

## Pushing Custom Variables in AngularJS

A guide for passing custom variables in AngularJS

## Quick Start Guide

Login Here: <https://secure.useriq.com>

Username: your email address

Password: YourPassword123!\* (This can be changed under user settings which can be found under your name in the top-right once you login to UserIQ)

## Installation Instruction

### 1. Add tracking code to your site

To start tracking, simply add the following javascript snippet to the footer of your application in the QA or test environment. **Place it just before the </body> tag.** Hopefully, you'll only need to add this to one file since it is the same on each page of your application.

```
<script type="text/javascript">
  var _uiq = _uiq || [];
  //TODO: Update the user and account example data with real data
  // _uiq.push(['setCustomVariable','1','user_id','1234','visit']);
  // _uiq.push(['setCustomVariable','2','user_name','Tony Stark','visit']);
  // _uiq.push(['setCustomVariable','3','account_id','5678','visit']);
  // _uiq.push(['setCustomVariable','4','account_name','Stark Industries','visit']);
  // _uiq.push(['setCustomVariable','5','user_email','tony.stark@stark.com','visit']);
  // _uiq.push(['setCustomVariable','6','signup_date','2015-03-14','visit']);
  _uiq.push(["enableLinkTracking"]);
  _uiq.push(["trackPageView"]);
  (function() {
    var _uiq_config_url=((("https:" == document.location.protocol) ? "https" : "http") +
"://feed.useriq.com/";
    var _uiq_prod_url=((("https:" == document.location.protocol) ? "https" : "http") +
"://secure.useriq.com/";
    _uiq.push(["setTrackerUrl", _uiq_prod_url+"visits/push"]);
    _uiq.push(["setSiteId", "403025301"]);
    var d=document, g=d.createElement("script"), s=d.getElementsByTagName("script")[0];
g.type="text/javascript";
    g.defer=true; g.async=true; g.src=_uiq_config_url+"useriq.js"; s.parentNode.insertBefore(g,s);
  })();
</script>
```

*See below for single page app install instructions (Only applicable if you have a single page app)*

---

To properly split the tracking code between a static footer and dynamic variables in a single page application (using Angular, for example), you will use the following tracking code in your footer (this can be a static JS resource):

## STATIC JS INCLUDE

```
<script>
var _uiq = _uiq || [];
(function() {
  var _uiq_config_url=(("https:" == document.location.protocol) ? "https" : "http") + "://feed.useriq.com/";
  var _uiq_prod_url=(("https:" == document.location.protocol) ? "https" : "http") + "://secure.useriq.com/";
  _uiq.push(["setTrackerUrl", _uiq_prod_url+"visits/push"]);
  _uiq.push(["setSiteId", "403025301"]);
  var d=document, g=d.createElement("script"), s=d.getElementsByTagName("script")[0]; g.type="text/javascript";
  g.defer=true; g.async=true; g.src=_uiq_config_url+"useriq.js"; s.parentNode.insertBefore(g,s);
})();
</script>
```

And you will set the variables in Angular.

You'll notice that there are no user identification variables being passed in the code above. That's because we're going to pass those variables from Angular once the user is logged in.

Here is a working example of a user authentication service that sets the user variables as soon as the user is logged in. I think you guys will only care about Custom Variable 1 at this point. We can set other variables as we need them for filtering later on:

## DYNAMIC ANGULAR CODE

```
UserService.get().$promise.then(function(current_user_data) {
  $scope.current_user = current_user_data;
  _uiq.push(['setCustomVariable', '1', 'user_id', $scope.current_user.email, 'visit']);
  _uiq.push(['setCustomVariable', '2', 'user_name', $scope.current_user.name, 'visit']);
  _uiq.push(['setCustomVariable', '3', 'account_id', $scope.current_user.customer_id, 'visit']);
  _uiq.push(['setCustomVariable', '4', 'account_name', $scope.current_user.account_name, 'visit']);
  _uiq.push(['setCustomVariable', '5', 'user_email', $scope.current_user.email, 'visit']);
  _uiq.push(["enableLinkTracking"]);
  _uiq.push(["trackPageView"]);
});
```

You can make this call anywhere in your application and it will establish the user as identified.

Once the code is called, UserIQ is able to recognize when an Angular view has changed and will send tracking data automatically. In other words, UserIQ doesn't need a page refresh to track user behavior.