Thank you for choosing the Free Auth0 plan. You have 21 days left in your trial to experiment with features that are not in the Free plan. If you want to continue using these features then provide your billing information.

Default App

Quick Start | Settings | Addons | Connections

Client ID: ibHLd5aHhBKPpZ8FCNaPM5IfimgN7WVJ

Documentation > Single Page App > jQuery

jQuery Tutorial

You can get started by either downloading the seed project or if you would like to add Auth0 to an existing application you can follow the tutorial steps.

System Requirements

This tutorial and seed project have been tested with the following:

- NodeJS 4.3
- Jquery 2.1.1

Download a sample project configured with your Auth0 API Keys.

GET SEED PROJECT

If you have an existing application, follow the steps below.

Before Starting

Please remember that for security purposes, you have to register the URL of your app on the Application Settings section as the callbackURL.

1. Add the Auth0 scripts and set the viewport

Add the code below to the `index.html` file to include Auth0's jQuery module and its dependencies and set the viewport:

```
<!-- Auth0 lock script -->
<script src="//cdn.auth0.com/js/lock-9.1.min.js"></script>

<!-- Setting the right viewport -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalab
```

2. Configure the Auth0Lock

Configure Auth0Lock with your `client-ID` and `domain` :

```
var lock = null;
$(document).ready(function() {
   lock = new Auth0Lock('ibHLd5aHhBKPpZ8FCNaPM5IfimgN7WVJ', 's3392244.auth0.com');
});
```

3. Implement the login

To implement the login, call the `.show()` method of Auth0's `lock` instance when a user clicks the login button. **Note:** This implementation uses Lock's redirect mode.

```
var userProfile;

$('.btn-login').click(function(e) {
  e.preventDefault();
  lock.show({ authParams: { scope: 'openid' } }); //Details: https://auth0.com/docs/scopes
});
```

```
<!-- ... -->
<input type="submit" class="btn-login" />
<!-- ... -->
```

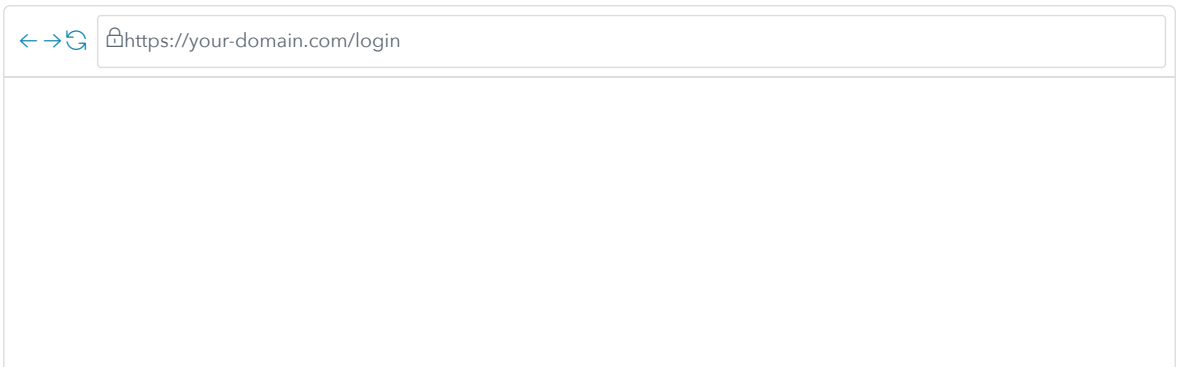
To discover all the available arguments for `lock.show`, see .show([options, callback]).

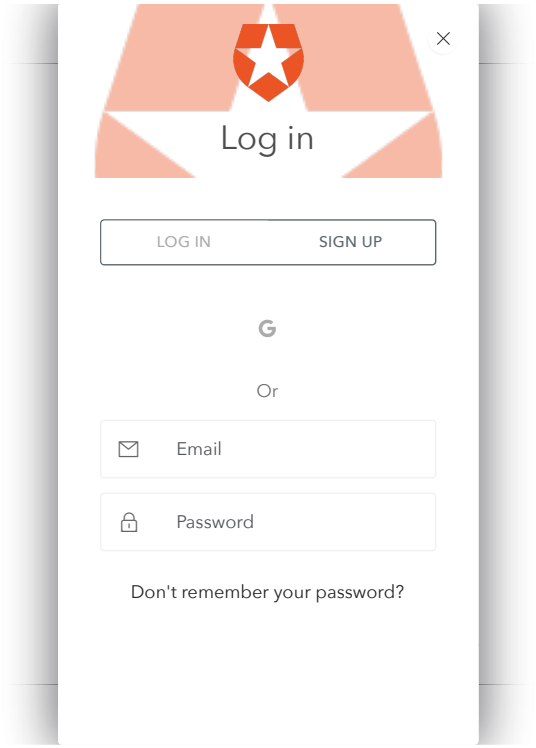
After authentication, Auth0 will redirect the user back to your application with an identifying token as a hash parameter of `window.location`. Use `lock.parseHash` to parse the hash and create the token. This token is used to retrieve the user's profile from Auth0 and to call your backend APIs.

In this example, the `id_token` is stored in `localStorage` to keep the user authenticated after each page refresh.

```
var hash = lock.parseHash(window.location.hash);
if (hash) {
  if (hash.error) {
    console.log("There was an error logging in", hash.error);
    alert('There was an error: ' + hash.error + '\n' + hash.error_description);
  } else {
    //save the token in the session:
    localStorage.setItem('id_token', hash.id_token);
  }
}
```

This is how it will look on a browser...





4. Configure secure calls to your API

To configure secure calls to the API you are creating , implement `$.ajaxSetup` to send on each request, in the `Authorization` header with every ajax call, the JWT token received on the login and saved to `localStorage` as shown in Step 3.

```
$.ajaxSetup({
  'beforeSend': function(xhr) {
    if (localStorage.getItem('id_token')) {
      xhr.setRequestHeader('Authorization',
            'Bearer ' + localStorage.getItem('id_token'));
    }
  }
});
```

Note: The settings specified in `ajaxSetup` will affect all calls to $.ajax or Ajax-based derivatives such as $.get(). This may cause undesirable behavior if other callers (for example: plugins) are expecting the default settings. Therefore, use of this API is not recommended. Instead, set the options explicitly in the call or define a simple plugin to do so. For more information, see jQuery.ajaxSetup().

5. Retrieve the user profile and display user information

Use the `id_token` to retrieve the user profile and display the user's nickname:

```
  //retrieve the profile:
var id_token = localStorage.getItem('id_token');
```



```
  if (err) {
    return alert('There was an error getting the profile: ' + err.message);
  }
  // Display user information
```

```
    $('.nickname').text(profile.nickname);

  });
}
```

```
<p>Welcome <span class="nickname"></span></p>
```

To discover all the available properties of a user's profile, see Auth0 Normalized User Profile. Note that the properties available depend on the social provider used.

6. Log out

In this implementation, a log out involves simply deleting the saved token from `localStorage` and redirecting the user to the home page:

```
localStorage.removeItem('id_token');
userProfile = null;
window.location.href = "/";
```

7. All done!

You have completed the implementation of Login and Signup with Auth0 and jQuery.

Setting Up Your API

Are you looking on tutorials for setting up your API and making authorized requests? You can check these documents with more information on the subject.