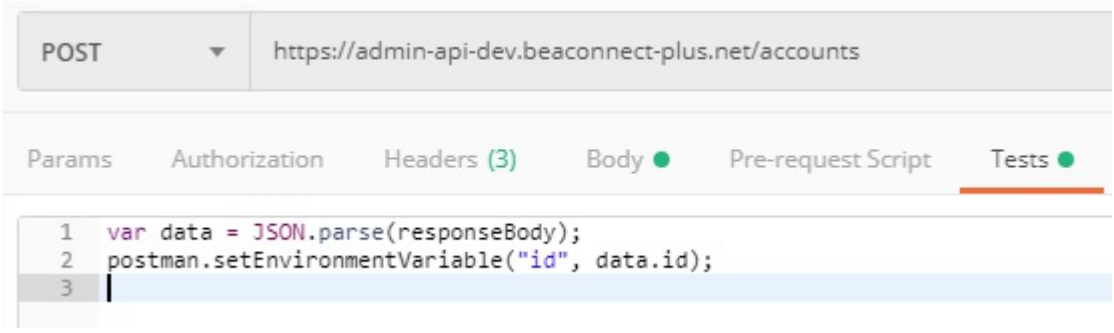
もし `アカウント登録` → `アカウント削除(更新)` にランダムなアカウントIDを用いていた場合、一連の流れでテストを行うことができません。そこで、先ほどのトークン設定で用いた環境変数に代入を利用します。

アカウント登録のリクエストを送ると以下のようなBodyが返ってくるので

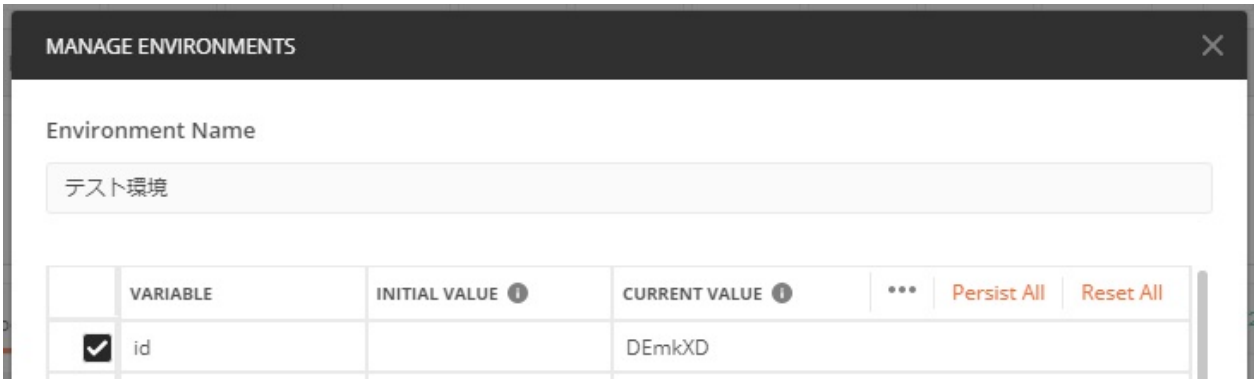


Testsタブに以下の記述を入力するとBodyに返ってきた連想配列内の `id` がPostmanの環境変数に `id` として登録されます

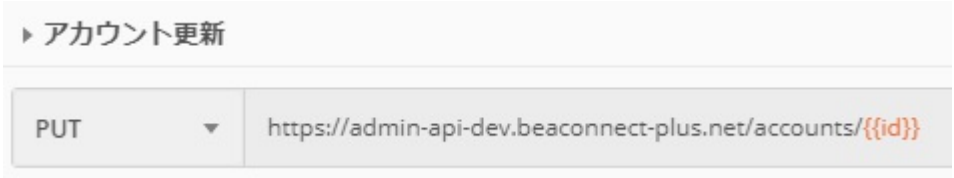
```
var data = JSON.parse(responseBody);
postman.setEnvironmentVariable("id", data.id);
```



環境画面を見るとVARIABLEに `id` が追加されていることが確認できます



次にアカウント更新のURLを次のように変更します

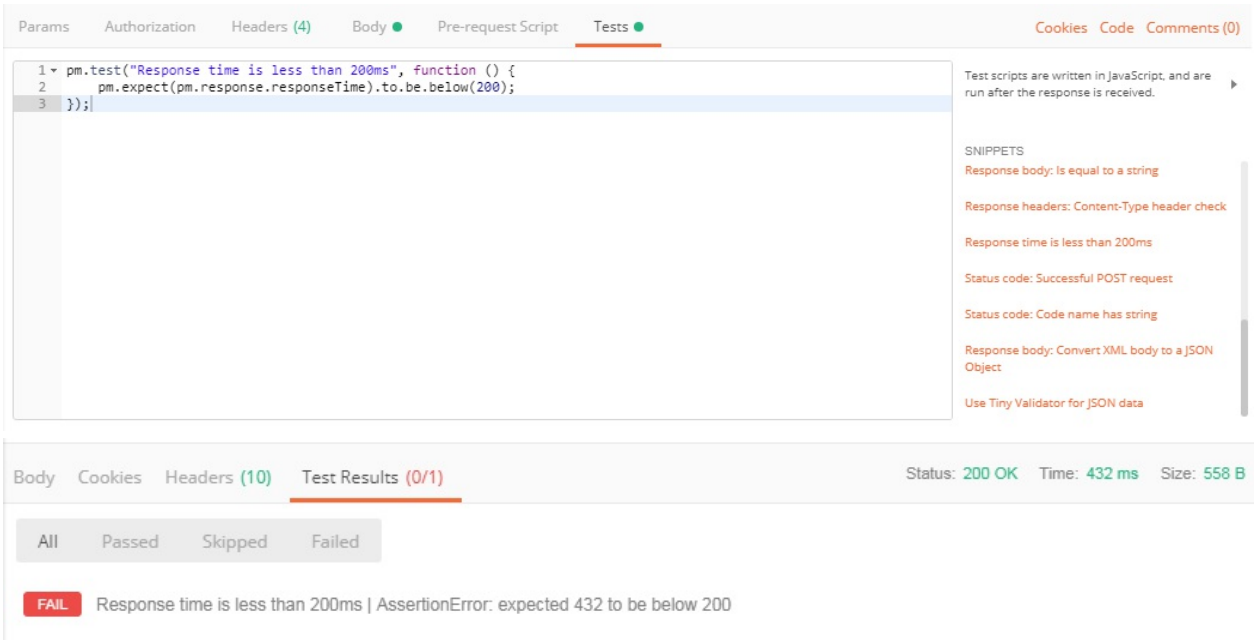


Postmanでは `{{環境変数名}}` と入力すれば基本的にどこでも環境変数に置き換えられます。そのため `アカウント作成` のTestsで環境変数 `id` に代入された値がURLに代入され、ランダムなアカウントIDに対応することができます。

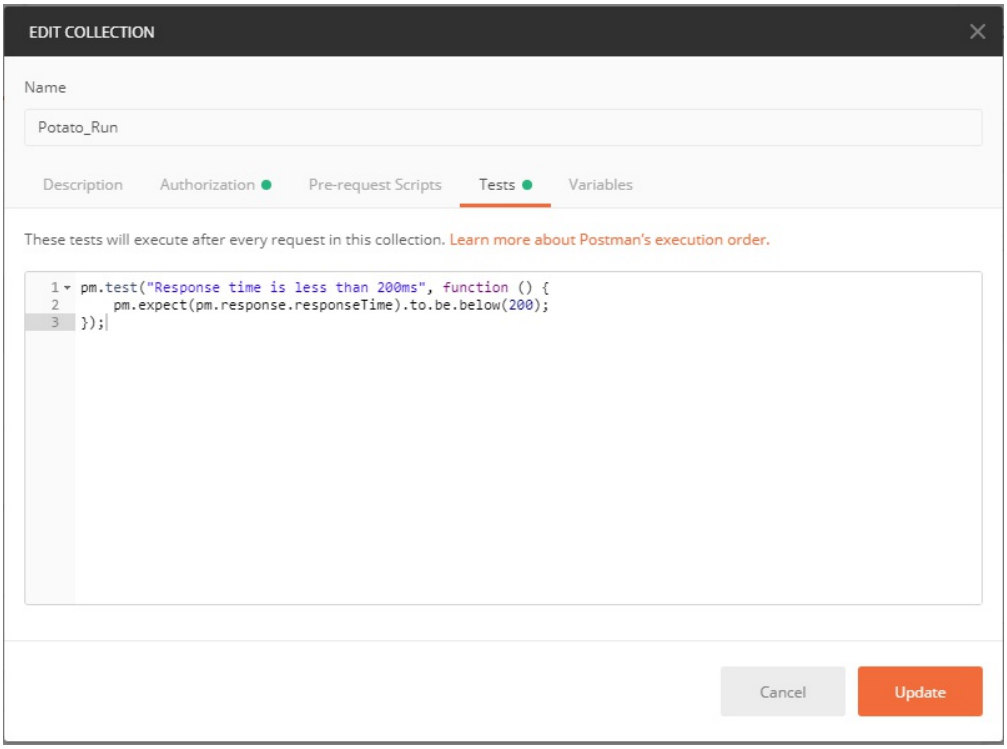
これまでは変数代入のためにTestsを利用しましたが、このTestsにはその名のとおりテストしたい内容を記述してテストすることができます。
例えば

```
pm.test("Response time is less than 200ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(200);
});
```

と記述すればレスポンス200ms以上の場合、Failedとなります



このTestsはCollectionsごとにも管理できます

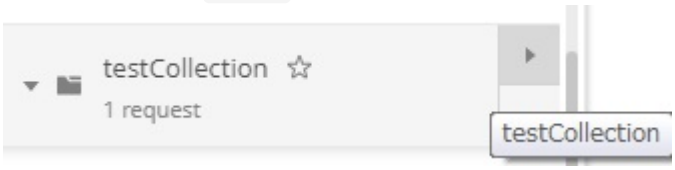


テストスクリプトはjavascriptで記述でき、画面右の `SNIPPETS` から追加できる例文や公式ドキュメントを参考にカスタマイズするとさらなる効率化が図れます

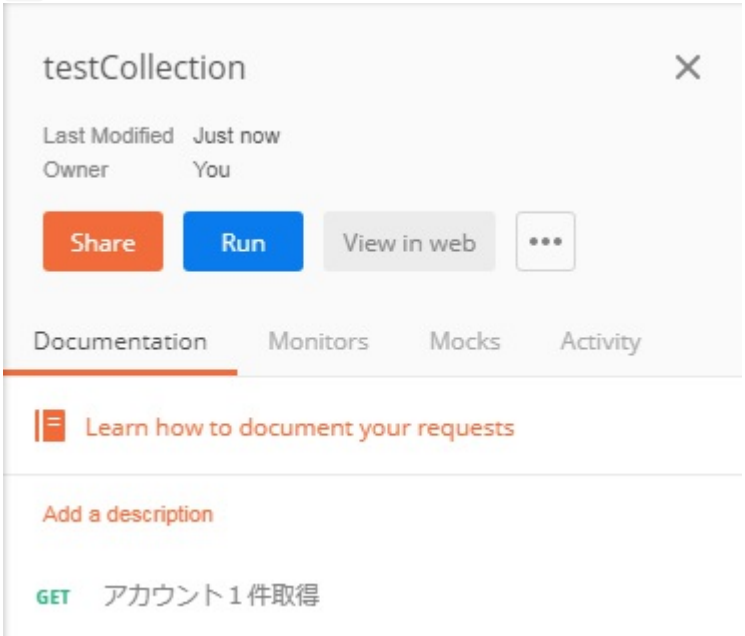
Runnerで確認

作成したCollectionsを一括でテストしてみましょう

1. 作成したCollectionsの `三角マーク` を押す

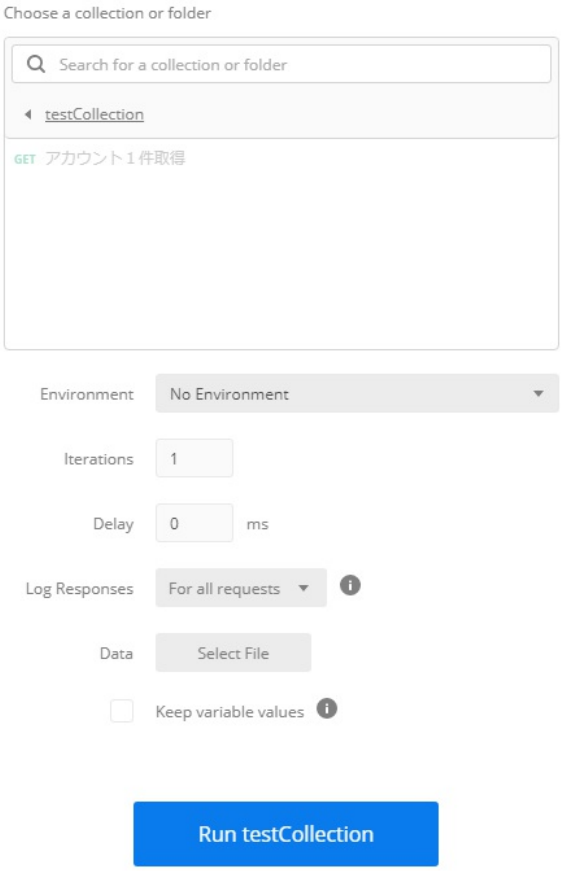


2. `Run` を押す

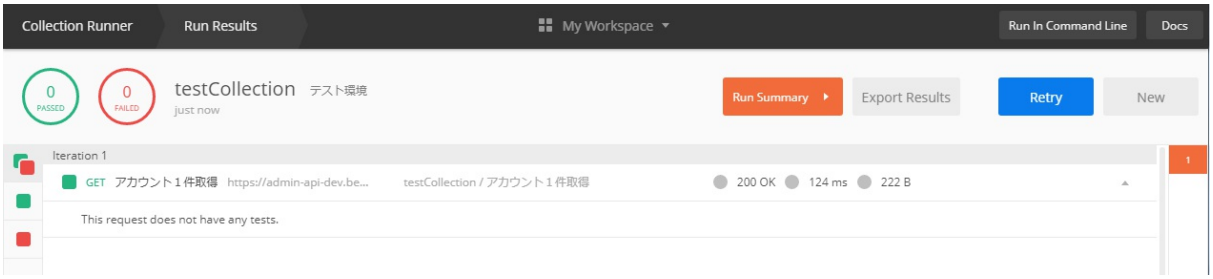


Runnerが起動します

3. 左下の `Run Collection` を押す



4. 自動でコレクション内のテストが始まり、結果が出ます

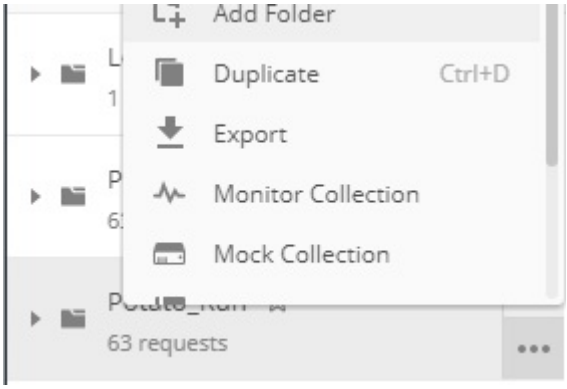


作成した場合、Testsの結果も出ます

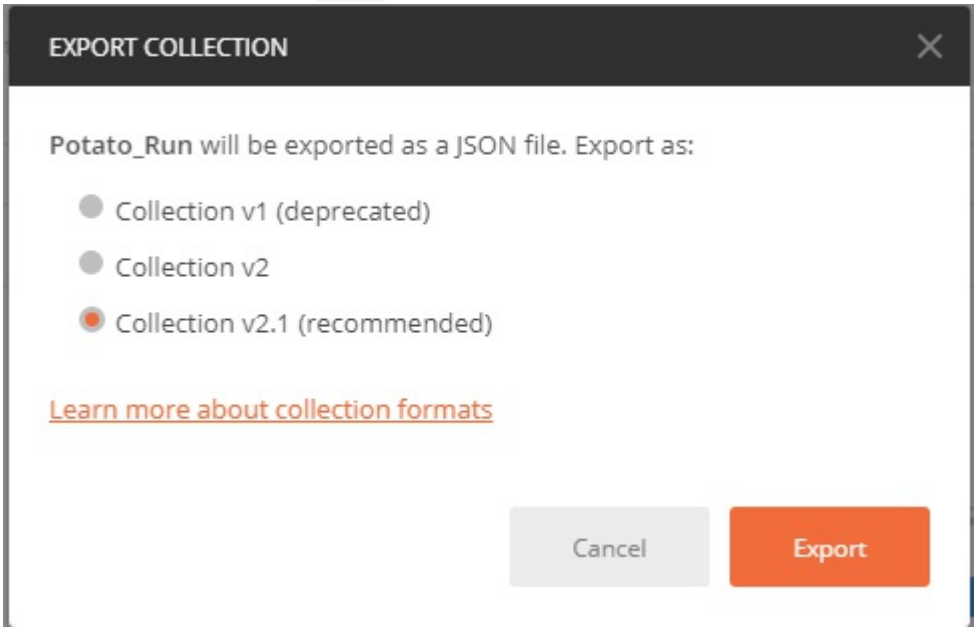
json出力

ここまでがうまくいっているのであればコレクションをjson形式で出力します

1. Collectionsの `...` から `Export` を押す



2. v2.1が選択されていることを確認し、`Export`を押す



3.Newmanをインストール

1. New manのインストールにはNode.js via package manager v6 以上が必要です
   無い場合ここから最新版をインストールしてください

2. コマンドラインで`npm install -g newman`と入力

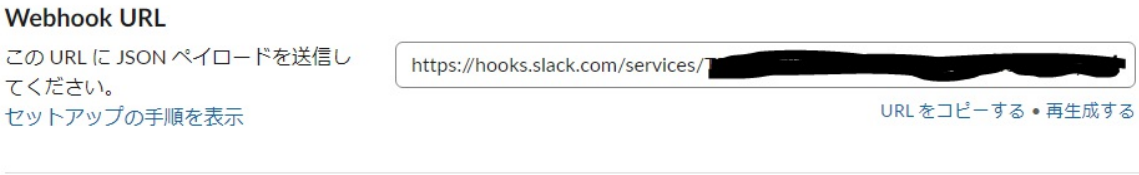
3. (オプション)コマンドラインで`newman -v`と入力し、バージョンが返ってくるか確認

4.Slack Incoming Webhook URLを取得

1. Slackの通知を受けたいワークスペースにサインインした状態でSlack Incoming Webhookにアクセス

2. 通知したいチャンネルを選択し、インテグレーションの追加を押す

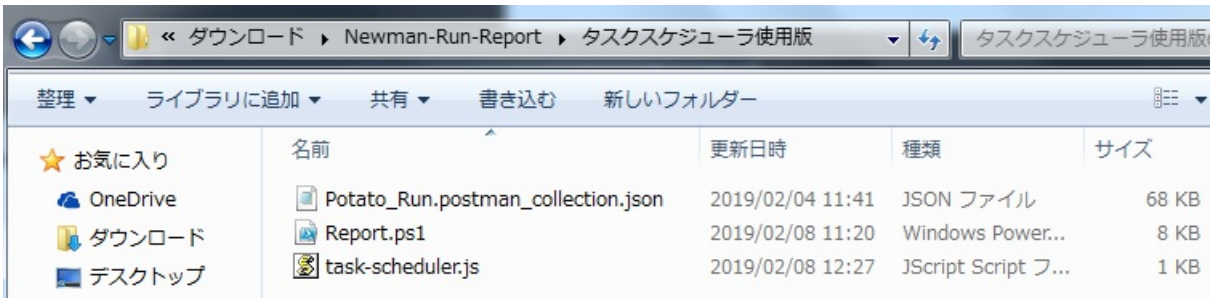
3. Webhook URLに記述されているURLをコピー



5.Newman Run Report

準備1 - スクリプト編集

1. `Newman-Run-Report\タスクスケジューラ使用版 or 未使用版` 内のReport.ps1と同じディレクトリに「Postman」で作成した `collection.json` を配置



2. `Report.ps1` をメモ帳などで開き、`#` で囲まれた部分を編集する
`newman run example.postman_collection.json -r html,json` の部分のjson名を「Postman」で作成した `collection.json` の名前に編集

```
Start-Transcript -path ".¥newman¥log¥newman-run-report.log" -Append #テキストログ出力
$startDate = Get-date
$nowDate = Get-date
Write-Host "`r`n------------${nowDate}: テスト開始------------" -ForegroundColor Green

##############################################################################################################

newman run example.postman_collection.json -r html,json #PostmanのCollectionファイルを指定

##############################################################################################################

# .¥newman¥最新json 読み込み
$files = @(Get-ChildItem -Path .¥newman -Filter *.json)
```

`$webhook = "https://hooks.slack.com/services/XXXXXXXXX/XXXXXXXXX/XXXXXXXXXXXXXXXXX` の部分を「Slack Incoming Webhook」で取得したURLに編集

```
#################################################################################################################
$webhook = "https://hooks.slack.com/services/XXXXXXXXX/XXXXXXXXX/XXXXXXXXXXXXXXXXXXX" #Slack Webhookを入力
#################################################################################################################
$json = ConvertTo-Json $payload
$body = [System.Text.Encoding]::UTF8.GetBytes($json)
Invoke-RestMethod -Uri $webhook -Method Post -Body $body
```

（オプション）準備2 - Pwershell設定

New man-Run-Reportを使用するには `Windows Powershell 3.0以上` が必要ですが、
OSごとにデフォルトでインストールされているバージョンが違います。
バージョンが気になる方はPow ershellに `Get-Host` と入力するとバージョンが確認できます。

Windows OSバージョン	デフォルト	導入可能
Windows Server 2016	5.1	(6.0)
Windows 10 RS1	5.1	(6.0)
Windows 10	5.0	5.1/(6.0)
Windows Server 2012 R2	4.0	5.0/5.1/(6.0)
Windows 8.1	4.0	5.0/5.1/(6.0)
Windows 2012	3.0	4.0/5.0/5.1/(6.0)
Windows 8	3.0	4.0/5.0/5.1/(6.0)
Windows Server 2008 R2	2.0	3.0/4.0/5.0/5.1/(6.0)
Windows 7 SP1	2.0	3.0/4.0/5.0/5.1/(6.0)
Windows Server 2008 SP2	1.0	2.0/3.0/4.0/5.0/5.1/(6.0)

デフォルトのバージョンが2.0以下の場合はPow ershellのバージョンアップを行ってください
`Windows Manage Framework (WMF)` をバージョンアップするとWMFに含まれるPow ershellもバージョンアップされます。今回は比較的新しく、安定しているPow ershell5.1にアップデートします。

WMF5.1のインストールと構成からご使用のOSに合うパッケージをダウンロード
しかしこのままではうまく実行できません。デフォルトでは `Posershellスクリプト(.ps1)` の実行禁止という実行ポリシー^1になっているからです。なので、実行ポリシーを変更します。

Powershellの実行ポリシーの変更

管理者として実行しているPow ershell上で `Set-ExecutionPolicy AllSigned` と入力

```
PS C:¥hoge> Set-ExecutionPolicy AllSigned

実行ポリシーの変更
実行ポリシーは、信頼されていないスクリプトからの保護に役立ちます。実行ポリシーを
変更すると、about_Execution_Policies
のヘルプ トピックで説明されているセキュリティ上の危険にさらされる可能性がありま
す。実行ポリシーを変更しますか?
[Y] はい(Y)  [N] いいえ(N)  [S] 中断(S)  [?] ヘルプ (既定値は "Y"): y
PS C:¥hoge>
```

すると画像のように質問されるので `Y` を入力

再び、Pw ershellバージョンアップ作業にもどります

1. **ダウンロードしたzipファイルを解凍し、`Install-WMF5.1.ps1` を管理者としてPw ershellで実行**
2. **実行するか聞かれるので `R` を入力**

```
C:¥hoge> Install-WMF5.1.ps1

この信頼されていない発行元からのソフトウェアを実行しますか?
ファイル C:¥hoge¥Install-WMF5.1.ps1 の発行元は CN=Microsoft Corporation,
 OU=MOPR, O=Microsoft Corporation, L=Redmond, S=Washington, C=US
であり、このシステムで信頼されていません。信頼された発行元からのスクリプトのみを
実行してください。

[V] 常に実行しない(V)  [D] 実行しない(D)  [R] 一度だけ実行する(R)  [A] 常に実行
する(A)  [?] ヘルプ (既定値は "D"):
```

あとは表示に従って進めていくとバージョンアップができます

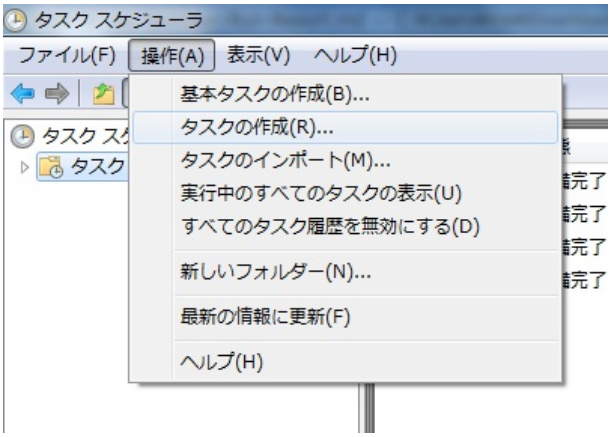
スクリプト実行

`Report.ps1` **をPow ershellで実行すると** `newman` **フォルダが作成され、その中にログがたまっていきます。また、Slackの指定したチャンネルに結果が通知されます。**

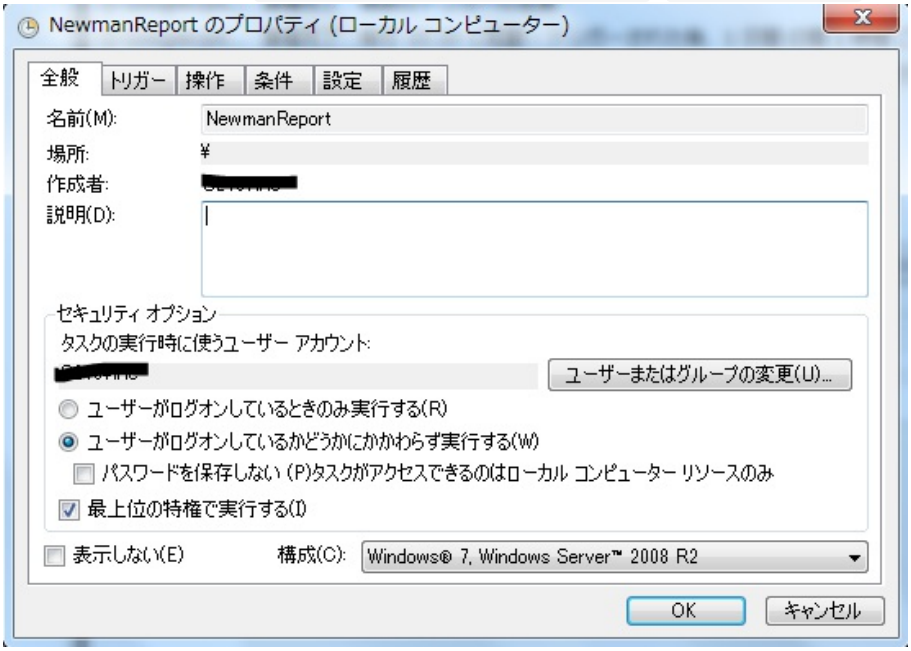
6.（オプション）WindowsタスクスケジューラにNewman Run Reportを登録する

テストを一定期間ごとに行いたい場合はWindow sタスクスケジューラにNew man Run Reportを登録します

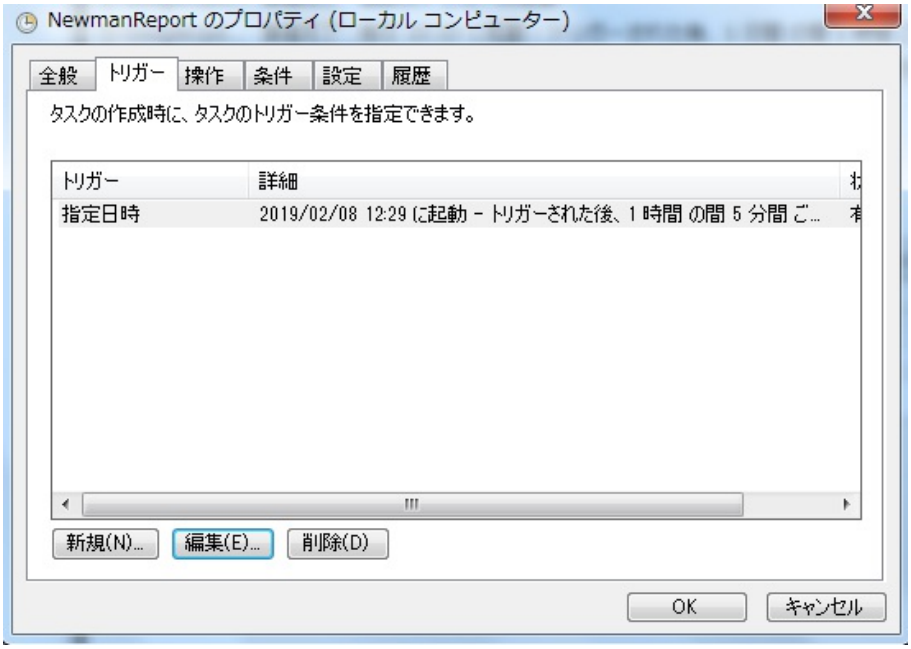
1. **タスクスケジューラを起動**
2. `操作` → `タスクの作成` **をクリック**



3. 全般の名前は任意のものを、ユーザーがログオンしているかどうかにかかわらず実行する と 最上位の特権で実行する にチェック



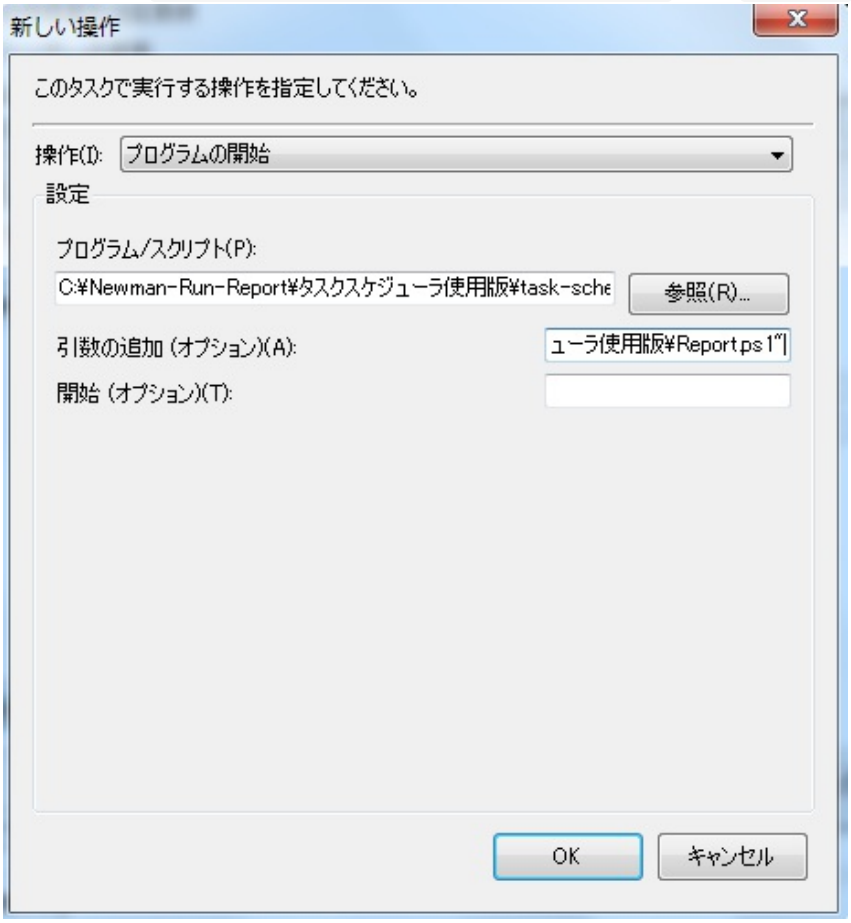
4. トリガーは任意のものを追加



5. 操作の 新規 から新しい操作を作成

6. **「プログラム/スクリプト」を** `C:\Newman-Run-Report\タスクスケジューラ使用版\task-scheduler.js` **のように** `task-scheduler.js` **のパスを指定**

7. **「引数の追加」を** `"C:\Newman-Run-Report\タスクスケジューラ使用版\Report.ps1"` **のように** `Report.ps1` **のパスを指定**



8. **そのほかの項目はお好みで変更し** `OK` **を押すとトリガー条件を満たしたときに**New man Run Report**が実行されます**

> Name: New man Run Report
> License: This softw are is released under the MIT License.
> Created date: 2019/02/12
> Author: Kouki Ooe