

# 在SSO模式下将第三方小工具与Finesse集成

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[SSO模式交互基本模型的解释](#)

[配置gadgets.io.ma请求SSO和NONSSO模式](#)

## 简介

本文档介绍在系统处于单点登录(SSO)模式时，将第3方小工具与Finesse集成所需的内容。还给出了非SSO模式的示例。

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- 思科Finesse
- SSO
- Finesse第三方小工具

### 使用的组件

本文档中的信息基于以下软件和硬件版本：

- 思科Finesse版本11.6
- SSO
- 第三方小工具
- 第三方REST服务。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

## 背景信息

这些是代理尝试登录并使用SSO或NONSSO进行身份验证时的初始步骤。

第二步介绍在SSO和NONSSO情况下，成功进行身份验证后需要考虑的事项。

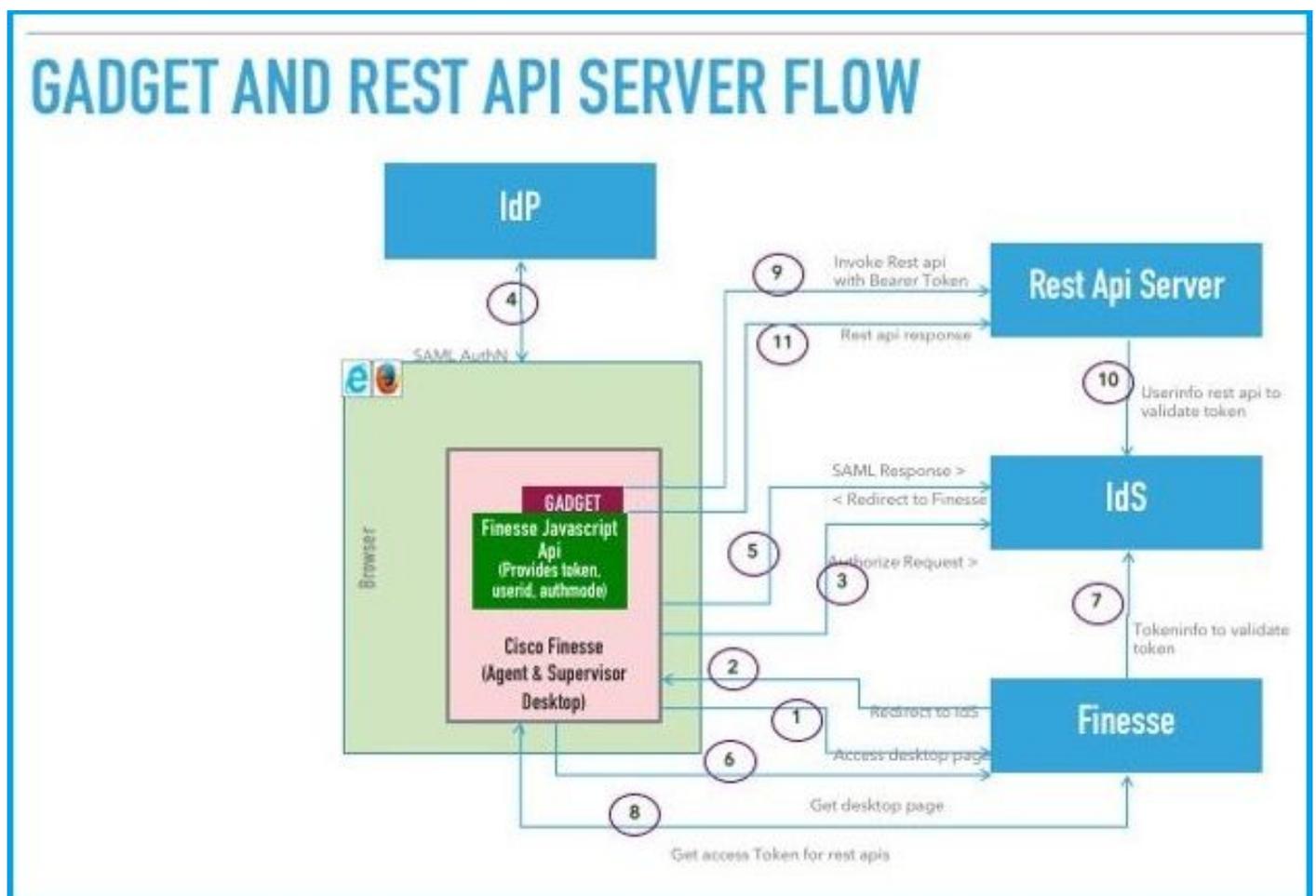
1. 在桌面登录时，Finesse会检测系统身份验证模式(SSO/NONSSO)，并且根据身份验证模式显

示相应的登录页。在SSO模式下，用户将看到IDP登录页；在NONSSO模式下，用户将看到Finesse登录页。

- 身份验证成功后，所有请求都会根据系统身份验证模式进行身份验证。对于SSO部署，对Finesse的所有请求都将访问令牌作为请求报头的一部分。令牌会针对IDP服务器进行验证，以成功进行身份验证。但是，对于向第三方Web服务的请求，必须根据由第三方Web服务实施的身份验证方案设置身份验证报头。在NONSSO部署中，所有请求都传送带Basic Auth报头和base64编码的用户名和密码。在这种情况下，所有请求都会根据Finesse本地数据库进行验证。

## SSO模式交互基本模型的解释

此图显示了系统处于SSO模式时第三方小工具Finesse、IDS和第三方REST服务之间交互的基本模型。



图像

下面是图中所示每个步骤的说明。

1. 座席/主管访问Finesse桌面URL。(例如：<https://finesse.com:8445/desktop>)
2. Finesse检测身份验证模式为SSO，并将浏览器重定向到IDS。
3. 浏览器向IDS发送重定向授权请求。此时，IDS会检测用户是否具有有效的访问令牌。如果用户没有有效的访问令牌，则IDS会重定向到身份提供程序(IdP)。
4. 如果请求重定向到IdP，IdP将提供用于验证用户的登录页。
5. 来自IdP的SAML断言被发送到IDS，IDS将重定向回Finesse桌面。

6. 浏览器执行Finesse桌面页面的GET。
7. Finesse从IDS获取具有SAML身份验证代码的访问令牌。
8. 桌面获取用于验证后续REST API的访问令牌。
9. 第三方小工具加载到桌面，并使用auth-header中的访问令牌（承载）调用第三方REST API。
10. 第3方REST服务使用IDS验证令牌。
11. 第3方REST响应返回到小工具。

## 配置gadgets.io.ma请求SSO和NONSSO模式

步骤1.对于通过Shindig进行的Finesse REST API调用，小工具需要在gadgets.io.makeRequest标头中添加“承载”授权标头。

步骤2.小工具需要对所有REST请求进行本地gadgets.io.makeRequest调用，必须在请求参数内设置授权报头。

对于非SSO部署，这是身份验证报头。

```
"Basic " + base64.encode(username : password)
```

对于SSO部署，这是Auth报头。

```
"Bearer " + access_token
```

可以从finesse.gadget.Config对象检索访问令牌。

```
access_token = finesse.gadget.Config.authToken
```

必须将新授权报头添加到请求参数。

```
params[gadgets.io.RequestParameters.HEADERS].Authorization = "Basic " + base64.encode(username : password);
```

```
params[gadgets.io.RequestParameters.HEADERS].Authorization = "Bearer " + access_token;
```

步骤3.实用程序方法getAuthHeaderString已添加到utilities.Utilities中。此实用程序方法将config对象作为参数并返回授权报头字符串。小工具可以使用此实用程序方法在请求参数中设置授权报头。

```
params[gadgets.io.RequestParameters.HEADERS].Authorization=  
finesse.utilities.Utilities.getAuthHeaderString(finesse.gadget.config);
```

**注意：**对于发往第三方Web服务的API请求，必须根据由第三方Web服务实施的身份验证方案设置身份验证报头。小工具开发人员可以自由使用基于基本身份验证或承载令牌的身份验证或他们选择的任何其他身份验证机制。