Using Parameters To Dictate Which Login Page To Show (OpenID Connect)

- Introduction
- How To
    - Setting parameters on the URL
    - The logic to display different login pages based on parameters
- Results

Introduction

This article discusses a way to display different login pages by passing parameters set on the URL. The requirement that will be used as an example is to have two different URLs of the same login page to display for admin users and non-admin users. This article will be using a CRM app as an example.

How To

Setting parameters on the URL

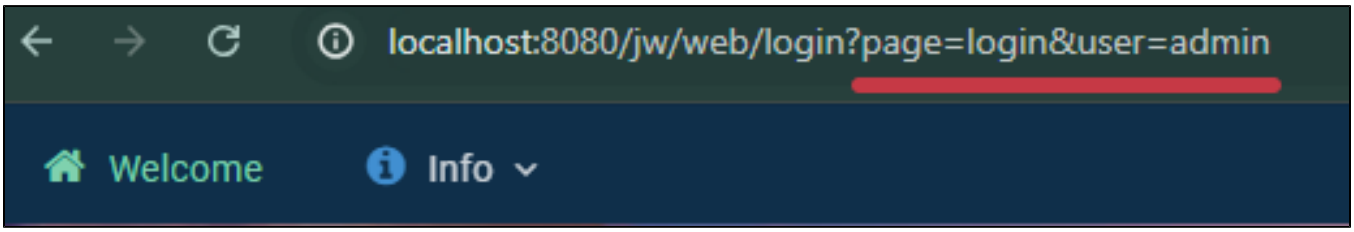

Figure 1: Login page URL with the parameters set for admin view.

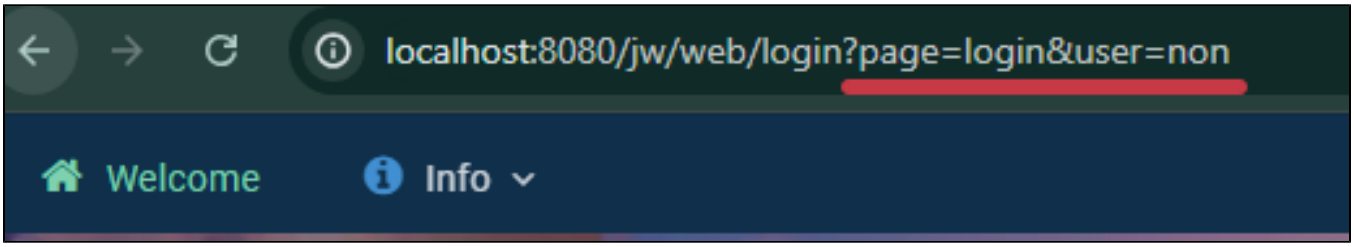

Figure 2: Login page URL with the parameters set for non-admin view.

Parameters:

- page=login
- user=admin or user=non

The logic to display different login pages based on parameters

To have different results based on the parameters, write a JavaScript code into the **UI Builder > Settings > Configure Layout > Advanced,** scroll down until the Custom JavaScript section.

Figure 3: Custom JavaScript section on UI Builder Advanced Settings.

The following script shows the different login pages based on the page and user parameters using JS:

```
$(document).ready(function(){
    const searchParams = new URLSearchParams(window.location.search);
    var page = searchParams.get('page');
    var userType = searchParams.get('user');

    if (page == 'login' && userType == 'admin') {
        $('#openIDLogin').remove(); //"openIDLogin" is the id of the cloud login button element
    } else if (page == 'login' && userType != 'admin') {
        $('#openIDLogin').appendTo('#loginForm'); //move button above table element
        $('table').remove(); //remove login form table
    }
});
```

This script takes in the parameters - if it is a login page and the user type is "admin", then remove the openID login button. Else if the type is not "admin", then move the openID login button and remove the login form.

A Custom CSS is also written for the cloud login button to keeps its styling after moving the element:

```
#openIDLogin {
    background: blue;
    padding: 10px 20px;
    border-radius: 3px;
    text-align: left;
    display: block;
    width: fit-content;
    color: white;
    margin: auto;
}
#icon {
padding-right: 20px;
}
```

Results

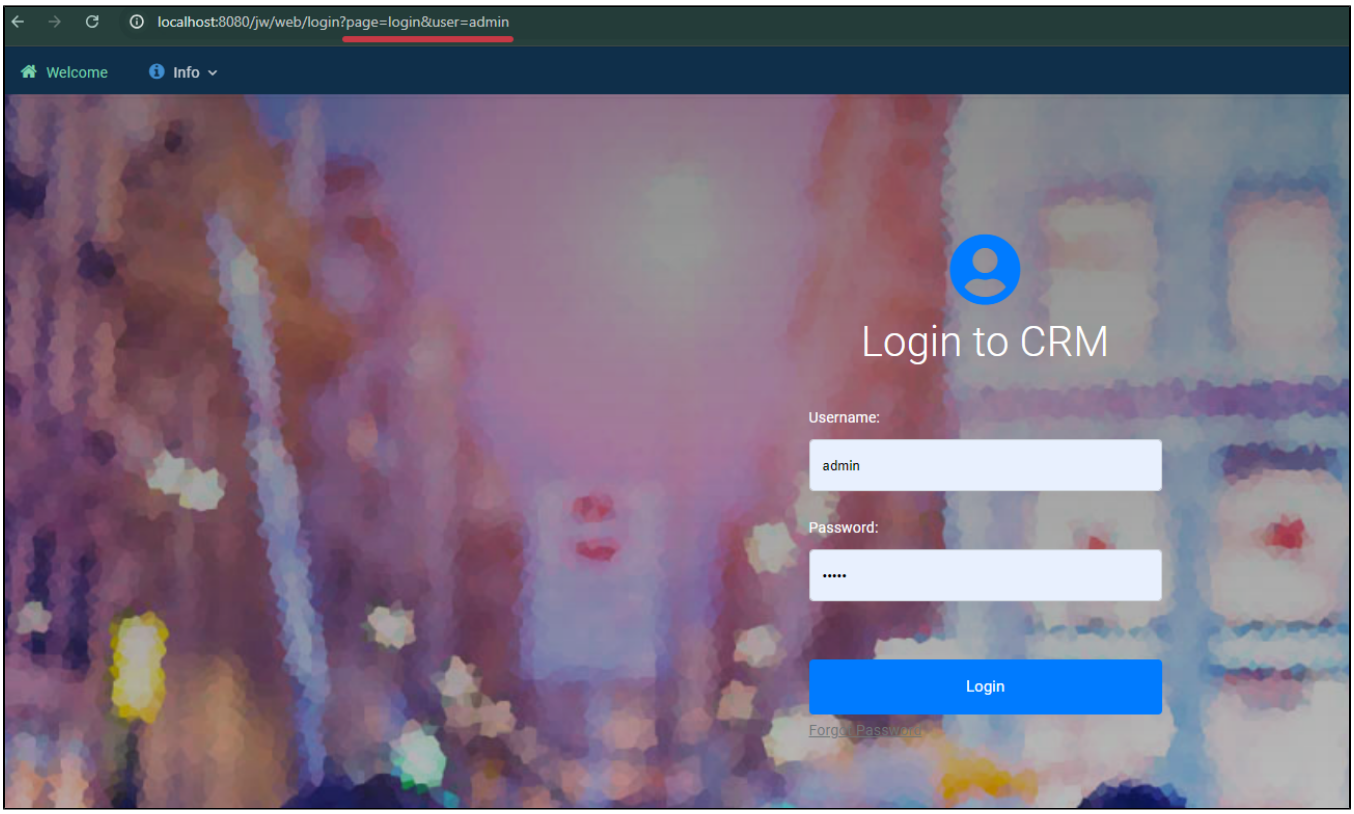

Figure 4: Admin login page, with username and password fields displayed.

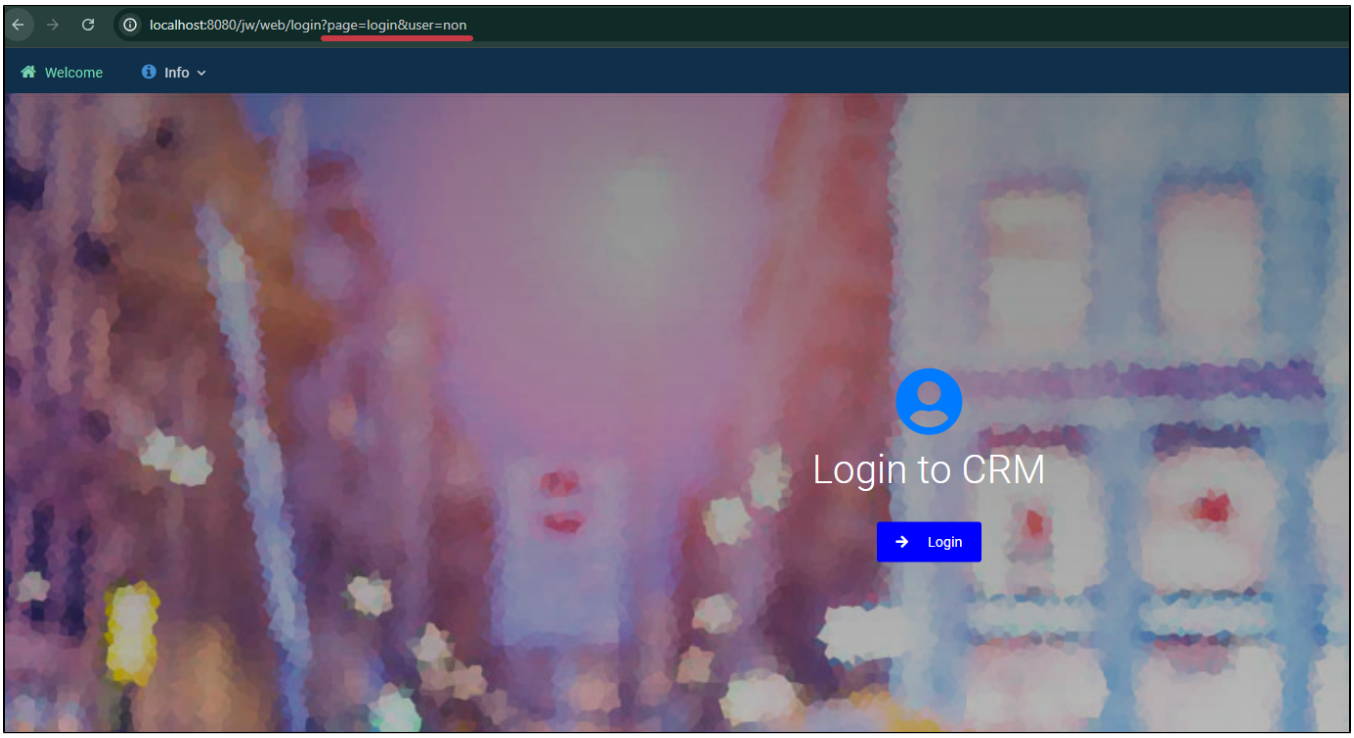

Figure 5: Non-admin login page, with only the cloud login button displayed.