

NYL Postman API Testing Setup & Tests

AgentTermination Request Test

* These tests require an authorization token. Please refer to the [general Postman API instructions](#) if needed.

**If testing in Staging, use [Azure Portal NYL Interceptor page](#) to change the values.

1. It is preferred to test with an account that is set up like NYL from the hub (Ask Peter, Michael or Royce for assistance)
2. Elastic Search needs to be running.
3. Run the NYL Interceptor in Visual Studio and Utopia (Atlantis will replace Utopia eventually)
4. Add 3 profile items called NYLID, Marketer ID, and Worker-Type and add them to a profile.
5. The folder structure needs to be set up with a drawer of agents from a certain region, then a folder with the agent's name, and two subfolders of Clients and Prospects.



6. Set up or edit a user and add the new profile and add in a number to both Marketer and NYL ID's (my example is 98765)



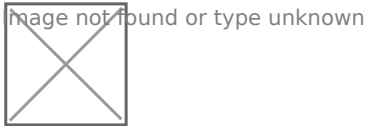
7. You will need the following database ID's:
 - Account ID: Find it in User Sessions URL
 - Profile Items ID's for Marketer ID: found in the URL of the profile item itself
 - NYLID: found in the URL of the profile item itself
8. When an agent terminates from their system, they will email us a marketer ID and email address. If they exist, it will locate the Marketer ID, then sets a purge governance 6 months from now on the agent's folder itself and applies to that item only.
9. In Visual Studio, open the local.settings.json and change the marketer ID and NYL ID to the ID's for your profile items.
10. Also change the environment URL if needed. I tested in local so I changed it to my local environment (only applicable in my environment):

"EnvironmentURL": "<https://localhost:44334/>",

"MarketerId": 397,

"NYLId": 395

11. Go to Postman API, and if needed, import the NYL Collection.json (Craig and Peter have copies or it can be found in a test plan ---Link coming soon)
12. Go to the Agent Termination folder, and test in the appropriate environment.
13. TBoth Marketer ID and NYL ID can be tested. For now, we will test with NYL ID. The NYL ID number you enter is not the DB ID obtained in step #6, it is the value you entered in the profile item itself in step # 5.



14. Send the request
15. Make sure status 200 is good is returned.
16. Make sure the response says something like this:

image.png

17. Return to Rubex and verify that a 6 month purge governance exists on the folder of the agent's name and applies to this item only.

image.png

18. Make sure that the subfolders and files are not inheriting the profile value. The Clients and Prospects folders should only have the existing NYL governance (if you have an NYL type of account created from the hub)

image.png


Undo - AgentTermination Request Test

1. Copy the body from the AgentTermination request and paste into the Undo - AgentTermination Request
2. Send the request.
3. Open Rubex, and check the governance on the agent's name folder that you previously ran a termination request again.
4. The 6 Month Purge Governance should be deleted

image.png

CreateMissing Request Test

1. Open Visual Studio and run the NYL Interceptor and Utopia (if running locally)

2. The .json template mentioned at the beginning of this article should include a folder called "CreateMissing" with three POST tests for Local, Staging and Production.
3. Choose the environment that you are testing.
4. Update or add the Authorization token to the test.
5. Post the following Body text below and update the accountId at the beginning and end with yours (found in Rubex - User Sessions) and update the userName and name and send the request. It will create a new user with those details in Rubex and will assign them as a full license user with these custom permissions:  and or type unknown

BODY TEST:

```
[
  {
    "batchObject": {
      "roleType": "2",
      "accountId": 90,
      "userName": "cbutler+NYLMAN@efilecabinet.com",
      "name": "IgnoreMe NYL MAN (cbutler+NYLMAN@efilecabinet.com)",
      "systemPermissions": {
        "users": {
          "enabled": false
        },
        "guests": {
          "enabled": false
        },
        "groups": {
          "enabled": false
        },
        "profiles": {
          "enabled": false
        },
        "nodeManagement": {
          "enabled": false
        },
        "templates": {
          "enabled": false
        },
        "systemSettings": {
          "enabled": false
        },
        "nodePermissionOverride": {
          "enabled": false
        },
        "licenseManagement": {
          "enabled": false
        },
        "retentionAdmin": {
          "enabled": false
        }
      }
    }
  }
]
```

```

    },
    "nodeExchange": {
      "enabled": true
    },
    "nodeExchangeManagement": {
      "enabled": false
    },
    "salesForceManagement": {
      "enabled": false
    },
    "workflow": {
      "enabled": false
    },
    "workflowManagement": {
      "enabled": false
    },
    "auditLogManagement": {
      "enabled": false
    },
    "retentionUnlock": {
      "enabled": false
    },
    "eSignatureManagement": {
      "enabled": false
    },
    "eSignature": {
      "enabled": false
    },
    "accessLinksManagement": {
      "enabled": false
    },
    "accessLinksUsage": {
      "enabled": false
    },
    "reporting": {
      "enabled": false
    },
    "accountID": 90
  }
}
}
]

```