Build a Full-Page Test Experience for Redirect with Persistent Parameters

You can use Monetate's Full-Page Test experiences to redirect from URL A to URL B, while keeping any existing parameters and values intact. You can also use this type of experience to redirect from URL A to URL B and add a new parameter value while keeping existing parameter value(s) intact.

Redirect from A to B with Persistent Parameters

For this use case, `site.com/?test=true` must redirect to the subdirectory `site.com/pathname/?test=true` while keeping the `test=true` parameter intact.

Within the regular expression (regex) are capture groups that correspond to the replacement text field below.

- `$1` is the first capture group.
- `$2` is the second capture group.

In this replacement text example, anything that the regex identifies as the second capture group is preserved and inserted after the replacement text by adding `$2` to **Replacement Page**.

Dissecting the Regular Expression

This URL regex consists of two capture groups.

First Capture Group

The initial piece of the first capture group, `https?:\/\/`, matches for the unsecure (`http`) *and* secure (`https`) versions of the URL protocol for `site.com`. By including an optional character with `s?`, it can match for either `http` or `https`.

The next piece of the first capture group, `(?:www|m|t)\.`, matches for the subdomain. The vertical pipe (`|`) character makes this into an array of OR options that matches for `www` or `m`, or `t` for desktop, mobile, or tablet top-level domains.

This example matches all top-level domain variants of the site `m.site` (mobile) or `t.site` (tablet), or `www.site` (desktop). The question mark and colon (`?:`) set this group to be a noncapture group.

The final piece of the first capture group, `site\.com\/`, matches for the domain `site.com` using the backslash escape character (`\`) to precede any symbols needed as literal characters, such as the forward slash (`/`) or the period (`.`).

Second Capture Group

The second capture group, `$|\?.*`, checks if the URL either ends by using a dollar sign (`$`) or looks for a parameter of any kind that's initialized with `?`. Finally, it checks if the parameter, if it exists, is followed by a value by looking for any number of any characters with a period and asterisk (`.*`).

With the second capture group intact, it can then be appended to a new URL in the Full-Page Test experience. On arriving a visitor is instantly redirected by the experience to the new URL with the same parameter in the URL.

Add a Parameter & Redirect from A to B with Previous Parameters Intact

For this use case, `site.com/?test=true` should redirect to `site.com/?test=true&string=test` *and* `site.com` should redirect to `site.com/?string=text`.

Within the regular expression (regex) are capture groups that correspond to the **Replacement Page** field below.

- `$1` is the first capture group.
- `$2` is the second capture group.
- `$3` is the third capture group.

In this example, the regex preserves any parameter value(s) it identifies as the third capture group and then inserts it after the new parameter value to be added by using `$3` in the **Replacement Page** input.

Dissecting the Regular Expression

This URL regex consists of three capture groups and one negative lookahead group.

A negative lookahead only succeeds if the regex inside the lookahead fails to match.

First Capture Group

The initial piece of the first capture group, `https?:\/\/`, matches for the unsecure (`http`) *and* secure (`https`) versions of the URL protocol for `site.com`. By including an optional character with `s?`, it can match for either `http` or `https`.

The second piece, `(?:www|m|t)\.`, matches for the subdomain. The vertical pipe (`|`) character makes this into an array of OR options that matches for `www` or `m`, or `t` for desktop, mobile, or tablet top-level domains.

This example matches all top-level domain variants of the site `m.site` (mobile) or `t.site` (tablet), or `www.site` (desktop). The question mark and colon (`?:`) set this group to be a noncapture group.

The final piece of the first capture group, `site\.com\/`, matches for the domain `site.com` using the backslash escape character (`\`) to precede any symbols needed as literal characters, such as the forward slash (`/`) or the period (`.`).

Second Capture Group

The second capture group, `$|\?`, checks if the URL either ends by using a dollar sign (`$`) or looks for a parameter of any kind that's initialized with `?`.

Third Capture Group

The third capture group checks if any number of any characters exist after the parameter(s) with `.*`.

Negative Lookahead Group

The negative lookahead group, `?!.*string=test`, looks for what does not match `?!` any number of any characters `.*` followed by a literal string of characters that matches the parameter and value being appended during this Full-Page Test experience `string=test`.

This piece of the regex is extremely important and ensures a loop occurs of constant redirecting and appending another copy of the same parameter and value.

Once the regex is complete, you can see how it works in the Full-Page Test experience. The first part of the URL remains intact and is placed at the beginning of the redirect URL with `$1`, then the parameter string and value `?string=test&` is appended. Including the ampersand (`&`) is very important because it preserves and appends any additional parameters and values captured in the third capture group that is added next with `$3`.

> The ampersand (`&`) does not negatively impact the redirect URL if there is no information in third capture group. However, it is present in the URL and therefore impacts any action conditions, such as the **URL ends with** option.