



Dataopsamlingsvejledning

Applikationer til
Erhvervsfremmeplatformen /
Virksomhedsguiden
Version 2.0.3

1. maj 2025

Indhold

1 Introduktion.....	3
2 Kort om Piwik.....	4
3 Kommunikationsvej	5
4 NPM bibliotek.....	6
5 Piwik-events og biblioteks-metoder	7
5.1 Piwik service	7
5.1.1 Options API.....	7
5.1.2 Composition API	7
5.2 PageView	8
Tilsvarende biblioteksmetode.....	8
Eksempel	8
5.2.1 hash/fragment	8
5.3 AppDownload.....	9
Tilsvarende biblioteksmetode.....	9
Eksempel	9
5.4 AppCTAClick	9
Tilsvarende biblioteksmetode.....	9
Eksempel	10
5.5 AppStart.....	10
Tilsvarende biblioteksmetode.....	10
5.6 AppSlut	10
Tilsvarende biblioteksmetode.....	10
6 Test	11

1 Introduktion

På Virksomhedsguiden (VG) er det muligt for leverandør-applikationer, at sende data om brugeradfærd til Erhvervsstyrelsens Piwik installation. Denne vejledning beskriver denne proces.

Det er ikke tilladt at lave opsamling af data om brugeradfærd på anden vis.

Data fra Piwik udleveres efter aftale med Erhvervsstyrelsen.

I forretningsmæssige og juridiske dokumenter kaldes applikationerne tredjeparts-applikationer.

2 Kort om Piwik

Piwik bruges til at tracke brugeres adfærd på Erhvervsstyrelsens hjemmesider. Piwik modtager datapunkter i form af events. Hvert event har en række attributter, og man kan efterfølgende lave statistik over lignende events.

Et eksempel kunne være et "Klik på åben accordion"-event med attributterne "url" og "accordionTitel"

Så kan man sidenhen undersøge, hvor mange, der åbner en accordion på en bestemt side (url) med en bestemt titel (accordionTitel)

3 Kommunikationsvej

Leverandør-applikationer skal ikke kommunikere direkte med Piwik. Dette styres af VG. I stedet emitter leverandør-applikationer Vue-events til VG, som håndterer resten. På denne måde sikres der et ensformigt kommunikations flow, og leverandør-applikationer skal ikke håndtere cookie consent o.lign. Der er lavet et npm bibliotek, til at hjælpe med kommunikationen fra leverandør-applikationer til VG. Dette er beskrevet nedenfor.



4 NPM bibliotek

VG stiller et npm bibliotek til rådighed, som udstiller en række metoder, som svarer til de Piwik-events, som er til rådighed for leverandør-applikationer. Events og metoder er beskrevet nedenfor.

Biblioteket er allerede til rådighed i demo leverandør-applikationen.

Biblioteket er til rådighed her: <https://www.npmjs.com/package/@erst-vg/piwik-event-wrapper> og kan hentes med:

```
$ npm install @erst-vg/piwik-event-wrapper
```

5 Piwik-events og biblioteks-metoder

Vi stiller følgende events til rådighed for leverandør-applikationer:

1. PageView
2. AppDownload
3. AppCTAClick
4. AppStart
5. AppSlut

Nedenfor er det beskrevet, hvordan og hvornår hvert event skal bruges, og hvilke input de tilsvarende biblioteksmetoder forventer.

5.1 Piwik service

Biblioteket udstiller en `piwikService` som skal kaldes for alle data events.

```
import { piwikService } from '@erst-vg/piwik-event-wrapper';
```

Både *Options API* og *Composition API* understøtter denne service, men skal initialiseres forskelligt.

5.1.1 Options API

Hvis leverandør-applikationen er skrevet i *options API* skal man i roden af leverandør-applikationen **Applikation.vue** gøre følgende:

```
created() {  
  piwikService.init(this.$emit);  
}
```

5.1.2 Composition API

Hvis leverandør-applikationen er skrevet i *Composition API* skal man i roden af leverandør-applikationen **Applikation.vue** gøre følgende:

```
<script setup lang="ts">
  import { piwikService, DataEvents } from '@erst-vg/piwik-event-wrapper';

  // Angiv kun de events der bruges i applikationen
  const emit = defineEmits([DataEvents.PAGE_VIEW,
    DataEvents.CTA_CLICK_EVENT, DataEvents.DOWNLOAD_EVENT,
    DataEvents.START_EVENT, DataEvents.SLUT_EVENT]);
  // Initialiser PiwikService, så den kan bruges i underliggende Vue
  komponenter
  piwikService.init(emit);
</script>
```

5.2 PageView

Dette event skal bruges, i forbindelse med en handling, som resulterer i hvad, der minder om et sideskift. Det kan f.eks. være efter et tryk på en "næste"-knap, eller hvis man bliver taget til en resultatside.

Dette skal implementeres sidestående med brugen af `window.location.hash` til at indikere hvilken side man er på (se nedenfor). Eventet skal først sendes **efter** man har opdateret `window.location.hash`, så det er den korrekte (nye) url, der bliver sendt til Piwik.

PageView-eventet skal **ikke** kaldes ved det initiale kald til leverandør-applikationen, da det allerede sker i VG.

Tilsvarende biblioteksmetode

`emitPageViewEvent` svarer til PageView-eventet, og kan modtage input som angiver URL fragment identifier (#):

Eksempel

Eksempel som sender hele URL'en til Piwik

```
piwikService.emitPageViewEvent();
```

Eksempel, hvor man manuelt angiver URL fragment identifier som vil blive sendt med til Piwik som # værdien.

```
piwikService.emitPageViewEvent("Indberetter-du-arsrapport_1c3ph9wdnhil");
```

Man skal kun manuelt angive URL fragment identifier i `emitPageViewEvent` funktionen hvis det ikke er muligt at angive det direkte i URL'en af tekniske årsager. Hvis det allerede er angivet i URL'en, så ignoreres den manuelle fragment identifier.

5.2.1 hash/fragment

Det er hele url'en inklusiv hash, der bliver sendt til Piwik ifm. `appPageView`. Denne url bliver brugt til at genkende de forskellige undersider af applikationen. Det er derfor vigtigt, at der bliver valgt en sigende værdi.

Man kan med fordel bruge titlen på siden, hvis siderne har en titel (husk at bruge slugify, så titlerne har en pæn form).

Hvis der er tale om et lineært forløb igennem applikationen kan man bruge nummerering i hash-værdien.

Hash værdien kan godt være en sammensætning af flere dele.

Det skal være muligt, at forstå hvor man er i applikationen ud fra hash-værdien alene.

Alle hash-værdier skal være unikke; der må ikke være to sider, der har den samme værdi.

Alle hash-værdierne skal følge et lignende format.

Hash-værdierne skal holdes forholdsvis korte.

5.3 AppDownload

Dette event skal bruges, i forbindelse med klik på en "download"-knap eller tilsvarende.

Det skal om muligt kun bruges, hvis download er succesfuldt.

Tilsvarende biblioteksmetode

`emitDownloadEvent` svarer til AppDownload-eventet og tager følgende input:

- `fileIdentifier: string` - en unik identifikation på den fil, der bliver downloadet; ofte fil-navnet - f.eks. "Årsrapport2021vers4.pdf"
- `data?: string` - en attribut, som kan bruges til eventuel ekstra data - skal kun bruges, hvis det er relevant, og skal holdes kort

Eksempel

```
piwikService.emitDownloadEvent('Årsrapport2021vers4.pdf',
this.myRelavantData);
```

5.4 AppCTAClick

Dette event skal bruges, i forbindelse med klik på en "call-to-action" (CTA) henvisning.

Call-to-action (CTA) er en opfordring til brugeren af en applikation specifikt designet til at fremkalde en reaktion, som ejeren af applikationen ønsker fra brugeren. Typisk anvender CTA'er et imperativt verbum som "Læs mere om ...", "Få inspiration til ..." eller "Tjek om du ...".

Typisk vil "call-to-action" være et link på en resultatside, som fører et nyt sted hen. Antallet af CTA i en applikation kan svinge meget afhængig af applikationens formål. Ofte vil CTA'erne også være forskellige alt afhængig af brugerens interaktion med applikationen. Uanset indholdet i "call-to-action", så vil eventen teknisk set være den samme. En nærmere analyse af CTA'erne kan ikke ske ved hjælp af eventen (AppCTAClick), men ved hjælp af en almindelige flow analyse i Piwik. CTA'en måler alene volumen af antal klik på henvisninger ud af applikationen.

Tilsvarende biblioteksmetode

`emitCTAClickEvent` svarer til AppCTAClick-eventet og tager følgende input:

- `eventType: string` - en unik identifikation på den type event, der bliver udført - f.eks. "Link til videre læsning"

- `data?: string` - en attribut, som kan bruges til eventuel ekstra data - f.eks. url'en på linket

Eksempel

```
piwikService.emitCTAClickEvent('Beregn', 'Fradrag');
```

5.5 AppStart

Dette event skal kun bruges når applikationen starter et forløb, som har en afslutning. Det bruges fx. når man klikker på en start knap, der starter et forløb der har en resultatside.

Tilsvarende biblioteksmetode

`emitStartEvent` tager ikke noget input

Eksempel

```
piwikService.emitStartEvent();
```

5.6 AppSlut

Dette event skal kun bruges når applikationen afslutter et forløb. Hvis der ikke er registreret et tilsvarende `AppStart` event, så ignorerer VG dette event.

Tilsvarende biblioteksmetode

`emitSlutEvent` tager ikke noget input

Eksempel

```
piwikService.emitSlutEvent();
```

6 Test

Demo leverandør-applikationen er sat op til, at skrive den data, som bliver sendt til biblioteks-metoderne direkte til konsollen i stedet. Test

Hvis man vil preview'e leverandør-applikationen fra administrationen af leverandør-applikationer vil data ligeledes blive skrevet til konsollen. I preview bliver der ikke sendt data til Piwik.

Hvis leverandør-applikationen allerede bliver brugt i produktion, så kan med fordel bruge plug-in'et Omnibug til at overvåge de events, der bliver sendt til Piwik.