Integrating with Duo for MFA (Multi-Factor Authentication)

Derek Pascarella
Updated 6 days ago

Unfollow

Applies To: Ayehu NG

Description

Cisco's **Duo** is a highly flexible and secure multi-factor authentication service that adds an extra layer of security when it comes to identity verification for members of an organization. It is used by large enterprises and governments around the world because it is well known for its high level of security. This tutorial details the usage of a custom integration workflow for **Ayehu NG**. Using this custom integration, you can leverage the security and convenience of **Duo**'s multi-factor authentication from right within your own workflows. Please note that by default, this workflow tells **Duo** to automatically choose the best mobile device and best method (push, SMS, etc.) to send the authorization request to the user.

Pre-Requisites

The following are required in order to integrate **Ayehu NG** and **Automic Engine**:

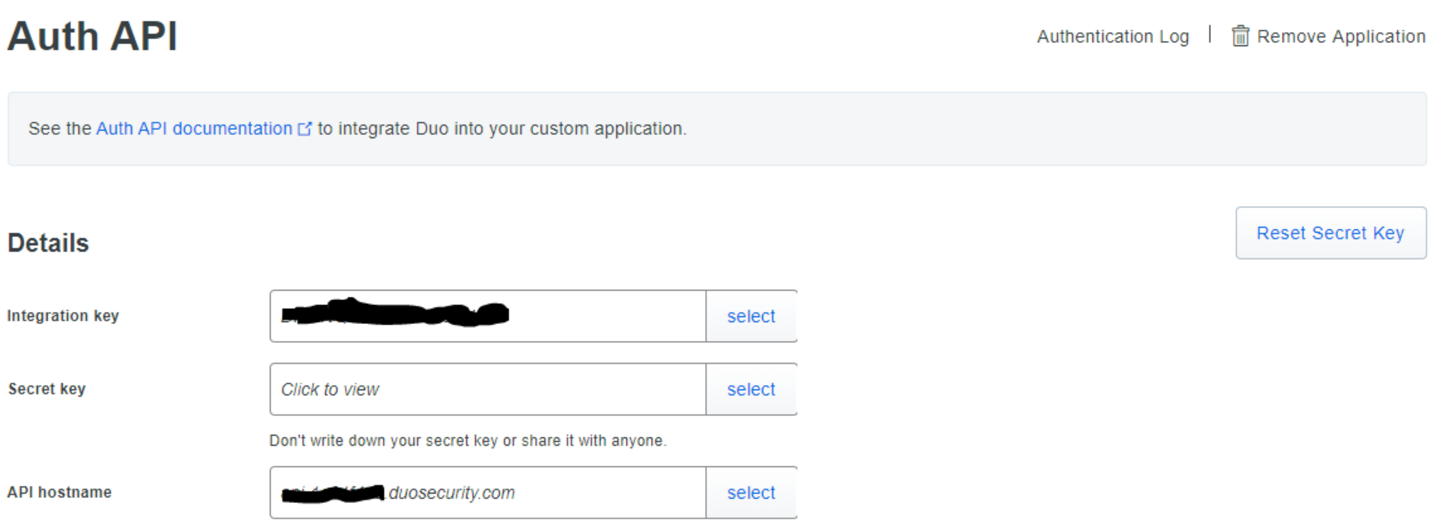
- Ayehu NG 1.3.x +
- A **Duo** account (https://signup.duo.com/) with its API enabled
- Knowledge of the **Duo** API (https://duo.com/docs/authapi#first-steps) if you want to extend or change the functionality of the custom integration workflow provided here
- The **Ayehu NG** workflows attached to this article for reference (**Duo - Authorize** and **Duo - Authorization Frontend Example** [Alt. GitHub Link])

Duo Preparation

If you don't already have a **Duo** account, create one using the steps below. If you already do have an account, skip Step 1. Steps 2 and 3 allow your **Duo** account to accept incoming API requests from external applications and services. If you already have the API configured on your account, then you may skip Steps 2 and 3, also.

1. Sign up for an account with **Duo** (https://signup.duo.com/).
2. Login to the **Duo Admin Panel** (https://admin.duosecurity.com/) and navigate to the **Applications** section.
3. Click **Protect an Application** and locate **Auth API** in the applications list. Click **Protect this Application** to get your **integration key**, **secret key**, and **API hostname** (see Getting Startedfor help).

After completing these steps (or if you already have an account configured with API access), ensure that you save the **integration key**, **secret key**, and **API hostname** in a safe place to reference later in this tutorial. They will appear like in the screenshot below.


Auth API

Authentication Log | Remove Application

See the Auth API documentation to integrate Duo into your custom application.

Details

Reset Secret Key

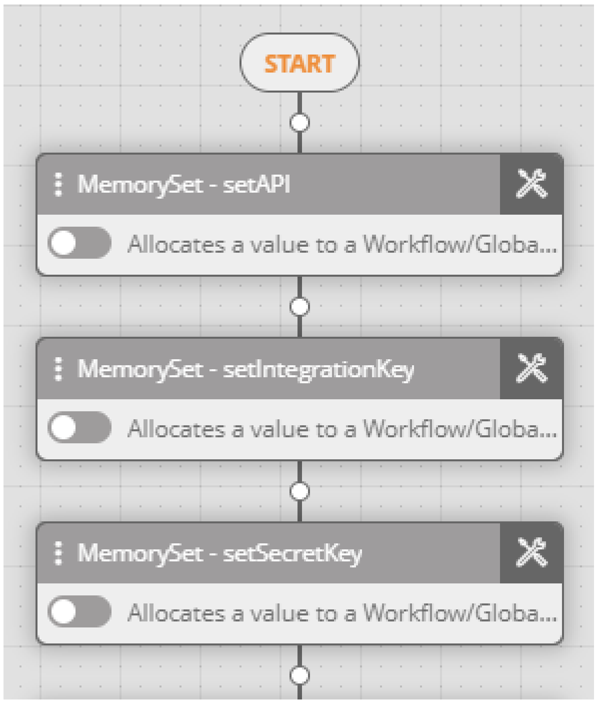
Integration key	[redacted]	select
Secret key	Click to view	select
	Don't write down your secret key or share it with anyone.	
API hostname	[redacted] duosecurity.com	select

Importing Custom Duo Integration Workflow

The **Duo - Authorize** workflow is a custom integration designed to function as a child workflow (read more https://support.ayehu.com/hc/en-us/articles/360008089093-Using-variables-from-a-parent-workflow-in-a-child-workflow-and-vice-versa). This means that the **Duo - Authorize**workflow does all of the heavy lifting for the communication with the **Duo** API, while all you need to worry about is calling on it using the **RunWorkflow** activity. An example of using **Duo - Authorize**as a child workflow can be seen in the attached workflow named **Duo - Authorization Frontend Example**. Ensure that both attached workflows are imported into your instance of **Ayehu NG**.

Configuring the "Duo - Authorize" Workflow

Open the **Duo - Authorze** workflow and locate the first three (3) **MemorySet** activities.


START
MemorySet - setAPI
Allocates a value to a Workflow/Globa...
MemorySet - setIntegrationKey
Allocates a value to a Workflow/Globa...
MemorySet - setSecretKey
Allocates a value to a Workflow/Globa...

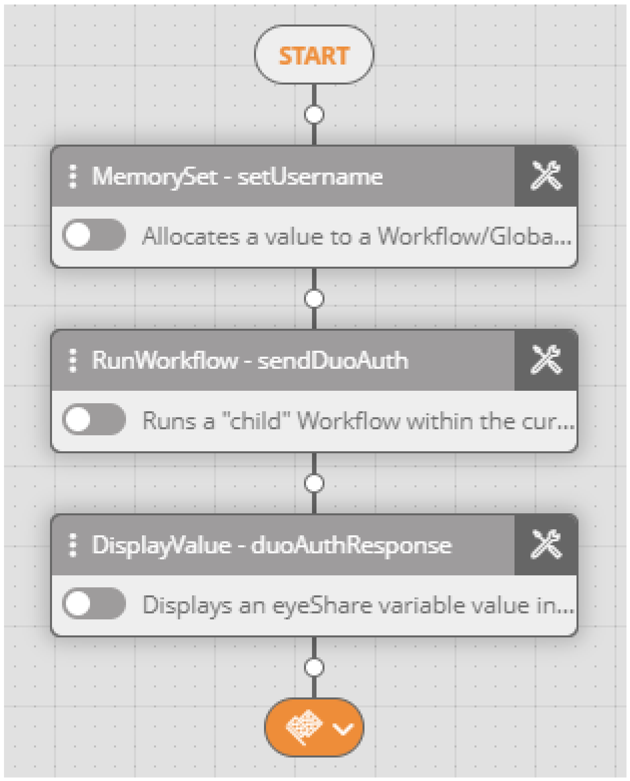
The following changes must be made:

- Configure the first **MemorySet** activity (named **setAPI**) to store the **API hostname** you were given from your **Duo** account.
- Configure the second **MemorySet** activity (named **setIntegrationKey**) to store the **integration key** you were given from your **Duo** account.
- Configure the third **MemorySet** activity (named **setSecretKey**) to store the **secret key** you were given from your **Duo** account.

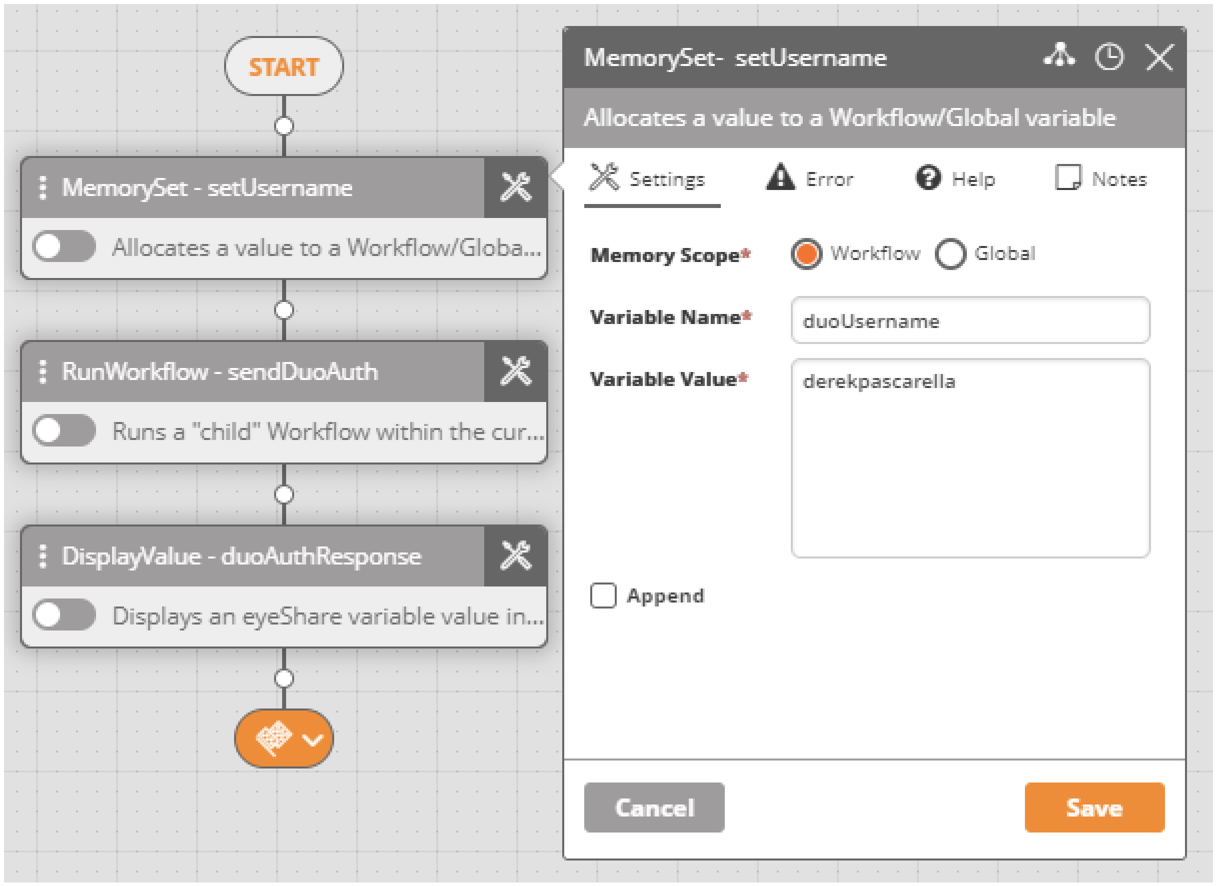
Once these changes have been made, save the **Duo - Authorize** workflow. Please note that this will only have to be done once, since this workflow will be called upon as a child workflow by other workflows whenever you need to send **Duo** an identity authorization request.

Using the Custom Duo Workflow (Frontend Example)

Open the **Duo - Authorization Frontend Example** workflow. This workflow functions as an example of a parent workflow created by you that calls upon the **Duo - Authorize** child workflow. Here, we see three (3) activities.

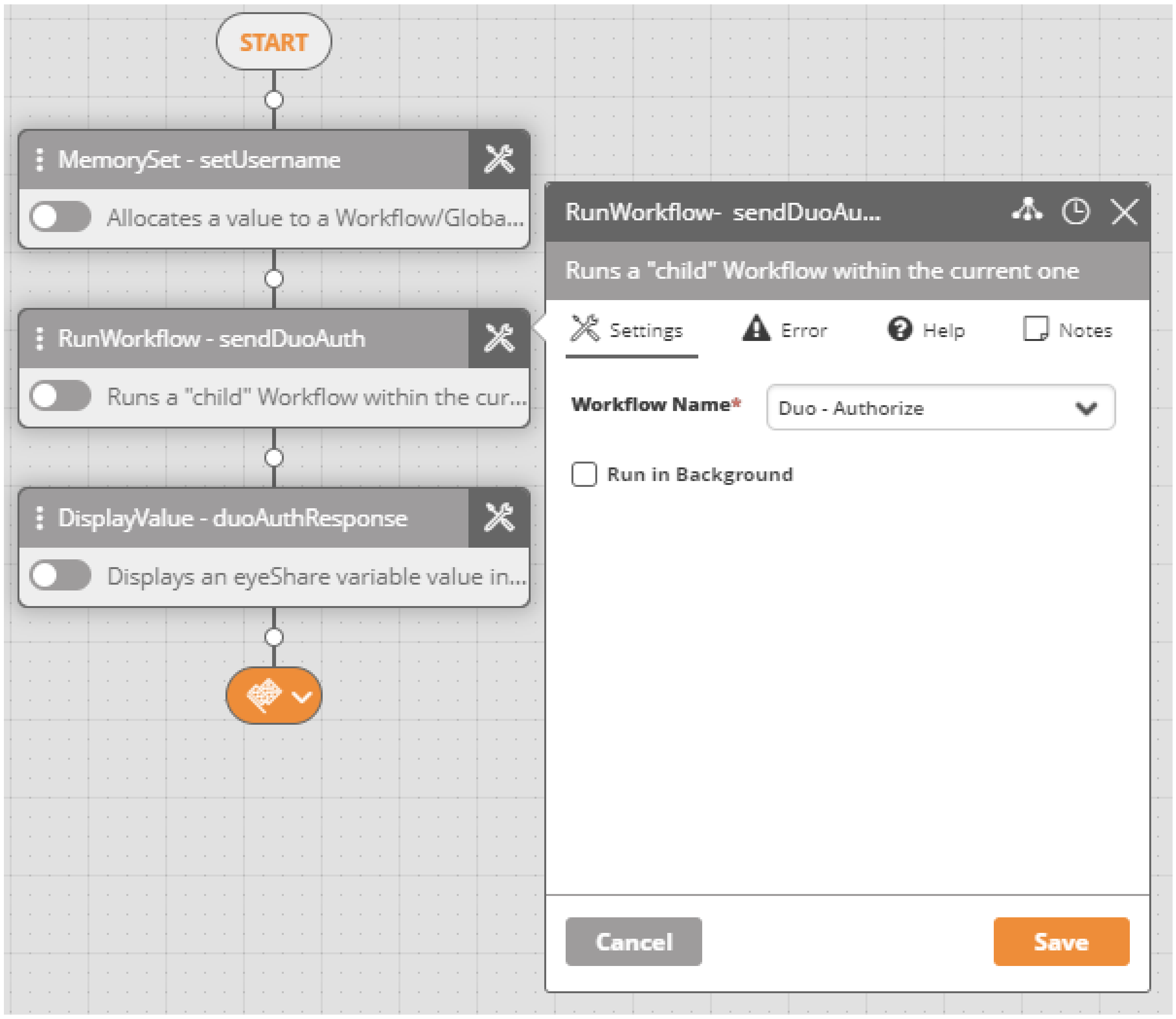

START
MemorySet - setUsername
Allocates a value to a Workflow/Globa...
RunWorkflow - sendDuoAuth
Runs a "child" Workflow within the cur...
DisplayValue - duoAuthResponse
Displays an eyeShare variable value in...

The first **MemorySet** activity stores the **Duo** username to which you are going to send an identity verification request. As seen in the screenshot below, this variable must be named **duoUsername** so that the child workflow (**Duo - Authorize**) knows how to find it. Below is a screenshot showing its configuration.


START
MemorySet - setUsername
Allocates a value to a Workflow/Globa...
RunWorkflow - sendDuoAuth
Runs a "child" Workflow within the cur...
DisplayValue - duoAuthResponse
Displays an eyeShare variable value in...
MemorySet- setUsername
Allocates a value to a Workflow/Global variable
Settings
Error
Help
Notes
Cancel
Save

Memory Scope* [✓ Workflow] Global
Variable Name* duoUsername
Variable Value* derekpascarella
☐ Append

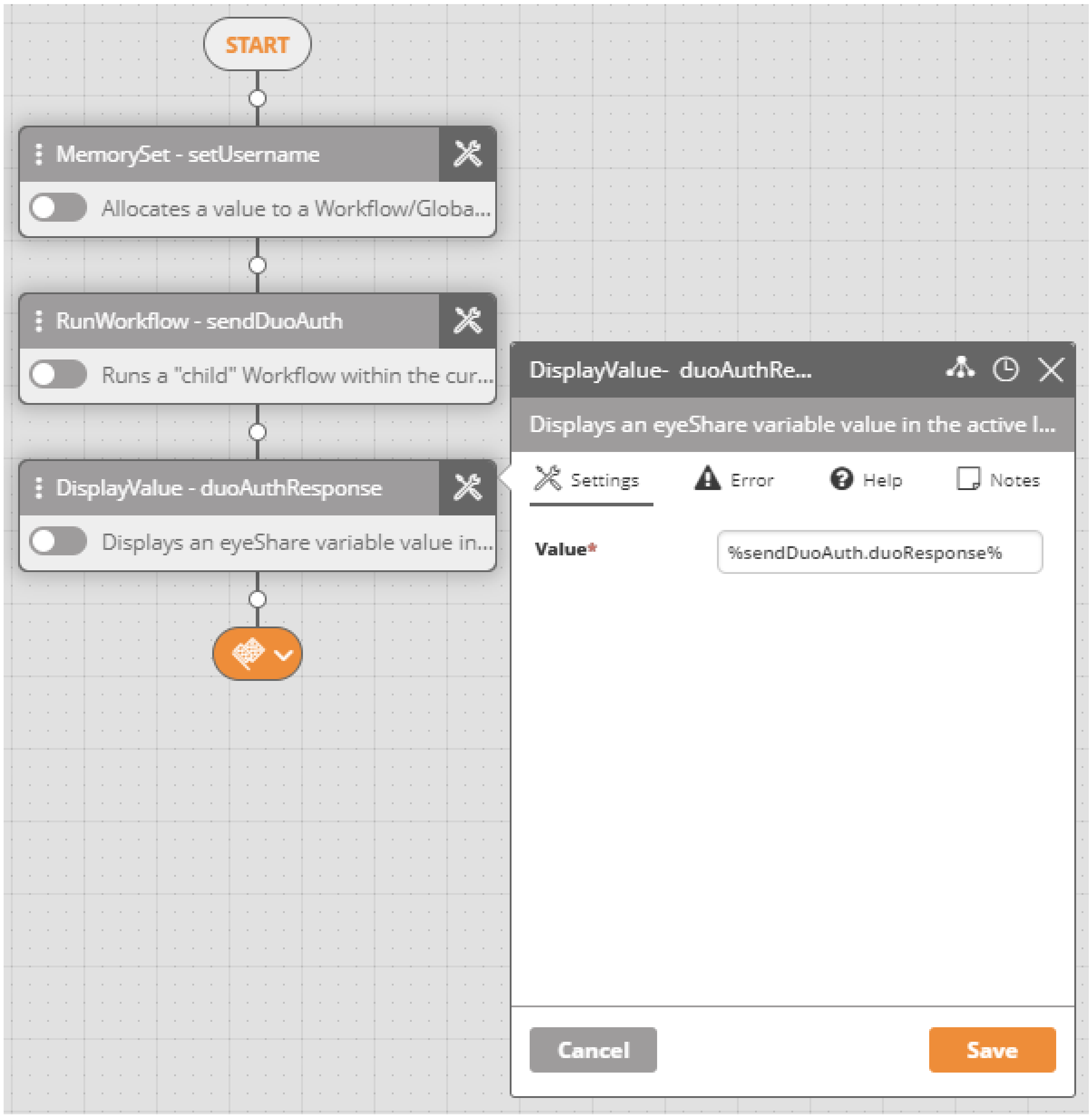
Next, we have a **RunWorkflow** activity. Open this activity's settings and re-select the **Duo - Authorize** workflow from the drop-down list labeled **Workflow Name**. This is to ensure that no workflow ID inconsistency issues arise after importing.


START
MemorySet - setUsername
Allocates a value to a Workflow/Globa...
RunWorkflow - sendDuoAuth
Runs a "child" Workflow within the cur...
DisplayValue - duoAuthResponse
Displays an eyeShare variable value in...
RunWorkflow- sendDuoAu...
Runs a "child" Workflow within the current one
Settings
Error
Help
Notes
Cancel
Save

Workflow Name* Duo - Authorize

☐ Run in Background

The third and final activity is a simple **DisplayValue** activity that shows you the variable that can be referenced for reading the result of the **Duo** identity verification request. As you can see, the **RunWorkflow** activity in this example parent workflow has been named **sendDuoAuth**. As a result, we reference its **duoResponse** variable using the full variable name **%sendDuoAuth.duoResponse%**.

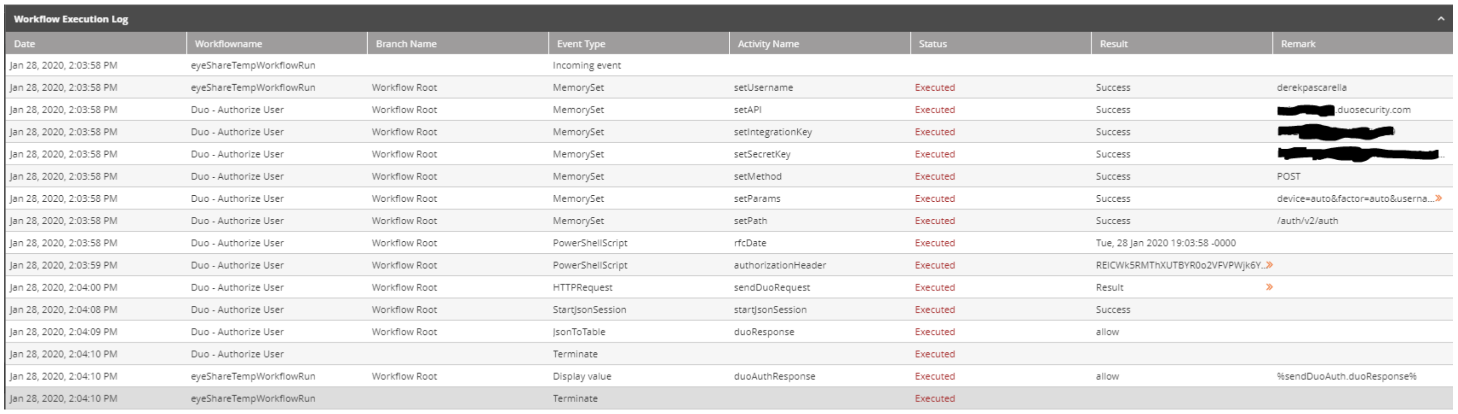

START
MemorySet - setUsername
Allocates a value to a Workflow/Globa...
RunWorkflow - sendDuoAuth
Runs a "child" Workflow within the cur...
DisplayValue - duoAuthResponse
Displays an eyeShare variable value in...
DisplayValue- duoAuthRe...
Displays an eyeShare variable value in the active l...
Settings
Error
Help
Notes
Cancel
Save

Value* %sendDuoAuth.duoResponse%

After setting the **Duo** username and ensuring the **RunWorkflow** activity is pointing at the **Duo - Authorize** child workflow, save this workflow and execute it.

Successful Execution of the Custom Duo Workflow

Below is a screenshot of the **Workflow Execution Log** after successfully executing the **Duo - Authorization Frontend Example** parent workflow, which calls on the child workflow **Duo - Authorize**.


Workflow Execution Log

Date	Workflowname	Branch Name	Event Type	Activity Name	Status	Result	Remark
Jan 28, 2020, 2:03:58 PM	eyeShareTempWorkflowRun		Incoming event				
Jan 28, 2020, 2:03:58 PM	eyeShareTempWorkflowRun	Workflow Root	MemorySet	setUsername	Executed	Success	derekpascarella
Jan 28, 2020, 2:03:58 PM	Duo - Authorize User	Workflow Root	MemorySet	setAPI	Executed	Success	[redacted].duosecurity.com
Jan 28, 2020, 2:03:58 PM	Duo - Authorize User	Workflow Root	MemorySet	setIntegrationKey	Executed	Success	[redacted]
Jan 28, 2020, 2:03:58 PM	Duo - Authorize User	Workflow Root	MemorySet	setSecretKey	Executed	Success	[redacted]
Jan 28, 2020, 2:03:58 PM	Duo - Authorize User	Workflow Root	MemorySet	setMethod	Executed	Success	POST
Jan 28, 2020, 2:03:58 PM	Duo - Authorize User	Workflow Root	MemorySet	setParams	Executed	Success	device=auto&factor=auto&userna...»
Jan 28, 2020, 2:03:58 PM	Duo - Authorize User	Workflow Root	MemorySet	setPath	Executed	Success	/auth/v2/auth
Jan 28, 2020, 2:03:58 PM	Duo - Authorize User	Workflow Root	PowerShellScript	rfcDate	Executed	Tue, 28 Jan 2020 19:03:58 -0000	
Jan 28, 2020, 2:03:59 PM	Duo - Authorize User	Workflow Root	PowerShellScript	authorizationHeader	Executed	REICWk5RMThXUTBYR0o2VFVPWjk6Y...»	
Jan 28, 2020, 2:04:00 PM	Duo - Authorize User	Workflow Root	HTTPRequest	sendDuoRequest	Executed	Result »	
Jan 28, 2020, 2:04:08 PM	Duo - Authorize User	Workflow Root	StartJsonSession	startJsonSession	Executed	Success	
Jan 28, 2020, 2:04:09 PM	Duo - Authorize User	Workflow Root	JsonToTable	duoResponse	Executed	allow	
Jan 28, 2020, 2:04:10 PM	Duo - Authorize User		Terminate		Executed		
Jan 28, 2020, 2:04:10 PM	eyeShareTempWorkflowRun	Workflow Root	Display value	duoAuthResponse	Executed	allow	%sendDuoAuth.duoResponse%
Jan 28, 2020, 2:04:10 PM	eyeShareTempWorkflowRun		Terminate		Executed		

Click image to view full-sized version.

As we can see, the **DisplayValue** activity from the **Duo - Authorization Frontend Example**workflow outputs "allow", as the user in this example verified the request on their mobile device. For identity verification requests that are rejected or that time-out, the