How To Guide to using RESTful API calls for a 2.5 million identity dataset.

Subscription FOCUS	Postman Collection Links	Instructional / Demonstration Video Links
TAG	https://testdataservices.com.au/docs/Postman_TAG_Public	https://testdataservices.com.au/docs/TDS_HowTo_Postman_TAG.mp4
OIDC	https://testdataservices.com.au/docs/Postman_OIDC_Public	https://testdataservices.com.au/docs/TDS_HowTo_Postman_OIDC.mp4
EMAIL	https://testdataservices.com.au/docs/Postman_EMAILCognito_Public (This collection signs up a new user to a real Cognito Website)	https://testdataservices.com.au/docs/EmailAutomationWIthPostman.mp4
EMAIL	https://testdataservices.com.au/docs/Postman_EMAIL (This collection shows how to handle of sets of email addresses)	https://testdataservices.com.au/docs/TDS_HowTo_EMAILSubscriptions.mp4
MFA	https://testdataservices.com.au/docs/Postman_MFA	https://testdataservices.com.au/docs/MFAAutomationWithPostman.mp4

Other Useful Videos:

Introduction to OpenID Connect: https://testdataservices.com.au/docs/OIDC_intro.mp4

Part 2 of OpenID Connect: https://testdataservices.com.au/docs/OIDC_with_postman_part2.mp4

Shifting towards ‘event driven’ email based testing: https://testdataservices.com.au/docs/TDS_EmailProcessing.mp4

Very Fast Signup – using event driven process: https://testdataservices.com.au/docs/TDS_VeryFastSignup.mp4

Email Storms – How and Why to create one: https://testdataservices.com.au/docs/TDS_EmailStormDemo.mp4

Complex workload profiles made easy: https://testdataservices.com.au/docs/TDS_Workload_v2.mp4

Postman Visualization with Complex Workload: https://testdataservices.com.au/docs/TDS_Postman_LoadTest.mp4

Discussion on Dataless Testing: https://testdataservices.com.au/docs/TDS_Dataless_Population.mp4

Quick Start walkthrough: https://testdataservices.com.au/docs/TD_QuickStartGuide.mp4

Including MFA in Test Automation: https://testdataservices.com.au/docs/TD_MFA_75.mp4

Downloadable subset of 25,000 manufactured identities: https://testdataservices.com.au/docs/Person_v0001F_25K.csv

Please note: The subscription key **dNQNhjq2GeWJ9DNQuaatZ7shdQKSQ2Uz** will work, but is significantly rate limited and should not be used in formal testing projects.

What problems are solved by using these services?

Problem	Solution
How can I automatically sign up a user to a service when it involves an email verification step? Automating a test script that needs to read the contents of an email has been difficult and messy. This problem impacts on manual testing, automated functional testing, performance and load testing and application performance monitoring.	Watch this short video (https://testdataservices.com.au/docs/EmailAutomationWIthPostman.mp4) that shows how to use a few simple calls to catch emails and extract the required verification code from the email message. You can open up this Postman Collection (https://testdataservices.com.au/docs/Postman_EMAILCognito_Public and step through the same steps as shown in the video, which actually signs up a random user to a production web site, including the email catching step.

How can I measure backend email generation response times? The time it takes for an end user to receive an email, in response to some event it usually very important. When a system is under heavy workload, the time taken to generate emails can degrade severely, so having a means of accurately measuring this 'response time' is very important.	When an email is fetched using https://api.testdataservices.com.au/v0001F_GetMessage a JSON response is returned, and it contains the 'Latency', which is the number of seconds from when the 'Clear Message' call was made, until the Email was received for processing. In the example below, the Latency was 14.439 seconds.
	
	`1.us-east-1.O6wNxoLBrI1fdA69qfQJ23/PCQ1XHhdgXfeF46CfFwc=:AmazonSES\r\n\r\n------=_Part_10188_1203383410.1585454023377\r\nContent-Type: text/html; charset=UTF-8\r\nContent-Transfer-Encoding: 7bit\r\n\r\n`**`Your verification code is 899155.`**` `
	`\r\n------=_Part_10188_1203383410.1585454023377--\r\n\r\n",`
	`"UID": "9f6cf933-764a-4dfc-9560-56dd04c58602",`
	`"EmailAndFromDomain": "sandra.whitley.1865@testdata.email|testdata.email",`
	`"Latency": 14.439`

How can I include a realistic 'email storm' event in my Load Test? If a system generates emails in response to a given event, and the recipients of those emails undertake actions that trigger additional emails being generated, then you have an 'Email Storm' scenario. This has traditionally been very difficult to simulate, because the ramp-up of workload is dependent on backend email generation time and the inherent difficulty of test scripts being capable of responding to those emails in a timely manner.	By making use of more sophisticated email catching calls, it is possible to 'subscribe' to a set of emails for a given 'From Domain'. Your test scripts can then implement a process loop that continuously checks if any new messages have arrived for any of the email addresses in the subscribed set, and process those messages. The example below shows the JSON response from such a call. Note that the UID of the recipient is included in the JSON structure, so that a subsequent call (in this case https://api.testdataservices.com.au/v0001F_GetPerson?UID=4c370969-2ceb-4d64-a1bf-61be8ada7551) to fetch all of the details of that Identity, so that the script is able to continue with any relevant interactions for that user that may cause additional emails to be generated.
	`{`
	`"RecieveTime": 1585452543590,`
	`"MessageText": "This is a short message about your recent transaction for Order Number 927099.",`
	`"Serial": 1,`
	`"UID": "4c370969-2ceb-4d64-a1bf-61be8ada7551",`
	`"EmailAndFromDomain": "benjamin.micheli.1894@testdata.email|rosestorm.com",`
	`"Latency": 123.541`
	`}`

How can I automate login when it requires a Time Based MFA token, such as Google or Microsoft Authenticator? These Time Based tokens are based on RFC4226 and are very low cost to implement, as no security servers or software licencing is required, and the security is based on the one-time loading of a 'secret' into an App.	Watch this short video (TBA) that shows how to use a few simple calls to save a RFC4226 Time Based token against an email address and then retrieve the token based on that email. You can open up this Postman Collection (https://testdataservices.com.au/docs/Postman_MFA) and step through the same steps as shown in the video, to get a hands-on feel for how it works.
How can my CRM system exchange emails with Test Identities? CRM workflows are frequently initiated from a known or unknown Identity, but traditional testing has made such interactions difficult to include in automated tests.	Test Data Services contains millions of manufactured Identities that look very like real people, but none of the data is real except for the physical address. Each identity has an email address, and it is possible for any Identity to both SEND and RECEIVE emails, by making simple calls to an API endpoint. This allows inclusion of complex interactions with identities in a test environment by including real email interactions. For example a known (or unknown) identity could trigger a workflow from an inbound email, resulting in an email exchange with a test identity.