

Running Tests in ADO Release Pipeline

Don't put credentials or keys for any Revver user in postman environment variables that have access to real customer data, or any other sensitive data in case these API keys were ever compromised.

You can run a Postman collection with a specific environment in an ADO release pipeline stage. The following steps can be taken to set this up.

1. Click on the **...** (three dots) icon next to the collection you want to run in the pipeline, which is outlined in **BLUE** in image 1.
2. Select **Run Collection** option in the dropdown that appears, which is outlined in **YELLOW** in Image 1.
3. In the right panel, in the **Choose how to run your collection** section, select **Automate runs via CLI**, which is outlined in **RED** in Image 1.
4. In the bottom of the right panel, click the **Configure command** link, which is outlined in **GREEN** in Image 1.
5. You will be taken to a new page.
6. In the **Choose the collections you want to run** section, make sure the correct **Collection** and **Environment** are selected. This is outlined in **ORANGE** in image 2.
 1. You can select multiple collection / environment pairs if you wish
7. In **CI/CD configuration** section, make sure **Azure Pipelines** and **Windows** is selected. This is outlined in **PURPLE** in image 2.
8. You will need to generate a new **API Key** if you don't already have one setup. You can do this by clicking the **Generate API Key** button, outlined in **PINK** in image 2.
9. In the azure pipeline stage that you wish to run the Postman tests in you will need to **Powershell** tasks. The first task will install the Postman CLI on the build agent, and the second task will authenticate with your API key and run the collection. On each task select the **Inline** type, and copy the scripts (highlighted in **BROWN** in image 2) into the **script** section of the task.
 1. In the second script you will need to replace **\$(POSTMAN_API_KEY)** with your actual postman API key.

Image 1

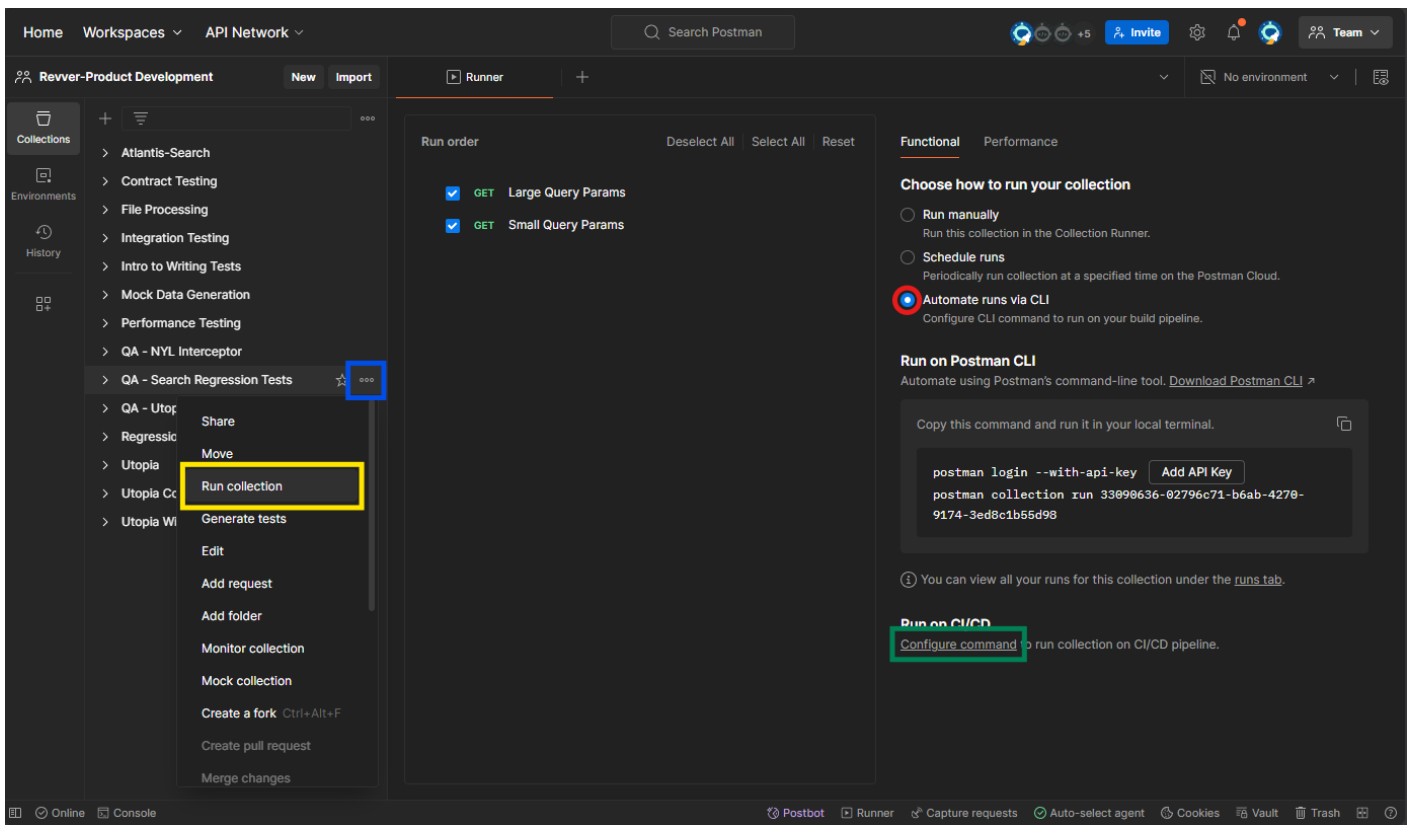


Image 2

Generate Postman CLI Configuration

Run your tests and validations on CI/CD using Postman CLI configuration. [Download Postman CLI](#) ↗

Choose the collections you want to run

Collection

QA - Search Regression Tests

Environment

QA - Utopia Search - Staging

+ Add Another Collection

CI/CD configuration

CI/CD Provider

Azure Pipelines

Operating system for CI/CD

Windows

Postman CLI command preview

Copy this command and paste it to your build configuration file

```
1 pool:
2   vmImage: 'windows-latest'
3
4 steps:
5   - script: |
6     powershell.exe -NoProfile -InputFormat None -ExecutionPolicy AllSigned -Command "[System.Net.ServicePointManager]
7       ::SecurityProtocol = 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://dl-cli.pstmn.io/
8       install/win64.ps1'))"
9     displayName: 'Install Postman CLI'
10
11   - task: CmdLine@2
12     displayName: 'Run automated API tests using Postman CLI'
13     inputs:
14       script: |
15         postman login --with-api-key $(POSTMAN_API_KEY)
16         # Run your collection using Postman CLI
17         postman collection run "33090636-b53a1059-f118-4009-a5c4-952fca8d38fd" -e
18           "33090636-6524ed64-6853-4a3b-9977-4608b71c7706"
```

Note: Postman API key is needed to log in to Postman CLI

We recommend that you save your Postman API key as a secret environment variable in your CI/CD pipeline to prevent exposure. The login command already contains \$POSTMAN_API_KEY variable to get you started.

Generate API Key

Copy Postman CLI Command

Image 3

eFileCabinet / Utopia / Pipelines / Releases / Utopia Release

U

All pipelines > Utopia Release

Save Create release View releases

Pipeline

Tasks

Variables

Retention

Options

History

Run Postman Tests V2

Deployment process

Agent job

Run on agent

Install Postman CLI

PowerShell

Run automated API tests using Postman CLI

PowerShell

PowerShell

Task version 2.*

Display name *
Install Postman CLI

Type

File Path

Inline

Script *

```
powershell.exe -NoProfile -InputFormat None -ExecutionPolicy AllSigned -Command "[System.Net.ServicePointManager]::SecurityProtocol = 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://dl-cli.pstmn.io/install/win64.ps1'))"
```

Preference Variables

Advanced

Control Options

Environment Variables

Output Variables

Revision #2
Created 29 May 2024 17:16:10 by Quinn Godfrey
Updated 29 May 2024 19:36:26 by Quinn Godfrey