title: Sematext Experience Integrations descriptions: The Sematext Experience integrations cover all types of websites and all major SPA frameworks.

```
<div class="mdl-cell mdl-cell--4-col">
    <a href="#angular">
        <div class="demo-card-event mdl-card mdl-shadow--2dp">
            <div class="flip-card-container">
                <div class="flip-card">
                    <div class="side">
                        <img src="../../images/integrations/angular.png" alt="Angular" title
                    </div>
                    <div class="side back">
                        <h5>Angular</h5>Full support for single-page applications built with
                    </div>
                </div>
            </div>
        </div>
    </a>
</div>
<div class="mdl-cell mdl-cell--4-col">
    <a href="#react">
        <div class="demo-card-event mdl-card mdl-shadow--2dp">
            <div class="flip-card-container">
                <div class="flip-card">
                    <div class="side">
                        <img src="../../images/integrations/react.png" alt="React" title="Re
                    </div>
                    <div class="side back">
                        <h5>React</h5>Full support for single-page applications built with t
                    </div>
                </div>
            </div>
        </div>
    </a>
</div>
<div class="mdl-cell mdl-cell--4-col">
    <a href="#vuejs">
        <div class="demo-card-event mdl-card mdl-shadow--2dp">
            <div class="flip-card-container">
                <div class="flip-card">
                    <div class="side">
                        <img src="../../images/integrations/vuejs.png" alt="Vue.js" title="V
                    </div>
                    <div class="side back">
                        <h5>Vue.js</h5>Full support for single-page applications built with
                    </div>
                </div>
            </div>
        </div>
    </a>
</div>
<div class="mdl-cell mdl-cell--4-col">
    <a href="#ember">
        <div class="demo-card-event mdl-card mdl-shadow--2dp">
            <div class="flip-card-container">
                <div class="flip-card">
                    <div class="side">
                        <img src="../../images/integrations/ember.png" alt="Ember" title="Em
                    </div>
                    <div class="side back">
                        <h5>Ember</h5>Full support for single-page applications built with t
                    </div>
                </div>
            </div>
        </div>
    </a>
</div>

    <a href="#static-websites">
        <div class="demo-card-event mdl-card mdl-shadow--2dp">
            <div class="flip-card-container">
                <div class="flip-card">
                    <div class="side">
                        <img src="../../images/integrations/static.png" alt="Static Websites
                    </div>
                    <div class="side back">
                        <h5>Static Websites</h5>Full support for all types of static website
                    </div>
                </div>
            </div>
        </div>
    </a>
</div>

    <a href="#server-side-rendered-websites">
        <div class="demo-card-event mdl-card mdl-shadow--2dp">
            <div class="flip-card-container">
                <div class="flip-card">
                    <div class="side">
                        <img src="../../images/integrations/server.png" alt="Server-side ren
                    </div>
                    <div class="side back">
                        <h5>Server-side rendered websites</h5>Full support for all types of
                    </div>
                </div>
            </div>
        </div>
    </a>
</div>
```

All of the integrations require adding and configuring the Experience script. However, single-page applications require one more configuration step to register route changes.

Angular

The Angular SPA integration supports both Angular.js and Angular 2+.

Angular 2+

If your webapp uses the newer Angular you should add a call to `routeChange` whenever the route changes in your webapp. This can be done in your top-level component where `Router` is defined by adding the `ngOnInit` function and subscribing to the `Router` events as shown in the following example.

If you use TypeScript, you need to declare the global function:

```
declare global {
  interface Window { strum: Function; }
}
```

Add this to your code.

```
import { Component, OnInit } from '@angular/core';
import { Router, NavigationStart } from '@angular/router';

@Component({ selector: 'app', templateUrl: 'app.component.html' })

export class AppComponent implements OnInit {
  constructor(private router: Router) {}

  ngOnInit() {
    this.router.events.subscribe(event => {
      if (event instanceof NavigationStart) {
        window['strum']('routeChange', event.url);
      }
    });
  }
}
```

Angular.js

If your webapp uses Angular.js 1.x you should add a call to `routeChange` whenever the route changes in your webapp. To do this you'll need to adjust your client side view inside Angular.js:

```
$scope.$on('$routeChangeStart', function () {
  window['strum']('routeChange', window.location.href);
});
```

React

The React integration ties into React Router and tracks calls to `routeChange`. You should add a call to `routeChange` whenever the route changes in your webapp. This can be done in your top-level component where `Router` is defined.

If you use TypeScript, you need to declare the global function:

```
declare global {
  interface Window { strum: Function; }
}
```

Add this to your code.

```
import React from 'react';
import { createBrowserHistory as createHistory } from 'history';

const history = createHistory();

history.listen((location, action) => {
  if (action !== 'REPLACE') {
    window['strum']('routeChange', window.location.href);
  }
})

export default function App() {
  return (
    <Router history={history}>
      ...
    </Router>
  );
}
```

Vue.js

The Vue.js integration works by watching for calls to the `routeChange` function where you have the `router-view` defined.

If you use TypeScript, you need to declare the global function:

```
declare global {
  interface Window { strum: Function; }
}
```

Add this to your code.

```
<template>
  <div id="app">
    <router-view/>
  </div>
</template>
<script>
  export default {
    name: 'app',
    watch: {
      $route() {
        strum('routeChange', document.location.href);
      }
    }
  }
</script>
```

Ember

The Ember integrations uses the `reopen` function to configure a function to trigger every time your application changes routes. This event is called `didTransition`.

If you use TypeScript, you need to declare the global function:

```
declare global {
  interface Window { strum: Function; }
}
```

Add this to your code.

```
import EmberRouter from '@ember/routing/router';
import { on } from '@ember/object/evented';

EmberRouter.reopen({
  doInformAboutRouteChange: on('didTransition', function() {
    // eslint-disable-next-line
    window['strum']('routeChange', window.location.href);
  }),
});

export default Router;
```

Static websites

Static websites don’t require any additional configuration, just add the Experience script to the `<head>` of your site and you’re ready to go!

Server-side rendered websites

Server-side rendered websites don’t require any additional configuration either. Add the Experience script to the `<head>` of your site and you’re ready to go!