



Teknisk vejledning

Applikationer til
Erhvervsfremmeplatformen /
Virksomhedsguiden
Version 2.0

Exported on Jan 19, 2022

Indhold

1 Introduktion.....	3
1.1 Leverandør applikationer.....	3
1.2 Formål	3
1.3 Målgruppe.....	3
2 Processer	4
2.1 Bliv applikation leverandør	4
2.1.1 Ansøgning til ERST.....	4
2.1.2 Login til redaktion.....	4
2.1.3 Adgang til applikation vejledning.....	4
2.1.4 Sandkassemiljø.....	4
2.2 Bygge og teste nye applikationer	4
2.2.1 Version.....	5
2.2.2 Multiple komponenter.....	5
2.2.3 Parameter varianter	5
2.2.4 Administration af leverandør applikation	5
2.3 Godkendelse af ny applikation	6
2.3.1 Huskeliste inden indsendelse.....	6
2.4 GIT hosts	7
2.4.1 GitHub.....	7
2.4.2 GitLab	7
2.4.3 Krav til applikation.....	8
2.5 Visning af godkendt leverandør applikation i VG	8
2.6 Opdatering af en eksisterende leverandør applikation	9
2.7 Slet en applikation	9
3 Teknisk guide.....	10
3.1 Teknologi	10
3.1.1 Vue	10
3.1.2 Typescript	10
3.1.3 SCSS	10
3.2 Tilgængelige biblioteker	10
3.2.1 Det fælles designsystem.....	10
3.2.2 Axios	11
3.2.3 Krav til leverandør.....	11
3.3 Brug af eksterne API	12
3.3.1 Personfølsom data.....	12
3.4 Begrænsninger/regler.....	12
3.4.1 TypeScript.....	12
3.4.2 Ingen \$parent og \$root.....	12
3.4.3 Ingen DOM manipulation	12
3.4.4 Alle links skal bruge target="_blank"	12
3.4.5 Alle kald til eksterne API skal være async.....	13
3.4.6 <style> tag skal gøre brug af scoped attribut	13
3.4.7 Ingen opdatering af eksisterende cookies.....	13
3.4.8 Error handling	13
3.4.9 Loading spinner	14

1 Introduktion

På Virksomhedsguiden (VG) er der et krav om at kunne præsentere indlejret indhold i form af eksterne applikationer, som er bygget af eksterne leverandører. Dette indhold bygges som selvstændige komponenter, og er refereret til som leverandør applikationer.

1.1 Leverandør applikationer

Leverandør applikationer er selvstændige applikationer, som skal sikre at eksterne leverandører kan bidrage med at producere indhold til VG. Samtidig sikrer leverandør applikationer at brugeren får en komplet brugeroplevelse uden at skulle navigeres væk fra VG.

En applikation kan ikke kommunikere med VG i form af database kald, men kan kun gøre brug af de biblioteker som bliver stillet til rådighed af VG. En applikation har dog mulighed for at kommunikere og hente data fra eksterne web services. Dette bliver beskrevet i flere detaljer i afsnit (Teknisk guide).

1.2 Formål

Formålet med leverandør applikationer er at præsentere brugeren af VG for eksternt indhold, som er bygget af eksterne leverandører, og dermed sikre at brugeren får den bedst mulige oplevelse på én samlet platform.

1.3 Målgruppe

Denne guide er henvendt til de eksterne leverandører, som skal udvikle applikationer til VG. Det forventes derfor at læseren har kendskab til web udvikling, herunder HTML, CSS og Javascript.

2 Processer

For at en leverandør applikation kan blive præsenteret på VG, er der en del processer der skal gennemføres. Disse processer bliver beskrevet i følgende afsnit og dækker både over de processer som en leverandør skal igennem, samt dem som administrator og redaktør kommer igennem.

2.1 Bliv applikation leverandør

2.1.1 Ansøgning til ERST

Nye leverandører af applikationer skal godkendes af Erhvervsstyrelsen. Se dokumentationen [Forretningskrav](#) for at læse mere om de forskellige krav der stilles til nye leverandører.

2.1.2 Login til redaktion

Ved oprettelse af ny leverandør får leverandøren tilsendt en mail med login til VG redaktionen, hvor leverandøren kan administrere sine applikationer og oprette nye.

2.1.3 Adgang til applikation vejledning

Ved oprettelse af ny leverandør får leverandøren tilsendt en mail med link til denne vejledning og til [Design vejledning](#).

2.1.4 Sandkassemiljø

Ved oprettelse af ny leverandør får leverandøren tilsendt en mail med link til et sandkassemiljø, som er tilgængeligt via et Git repository. Sandkassen gør det nemt og hurtigt for leverandøren at komme igang med udvikling og test af leverandør applikationer.

2.2 Bygge og teste nye applikationer

Når man er godkendt som leverandør kan man begynde at udvikle og teste sin applikation. Dette gøres ved at forke det Git repository som man får adgang til ved oprettelse. Sandkassen sættes hurtigt og nemt op ved at følge de steps som er beskrevet i *README.md*. Disse er:

1. Åben et command-line vindue i roden af det forkede sandkasse repo
2. Kør npm install
3. Kør npm run serve
4. Tilgå localhost på den port som angives i konsollen
5. Start udvikling i Applikation.vue (herefter kaldet entry point) som ligger i *src/components/Aplikation.vue*)

Der medfølger hot reload i sandkassemiljøet, så ændringer i Applikation.vue slår igennem på localhost med det samme når der gemmes.

2.2.1 Version

Sandkassemiljøet version 2.x er ikke længere kompatibel med version 1.x grundet mange ændringer, som bla. har gjort den mindre afhængig af VG, og har betydeligt reduceret størrelsen på den færdige leverandør applikation. Derfor hvis man har en leverandør applikation, som er udviklet med version 1.x bør man arbejde videre på den ud fra tag *v1.0.1* som er den seneste version inden skiftet til 2.x.

Nye leverandør applikationer bør udvikles på 2.x eller nyere.

2.2.2 Multiple komponenter

Fra version 2.0 understøtter sandkassen, at en leverandør applikationen kan spænde over flere Vue komponenter. Det er vigtigt at vide, at alle komponenter og stylesheets der har referencer til entry point komponenten, både direkte og indirekte vil blive inkluderet i den færdige leverandør applikation. Vue rammeværket som ligger rundtom (bla. *main.ts* og *App.vue*) vil ikke blive inkluderet.

2.2.3 Parameter varianter

VG tilbyder en applikation Vue property kaldet *variant*, som sendes med ind til leverandørens applikation. Værdien af varianten angives under administration af leverandør applikationen som beskrives i næste afsnit

```
interface Variant {  
  navn: string;  
  aktiv: boolean;  
  parametre: {  
    parameternavn: string;  
    parametervaerdi: string;  
  }[];  
}
```

Se eksempel på brugen af variant property i Applikation.vue i sandkassen, som styrer hvilken baggrundsfarve et element styles med.

2.2.4 Administration af leverandør applikation

Administration af leverandør applikationen foregår igennem VG platformen. Leverandøren logger ind i VG administrationen (login er tilsendt på mail i forbindelse med oprettelse af leverandør), klikker på "Administrer applikationer" og indtaster de nødvendige oplysninger.

Leverandøren har her mulighed for at angive værdien af evt. parameter varianter som gøres ved at vælge checkbox "Applikation bruger parameter" som er præudfyldt med et eksempel på hvordan JSON strukturen skal være.

Leverandøren skal herefter tilpasse JSON til applikationen og efterfølgende gemme.

JSON forventer et array af typen Variant, men vær opmærksom på at der kun vælges én variant når applikationen indsættes af en redaktør. Vælg venligst et sigende variant navn så redaktøren kan vælge den rette variant.

Leverandøren kan deaktivere en variant ved at ændre "aktiv" flaget i JSON. Efterfølgende vil en redaktør ikke kunne vælge denne variant, men eksisterende ydelser der allerede benytter denne variant vil fortsat bruge den, så varianten skal stadig understøttes af leverandør applikationen.

Når leverandøren kun vil opdatere variant JSON, så er det vigtigt der klikkes på "Gem" knappen og ikke "Gem som kladde" da sidstnævnte også ændrer tilstanden på applikationen.

Ved at klikke på "preview" knappen kan leverandøren se hvordan applikationen vises når den bliver indsat på en ydelse, og dette bør afprøves inden den sendes til godkendelse. Ved preview vælges den første variant som default, men der er mulighed for at skifte variant vha. dropdown menu.

2.3 Godkendelse af ny applikation

Når leverandøren er færdig med at udvikle og teste sin applikation, så skal der angives en review URL til den leverandør applikation der skal godkendes.

Hvis kildekoden ligger i et beskyttet repository, er det leverandørens eget ansvar at oprette og angive en access_token i URL'en så kildekoden kan tilgås.

Da leverandørens kildekode hentes og bearbejdes programmatisk, er det vigtigt det fulde repo kan clones.

2.3.1 Huskeliste inden indsendelse

- Leverandør bør altid teste en leverandør applikation med preview knappen i VG administrationen inden den sendes til godkendelse
- URL til repo (ikke review URL) skal kunne clones via "git clone"

Når alle nødvendige informationer er angivet, indsendes leverandør applikationen til godkendelse. En administrator vil herefter reviewe koden og enten afvise eller godkende applikationen. Hvis applikationen bliver afvist, sender administrator en mail til leverandør med grundlag for afvisningen.

2.4 GIT hosts

Følgende GIT hosts understøttes

2.4.1 GitHub

URL til GitHub hostet kildekode skal angives på følgende form:

```
https://${ACCESS_TOKEN+@}github.com/${REPO}/${PROJECT}/tree/${SHA}/
```

Eksempel:

REPO: Erhvervsfremmeplatformen

PROJECT: tredjepartsapps

ACCESS_TOKEN: stzJfjlRp5TgWlJc5T3

SHA: 1cd9a9a39e094a31dfd33319089320ae49d605cf

Browser URL til kildekode med commit SHA:

<https://github.com/Erhvervsfremmeplatformen/tredjepartsapps/commit/1cd9a9a39e094a31dfd33319089320ae49d605cf>

kan hentes på:

<https://github.com/Erhvervsfremmeplatformen/tredjepartsapps.git/>

leverandør skal angive denne URL som review URL:

<https://github.com/Erhvervsfremmeplatformen/tredjepartsapps/tree/1cd9a9a39e094a31dfd33319089320ae49d605cf/>

eller hvis repo er privat er der brug for en token:

<https://stzJfjlRp5TgWlJc5T3@github.com/Erhvervsfremmeplatformen/tredjepartsapps/tree/1cd9a9a39e094a31dfd33319089320ae49d605cf/>

Bemærk URL'en skal afsluttes med '/' og pege på et commit SHA (1cd9a9a39e094a31dfd33319089320ae49d605cf), så versionen af kildekoden kan fastfryses. En URL med angivelse af branch vil derfor ikke blive godkendt, da den bagvedliggende kode kan ændre sig.

2.4.2 GitLab

URL til GitLab hostet kildekode skal angives på følgende form:

```
https://${GITLAB_HOST}/git/${REPO}/${PROJECT}/-/commit/${SHA}/
```

Eksempel:

```
GITLAB_HOST: gitlab.com
FILE_PATH: src/components/Applikation.vue
REPO: vgapplication
PROJECT: appl
PROJECT_ID: 1817
ACCESS_TOKEN: aUx18aeb31PUfw-PNAMQ
SHA: 1f7f1b71c5961df99e5131572d87a31f4d9edc98
```

Browser URL til kildekode:

<https://gitlab.com/git/vgapplication/appl/-/commit/1f7f1b71c5961df99e5131572d87a31f4d9edc98>

kan hentes på:

<https://gitlab.com/git/vgapplication/appl.git/>

leverandør skal angive denne URL som review URL:

<https://gitlab.com/git/vgapplication/appl/-/tree/1f7f1b71c5961df99e5131572d87a31f4d9edc98/>

eller hvis repo er private, og der er brug for en token:

<https://username:stzJfjIRp5TgW1Jc5T3@gitlab.com/git/vgapplication/appl/-/tree/1f7f1b71c5961df99e5131572d87a31f4d9edc98/>

Bemærk URL'en skal afsluttes med '/' og pege på et commit SHA (1f7f1b71c5961df99e5131572d87a31f4d9edc98), så versionen af kildekoden kan fastfrys. En URL med angivelse af branch vil derfor ikke blive godkendt, da den bagvedliggende kode kan ændre sig.

2.4.3 Krav til applikation

- Eksterne HTTP kald skal være HTTPS ligesom VG
- Stier til imports af både komponenter og stylesheets skal være relative, og må derfor ikke bruge evt. alias
- Der må ikke være fejl eller warnings ved kørsel af "npm run serve"

2.5 Visning af godkendt leverandør applikation i VG

Så snart en leverandør applikation er godkendt af administrator, er applikationen aktiv og tilgængelig for redaktører, så de kan indsættes på VG.

2.6 Opdatering af en eksisterende leverandør applikation

Når der skal laves opdateringer til en leverandør applikation, kan leverandøren gå ind i VG administrationen, klikke på Administrer applikationer, og vælge den applikation som skal opdateres, fra en liste. Listen indeholder de eksisterende applikationer som leverandøren har oprettet hos VG. Det er muligt at opdatere navnet på applikationen eller indsende ny review URL til godkendelse.

På samme måde som når der oprettes en ny applikation, bliver den opdaterede leverandør applikation kode reviewet af en administrator, som enten godkender eller afviser koden. Godkendes koden, bliver den gamle kode overskrevet af den nye.

2.7 Slet en applikation

Hvis en leverandør vil have fjernet en applikation fra VG, skal der sendes en mail til Erhvervsstyrelsen med navnet på den applikation som skal slettes. Husk at angive i mail om applikationen bare skal afpubliceres eller fjernes helt fra systemet. Hvis en applikation afpubliceres, kan redaktører ikke længere tilføje applikationen til ydelser, men den eksisterer stadig i systemet og kan publiceres igen på et senere tidspunkt.

I en senere version bliver der tilføjet en slet knap for at lette arbejdsgangen.

3 Teknisk guide

Dette afsnit beskriver de teknologiske muligheder og afgrænsninger en leverandør har til rådighed, for at kunne udvikle leverandør applikationer.

3.1 Teknologi

3.1.1 Vue

Frontend i VG er udviklet med Javascript frameworket Vue.js

2.6.x (<https://vuejs.org/v2/guide/>) og det er pt. et krav at en leverandør applikation også udvikles i Vue.js med samme version.

Det forventes Vue komponenter udvikles som Single File Components (<https://vuejs.org/v2/guide/single-file-components.html>) med standard *options API* hvor *src/components/Aplikation.vue* i sandkasse miljøet kan evt. bruges som skabelon.

3.1.2 Typescript

For at sikre høj kodekvalitet, er alle Vue komponenter i VG udviklet i Typescript (<https://www.typescriptlang.org/>). Det er også et krav for leverandør applikationer. Se i *package.json* for hvilken version der pt. bruges af TypeScript.

3.1.3 SCSS

VG anvender SCSS (<https://sass-lang.com>) til styling af brugerfladen. En leverandør applikation skal også bruge SCSS.

3.2 Tilgængelige biblioteker

Sandkassemiljøet tilbyder fra start nogle af de NPM biblioteker som VG også bruger.

3.2.1 Det fælles designsystem

Hele brugergrænsefladen i VG er bygget op omkring Det Fælles Designsysteem (DKFDS - <https://designsystem.dk/>). Det Fælles Designsysteem er udviklet for at sikre en ensartet funktionalitet og design i selvbetjeningsløsninger på tværs af myndigheder. Det skaber genkendelse for brugerne af løsningerne og gør det nemmere for dem at foretage selvbetjening.

Alle komponenter som er tilgængelige i DKFDS er tilgængelige for applikation leverandøren. Man behøver ikke at importere DKFDS, da det allerede er aktiveret globalt i VG og i sandkassen. De tilgængelige komponenter er beskrevet i detaljer i applikation Design Guiden ([Design vejledning](#)).

VG bruger på nuværende tidspunkt version: **dkfds@7.4.0** og **dkfds-plugins@6.0.1**

Det er et krav, at man som applikation leverandør gør brug af de komponenter, som er tilgængelige i DKFDS, og det er derfor ikke nødvendigt at udvikle sine egne standard komponenter.

Bemærk VG har egen styling af DKFDS standard komponenter, så fx. standard DKFDS knapper vil blive stilet anderledes når applikationen er indlejret i VG end når den vises i sandkasse miljøet.

1.1.1.1 3.2.1.1 SVG ikoner

DKFDS tilbyder en række ikoner (<https://designsystem.dk/design/ikoner/>) som leverandør applikationen kan bruge frit. Samlingen af de ikoner der skal bruges skal angives i leverandør applikationen. Se hvordan i *src/components/SvgIcons.vue* i sandkassen.

Bemærk VG bruger også DKFDS ikon samling, så hvis der skal tilføjes custom ikoner så bør de have et ID som er unikt, og som derfor ikke kolliderer med VG egne custom ikoner. ID kan fx være prefixet med 'xla-' eller lignende.

Hvis der er ID kollision, så vil ikonet fra VG vises.

3.2.2 Axios

VG bruger NPM modulet Axios (<https://vuejs.org/v2/guide/single-file-components.html>) til netværkskald, og det anbefales også til leverandør applikationer. Axios er tilgængelig i sandkasse miljøet og *src/components/Aplikation.vue* viser et eksempel på brug af Axios.

Det er ikke tilladt at ændre den globale Axios standardkonfiguration eller tilføje interceptors til den globale Axios instans. Applikationer der bruger den globale Axios instans på denne måde, godkendes ikke.

3.2.3 Krav til leverandør

Hvis leverandør applikationen benytter sig af NPM moduler, som endnu ikke er tilføjet til sandkassen, så er det meget vigtigt at leverandøren informerer omkring dette så tidligt i forløbet som muligt, da det vil kræve en ny deployment release af VG, der skal planlægges i god tid. Følgende information skal angives ved nye NPM moduler:

- NPM modul navn
- Version
- Typescript declaration file (hvis nødvendigt)

3.3 Brug af eksterne API

3.3.1 Personfølsom data

Alle applikationer, som præsenterer data fra eksterne API skal overholde GDPR loven (<https://gdpr.dk/>) i forhold til personfølsom data.

3.4 Begrænsninger/regler

Grundet sikkerhed og stabilitet er der nogle begrænsninger som applikation leverandører skal følge for at kunne få godkendt en applikation. Enhver applikation der ikke følger de nedenstående regler, vil ikke blive godkendt i review processen.

3.4.1 TypeScript

Det er et krav at leverandør applikationer udvikles i TypeScript.

3.4.2 Ingen \$parent og \$root

Det er ikke tilladt at bruge this.\$parent eller this.\$root for at tilgå parent componenter i Vue.

3.4.3 Ingen DOM manipulation

JavaScript i en leverandør applikation har ikke tilladelse til at tilgå eller manipulere DOM udenfor leverandør applikationen. Dette sikres ved at angive et root element, hvor der kun må vælges elementer under elementet. Dette må højst være applikationens root element i HTML strukturen.

Fremfor at vælge et element globalt:

```
document.querySelector(".name")
```

bør det vælge elementet indenfor leverandør applikationen:

```
document.querySelector(".applikation-container .name")
```

Ydermere er det generelt ikke tilladt at mutere window objektet på nær følgende:

- window.location.hash.
- window.scrollTo

3.4.4 Alle links skal bruge target="_blank"

Alle links (<a href>) skal bruge attribut target="_ blank" for at åbne linket i en ny fane / et nyt vindue.

`<a href="https://www.google.com" target="_blank">Link til Google</a>`

3.4.5 Alle kald til eksterne API skal være async

Al adgang til Web API'er skal bruge asynkrone kald for at holde VG responsiv.

```
callAPI() {
  this.pending = true;
  this.error = false;
  axios.get("https://jsonplaceholder.typicode.com/todos/").then(({
    data }) => {
    // will show the returned data in DOM
    this.response = data;
  }).catch(() => {
    // will make the error message DOM visible
    this.error = true;
  }).finally(() => {
    // stop showing loading spinner
    this.pending = false;
  });
}
```

3.4.6 <style> tag skal gøre brug af scoped attribut

<style> tag skal altid indeholde attributten "scoped" for at begrænse CSS-styling til selve applikation komponenten.

```
<style lang="scss" scoped>
  @import '../styles/components/_external.scss';
</style>
```

3.4.7 Ingen opdatering af eksisterende cookies

Det er ikke tilladt at manipulere eksisterende cookies eller request headers.

3.4.8 Error handling

Sørg for, at koden håndterer fejl, og viser en meningsfuld fejlmeddelelse til brugeren:

- Alle Web API-opkald skal have en fejlhåndtering, der viser brugeren, at noget gik galt og muligvis giver dem mulighed for at prøve igen.

3.4.9 Loading spinner

Asynkrone operationer med Web API kald skal vise en form for loading spinner indtil operationen fejler eller fuldføres succesfuldt. Følgende DKFDS klasse kan bruges til at vise en spinner:

```
<div v-if="pending" class="spinner" aria-label="Henter indhold" />
```

[Synes godt om](#) Vær den første til at synes godt om dette

-