Reference Manual

Release 0.01

Alex Basin, Cina Shaykhian

Apr 26, 2021

CONTENTS

CHAPTER

ONE

OVERVIEW

This guide explains how to vary the authentication behaviour to Salesforce depending on the domain found in the username, by taking advantage of Apex handlers. By default, Salesforce does not have a built-in method to adapt the authentication requirement for different users. This complicates integration such as with Chatter, where external users may require a different form of authentication than the other Salesforce users.

In this guide, we configure Salesforce in a way that only specific users will use strong multi-factor authentication with SafeNet Trusted Access ("*STA*"), a leading-class cybersecurity platform to control in real-time access patterns.

- **Internal users** will login to Salesforce using *STA* as a 3rd party SAML SSO IdP, based on a specific list of domains `domainFilters` that are compared against the `Salesforce username`.
- For **all other users**, authentication is based on the basic `Salesforce password`.

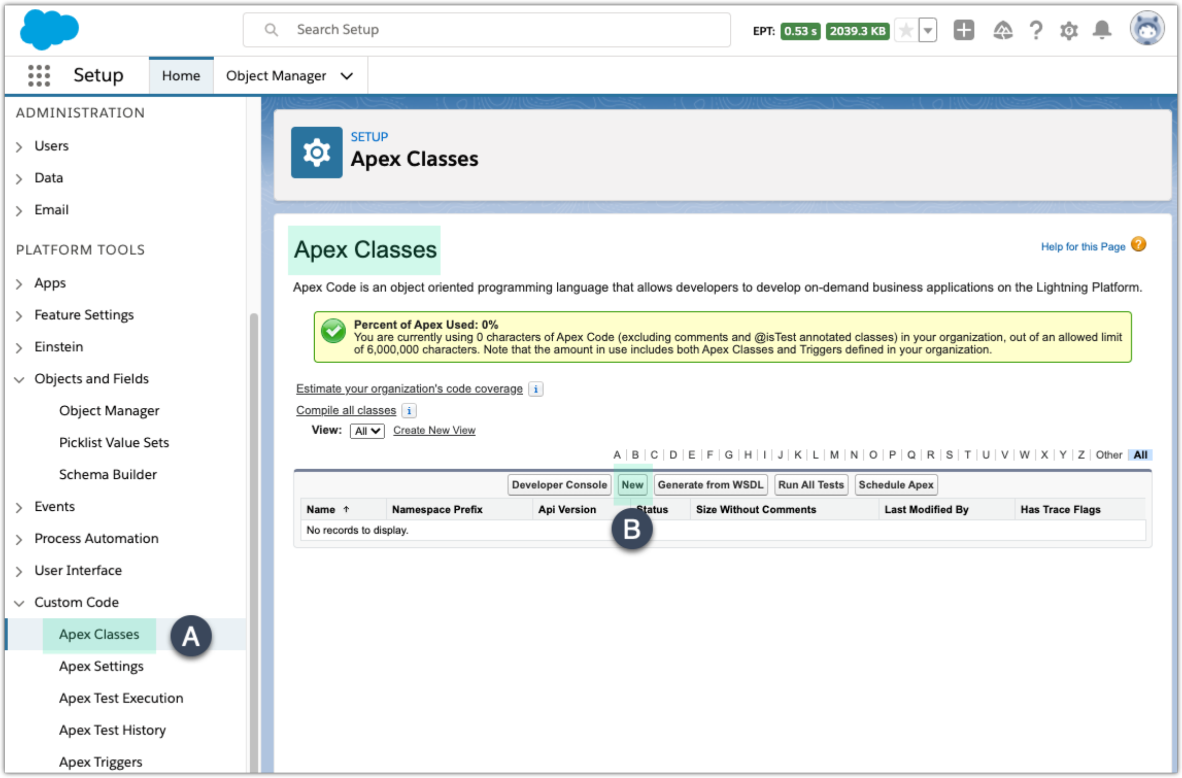
CHAPTER TWO

INSTRUCTIONS

2.1 Salesforce Configuration

Note: The steps provided are using the Salesforce Lightning experience. While the steps may slightly vary in the Classic experience, the overall procedure remains the same.

1. From your **Salesforce admin console**, search or navigate to *Apex Classes* and click `New`.



Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

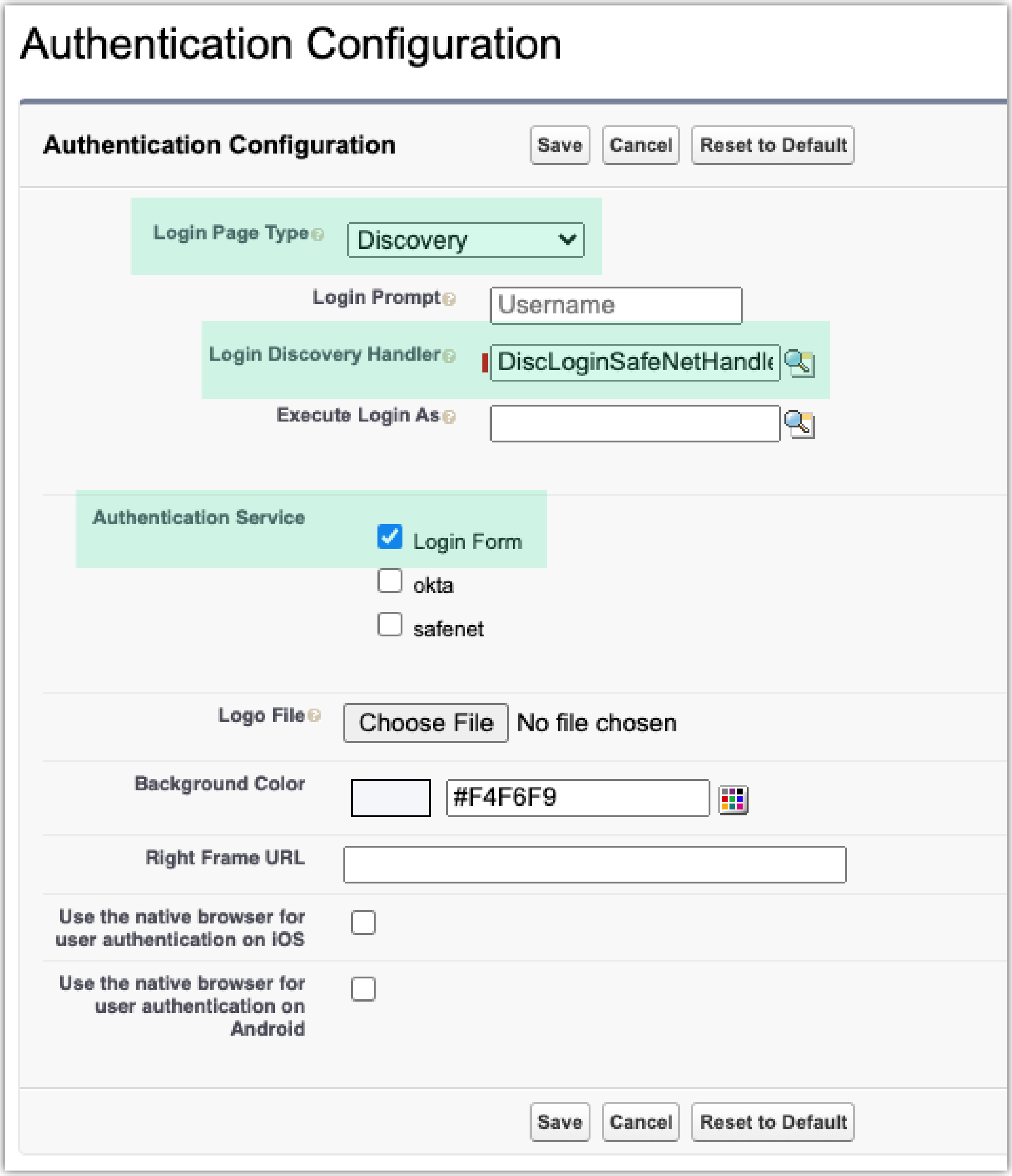
Percent of Apex Used: 0%
You are currently using 0 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Name ↑	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
No records to display.						

2. Adjust and paste the *Apex handler* code below, then click `Save`.

3. Enable this custom login discovery handler under *Company Settings* → *My Domain*.

    1. Under **Authentication Configuration** select the *Discovery Login Page* Type.
    2. For **Login Discovery Handler** select the `DiscLoginSafeNetHandler` handler from the list of Apex classes.
    3. Also confirm that `Login Form` is checked as the *Authentication Service*.



Authentication Configuration

Authentication Configuration

Login Page Type: Discovery

Login Prompt: Username

Login Discovery Handler: DiscLoginSafeNetHandle

Execute Login As:

Authentication Service:
☑ Login Form
□ okta
□ safenet

Logo File: Choose File No file chosen

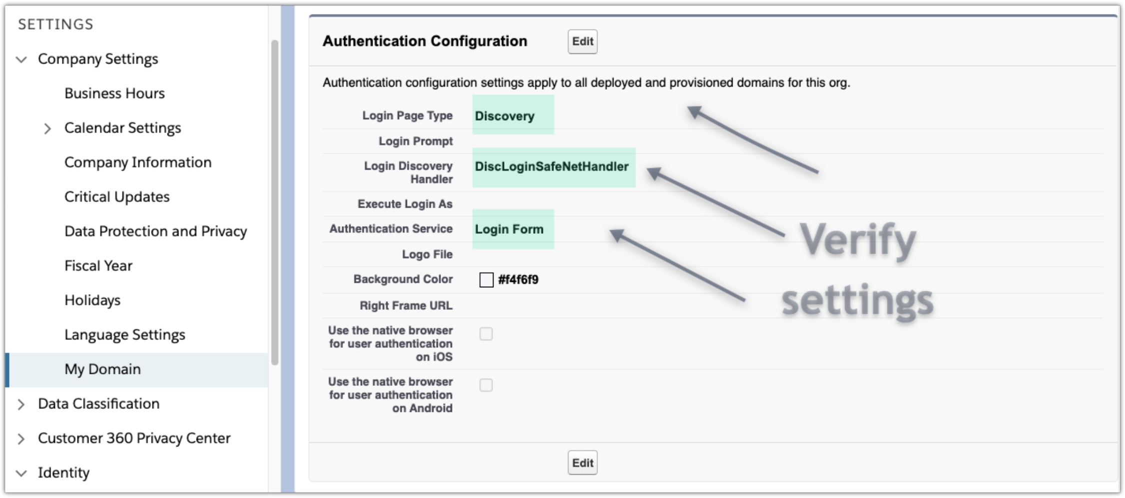
Background Color: #F4F6F9

Right Frame URL:

Use the native browser for user authentication on iOS □

Use the native browser for user authentication on Android □

    4. Click on `Save`.
    5. Verify your settings.



Authentication Configuration

Authentication configuration settings apply to all deployed and provisioned domains for this org.

Login Page Type	Discovery
Login Prompt	
Login Discovery Handler	DiscLoginSafeNetHandler
Execute Login As	
Authentication Service	Login Form
Logo File	
Background Color	□ #f4f6f9
Right Frame URL	
Use the native browser for user authentication on iOS	□
Use the native browser for user authentication on Android	□

4. **(Optional)** Enable the *Prevent login from https://login.salesforce.com* policy under *Company Settings → My Domain → Policies*. Complete this step after validating the solution if this policy is not already enabled.

2.1.1 Apex Handler

Note: The following handler which implements a custom Login Discovery page is in Salesforce preview.

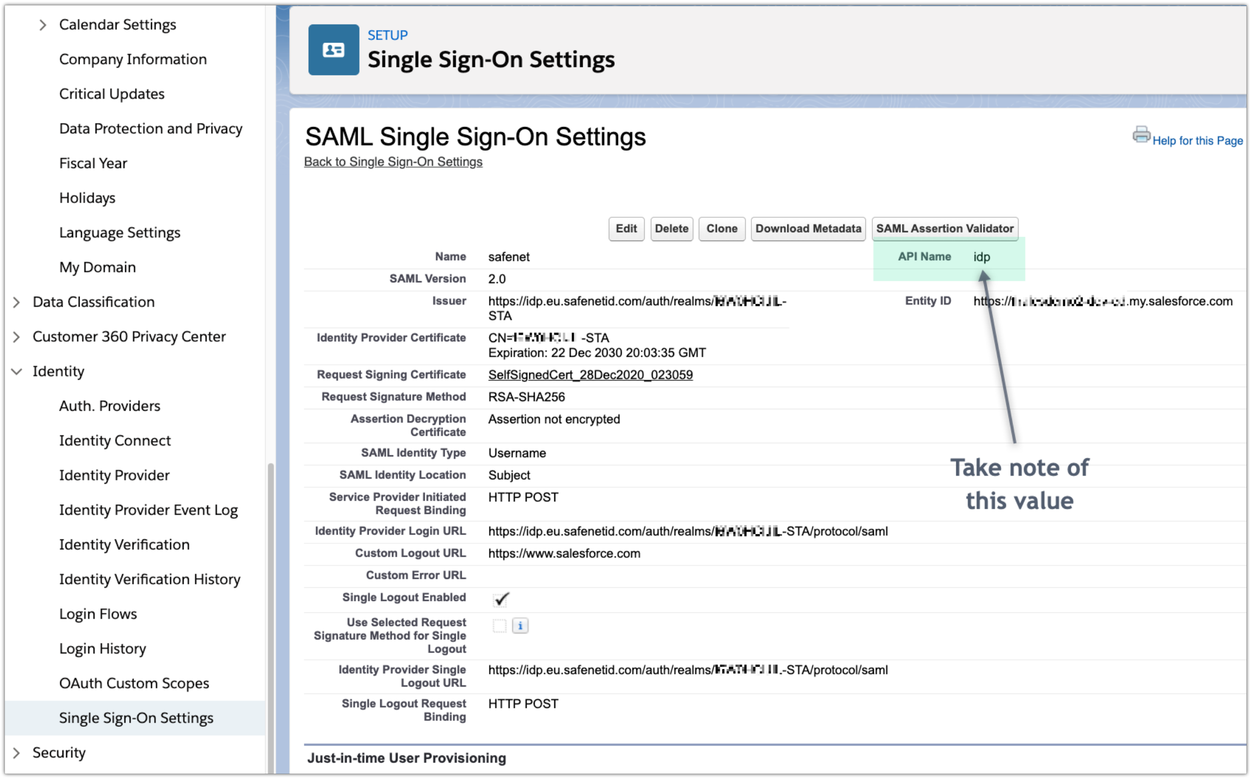
2.1.2 Parameters

The following parameters will need to be modified in the apex code to suit your environment:

`domainFilters` comma-separated list of domain names to be parsed from the Salesforce username for redirection to the *STA* IDP

`idpName` *API name* of the *STA* IDP in Salesforce

Tip: The *API name* can be found under *Identity → Single Sign-On Settings* by viewing the IDP config details (click on the *STA* IDP name).



SAML Single Sign-On Settings

Name	safenet	API Name	idp
SAML Version	2.0		
Issuer	https://idp.eu.safenetid.com/auth/realms/[redacted]-STA	Entity ID	https://[redacted].my.salesforce.com
Identity Provider Certificate	CN=[redacted] -STA Expiration: 22 Dec 2030 20:03:35 GMT		
Request Signing Certificate	SelfSignedCert_28Dec2020_023059		
Request Signature Method	RSA-SHA256		
Assertion Decryption Certificate	Assertion not encrypted		
SAML Identity Type	Username		
SAML Identity Location	Subject		
Service Provider Initiated Request Binding	HTTP POST		
Identity Provider Login URL	https://idp.eu.safenetid.com/auth/realms/[redacted]-STA/protocol/saml		
Custom Logout URL	https://www.salesforce.com		
Custom Error URL			
Single Logout Enabled	✓		
Use Selected Request Signature Method for Single Logout	□		
Identity Provider Single Logout URL	https://idp.eu.safenetid.com/auth/realms/[redacted]-STA/protocol/saml		
Single Logout Request Binding	HTTP POST		

Just-in-time User Provisioning

2.1.3 Code[1]

Refer to the definitions in the previous section to set the yellow highlighted parameters for your own environment.

```
// This Salesforce Login Discovery class enables adaptive authentication logic from the domain
// found in the Salesforce username (that is after an '@' symbol).
// e.g. domain.com if incoming username is xyz@domain.com
// Use Auth.DiscoveryCustomErrorException to throw custom errors which will be shown on login page.

global class DiscLoginSafeNetHandler implements Auth.MyDomainLoginDiscoveryHandler {

   global PageReference login(String identifier, String startUrl, Map<String, String> requestAttributes)
   {
     if (identifier != null) {
       // Search for user by username
       List<User> users = [SELECT Id, Username FROM User WHERE Username = :identifier AND IsActive = TRUE];
       if (!users.isEmpty() && users.size() == 1) {
         return discoveryResult(users[0], startUrl, requestAttributes);
       } else {
```

(continues on next page)

[1] Salesforce code example https://developer.salesforce.com/docs/atlas.en-us.apexref.meta/apexref/apex_interface_Auth_MyDomainLoginDiscoveryHandler.htm

(continued from previous page)

```
         throw new Auth.LoginDiscoveryException('No unique user found. User count=' + users.size());
       }
     }
     throw new Auth.LoginDiscoveryException('Invalid Identifier');
   }

   private PageReference getSsoRedirect(User user, String startUrl, Map<String, String> requestAttributes)
   {
     // API name of the SAML IDP
     String idpName = 'idp';

     // Look up if the user should log in with IDP and return the URL to initialize SSO.
     SamlSsoConfig SSO = [select Id from SamlSsoConfig where DeveloperName=:idpName limit 1];

     // To get the URL for a My Domain subdomain, you can pass null in the communityURL parameter.
     String ssoUrl = Auth.AuthConfiguration.getSamlSsoUrl(null, startUrl, SSO.Id);
     return new PageReference(ssoUrl);
   }

   private PageReference discoveryResult(User user, String startUrl, Map<String, String> requestAttributes)
   {
     String domain = user.Username.split('@').get(1);

     // Modify the list of domains between the brackets. For single domain, do not include a comma separator.
     List<String> domainFilters = new List<String>{'thalesdemo.ml', 'thalesgroup.com'};

     PageReference ssoRedirect = null;
     try { ssoRedirect = getSsoRedirect(user, startUrl, requestAttributes); }
     catch(Exception e) { ssoRedirect = null; }

     if(ssoRedirect != null && domainFilters.contains(domain)) {
       return ssoRedirect;
     } else {
       return Auth.SessionManagement.finishLoginDiscovery(Auth.LoginDiscoveryMethod.password, user.Id);
     }
   }
}
```

CHAPTER

THREE

VALIDATING THE SOLUTION

With the above `domainFilters`, when the domain in the username is either @*thalesdemo.ml* or @*thalesgroup.com*, users will be redirected to the STA IDP for authentication to subsequently access Salesforce.

Other users will login with the regular Salesforce password. This is achieved by changing the login experience on Salesforce. That is, instead of entering both username and password on the first login window, the form now only asks for the username.

1. **Login** with `admin@thalesdemo.ml`
   Result: redirected to STA based on domain filter (long video)

2. **Login** with `sales@thalesgroup.com`
   Result: redirected to STA based on domain filter (short video)

3. **Login** with `test-user@mailinator.com`
   Result: prompted for Salesforce password

CHAPTER

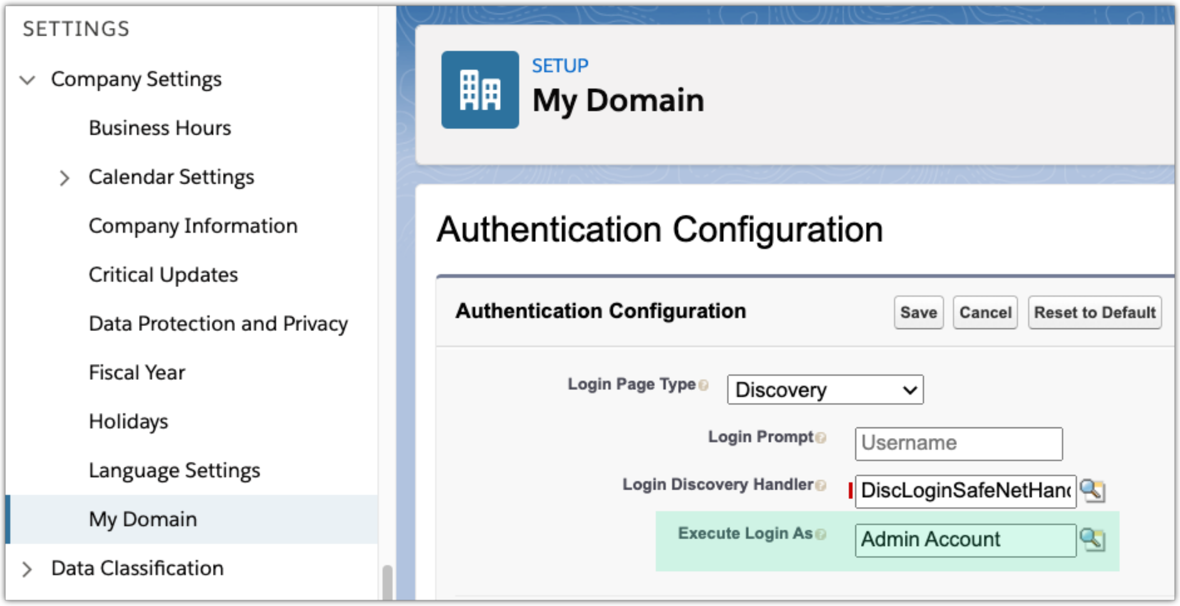
FOUR

TROUBLESHOOTING

You can debug apex classes with the `System.Debug()` statement. In order to capture those logs, you first need to:

1. Return to **Authentication Configuration** in *Company Settings* → *My Domain* and click `Edit`.

2. For **Execute Login As** select an administrator account and `Save` the settings.



Authentication Configuration

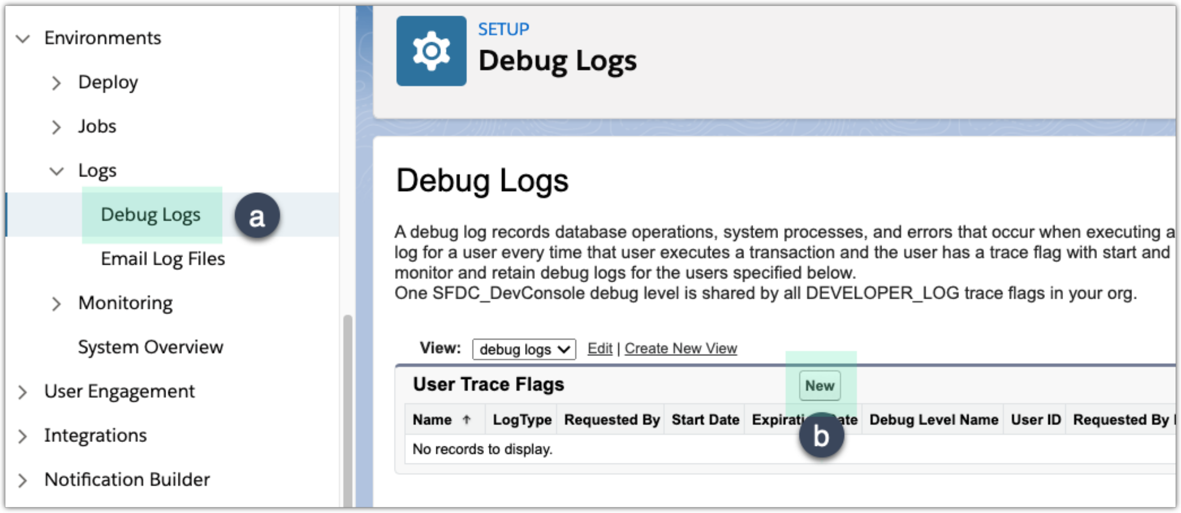
Login Page Type: Discovery

Login Prompt: Username

Login Discovery Handler: DiscLoginSafeNetHan(

Execute Login As: Admin Account

3. Navigate to *Platform Tools* → *Environments* → *Logs* → *Debug Logs* and click `Edit`.



Debug Logs

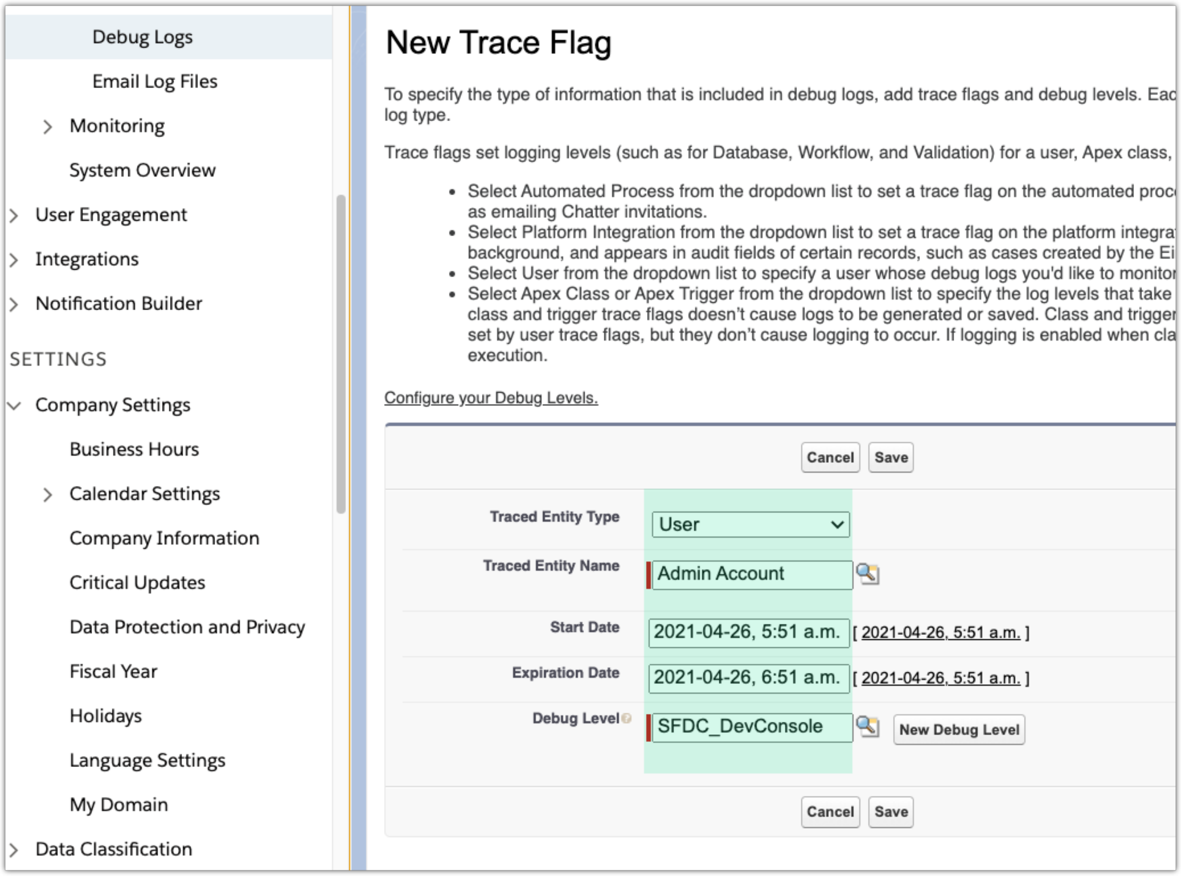
A debug log records database operations, system processes, and errors that occur when executing a log for a user every time that user executes a transaction and the user has a trace flag with start and monitor and retain debug logs for the users specified below.
One SFDC_DevConsole debug level is shared by all DEVELOPER_LOG trace flags in your org.

User Trace Flags

Name ↑	LogType	Requested By	Start Date	Expirati ate	Debug Level Name	User ID	Requested By I
No records to display.							

4. Configure the trace:

Property	Value
Trace Entity Type	Select **User**
Traced Entity Name	Name of the admin account from the previous step
Start Date	Start time for the trace logs
Expiration Date	Stop time for the trace logs
Debug Level	Set or create a debug level (or higher) for *Apex Code* category



New Trace Flag

To specify the type of information that is included in debug logs, add trace flags and debug levels. Eac
log type.

Trace flags set logging levels (such as for Database, Workflow, and Validation) for a user, Apex class,

- Select Automated Process from the dropdown list to set a trace flag on the automated proc
as emailing Chatter invitations.
- Select Platform Integration from the dropdown list to set a trace flag on the platform integra
background, and appears in audit fields of certain records, such as cases created by the Ei
- Select User from the dropdown list to specify a user whose debug logs you'd like to monito
- Select Apex Class or Apex Trigger from the dropdown list to specify the log levels that take
class and trigger trace flags doesn't cause logs to be generated or saved. Class and trigge
set by user trace flags, but they don't cause logging to occur. If logging is enabled when cla
execution.

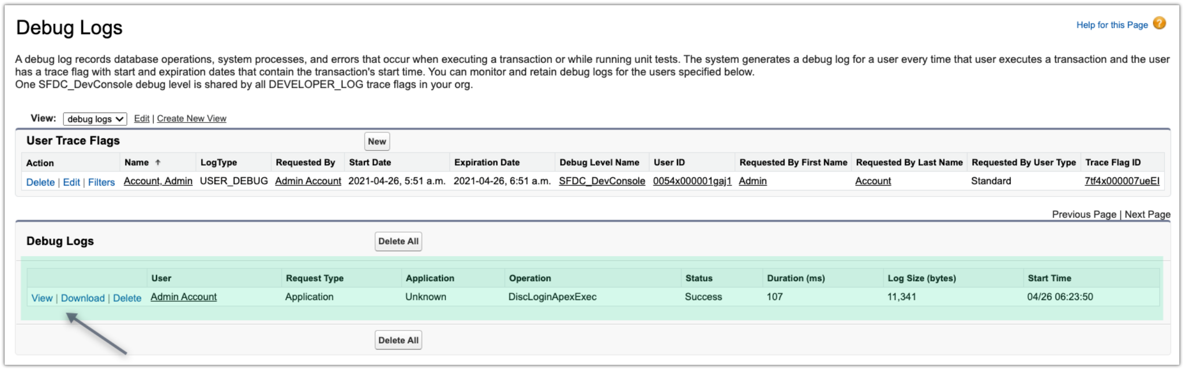
Configure your Debug Levels.

Traced Entity Type	User
Traced Entity Name	Admin Account
Start Date	2021-04-26, 5:51 a.m. [2021-04-26, 5:51 a.m.]
Expiration Date	2021-04-26, 6:51 a.m. [2021-04-26, 5:51 a.m.]
Debug Level	SFDC_DevConsole

5. Modify the Apex Class Handler to add wherever desired `System.Debug()` statements.

6. Login to Salesforce (using the Discovery page) to generate logs.

7. Go back to *Platform Tools* → *Environments* → *Logs* → *Debug Logs* to view or download the logs.



Debug Logs

A debug log records database operations, system processes, and errors that occur when executing a transaction or while running unit tests. The system generates a debug log for a user every time that user executes a transaction and the user has a trace flag with start and expiration dates that contain the transaction's start time. You can monitor and retain debug logs for the users specified below.
One SFDC_DevConsole debug level is shared by all DEVELOPER_LOG trace flags in your org.

View: debug logs Edit | Create New View

User Trace Flags

Action	Name ↑	LogType	Requested By	Start Date	Expiration Date	Debug Level Name	User ID	Requested By First Name	Requested By Last Name	Requested By User Type	Trace Flag ID
Delete | Edit | Filters	Account, Admin	USER_DEBUG	Admin Account	2021-04-26, 5:51 a.m.	2021-04-26, 6:51 a.m.	SFDC_DevConsole	0054x000001gaj1	Admin	Account	Standard	7tf4x000007ueEI

Previous Page | Next Page

Debug Logs

	User	Request Type	Application	Operation	Status	Duration (ms)	Log Size (bytes)	Start Time
View | Download | Delete	Admin Account	Application	Unknown	DiscLoginApexExec	Success	107	11,341	04/26 06:23:50

Contact

If you have any remarks about this guide, please don’t hesitate to contact us directly !