



Teknisk vejledning

Applikationer til
Erhvervsfremmeplatformen /
Virksomhedsguiden
Version 3.0

31. august 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	Sandkasse-miljø.....	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	Side navigation i leverandør-applikation	6
2.2.5	Administration af leverandør-applikation	6
2.3	Godkendelse af ny applikation	7
2.3.1	Huskeliste inden indsendelse.....	7
2.4	GIT hosts	7
2.4.1	GitHub.....	8
2.4.2	GitLab	8
2.4.3	Krav til applikation	9
2.5	Visning af godkendt leverandør-applikation i VG	9
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	11
3.3.1	Personfølsom data	11
3.4	Begrænsninger/regler.....	11
3.4.1	TypeScript.....	12
3.4.2	Ingen \$parent og \$root.....	12
3.4.3	Ingen DOM operationer udenfor leverandør-applikationen	12
3.4.4	Alle eksterne links skal bruge target="_blank"	12
3.4.5	Alle kald til eksterne API skal være async.....	12
3.4.6	Styles skal være scoped	13
3.4.7	Ingen opdatering af eksisterende cookies.....	13
3.4.8	Error handling	13
3.4.9	Loading spinner	13
3.4.10	Event listeners	14
3.4.11	Logning	14
3.4.12	Miljøvariabler.....	14
4	Login.....	15
4.1	Props	15
4.2	Events.....	15
4.3	Konfiguration	16
4.3.1	Sletning af brugerdata.....	16
4.3.2	API key.....	16
4.4	Forside.....	16
4.4.1	Logget ind	17
4.4.2	Ikke logget ind	17

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).

I forretningsmæssige og juridiske dokumenter kaldes leverandør-applikationerne tredjeparts-applikationer. En leverandør-applikation skal i princippet være helt isoleret fra VG med få undtagelser.

Det er vigtigt at understrege, at en leverandør-applikation er et Vue komponent med evt. under komponenter, som kan vises i VG som var det et almindeligt Vue komponent tilhørende VG. Det er derfor ikke en fuld SPA med råderet over egen Vue instans, som kan integreres i VG.

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 [Grundlæggende vejledning \(forretningskrav\)](#) 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 Sandkasse-miljø

Sandkassen gør det nemt og hurtigt for leverandøren at komme i gang med udvikling og test af leverandør-applikationer.

For at komme hurtigt igang, er der etableret et sandkasse-miljø. Det er en fungerende leverandør-applikation, som kan bruges som fundament til ny leverandør-applikationen.

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

Man bør checke det tag ud, som svarer til versionen af den tekniske vejledning, da *master* branch kan indeholde udokumenterede features og ændringer.

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. Åbn et command-line vindue i roden af det forkede sandkasse repo
2. checkout seneste tag på master, som passer med versionen af dokumentationen fx. v2.0.0
3. Kør npm install
4. Kør npm run serve
5. Tilgå localhost på den port som angives i konsollen
6. Start udvikling i Applikation.vue (herefter kaldet entry point) som ligger i *src/components/Aplikation.vue*
7. Fjern sandkasse kode der ikke skal med i leverandør-applikationen (Vue komponenter, styles og templates)

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

Hvis der er problemer med at bygge sandkasse-miljøet kan det være Node eller NPM versionen der ikke er kompatibel. Det er udviklet med node version **14.x** og NPM version **6.x** som anbefales.

I *README.md* filen finder man desuden listen af de NPM moduler og versioner som VG kender til i bygningsfasen, og kan derfor tilføjes til *package.json* filen hvis det er nødvendigt.

2.2.1 Version

Sandkasse-miljøet 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 det udelukkende er ressourcer (komponenter, stylesheets etc) der importeres (direkte eller indirekte) fra entry point komponenten (Applikation.vue), der vil blive inkluderet i den færdige leverandør-applikation. Vue rammeværket som ligger rundt om (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;
    parameterværdi: string;
  }[];
}
```

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

Følgende eksempel giver en leverandør-applikation mulighed for at blive vist i 2 forskellige varianter med hver deres variant værdi.

```
[
  {
    "navn": "blå",
    "aktiv": true,
    "parametre": [
      {
        "parameternavn": "farve",
        "parameterværdi": "blue"
      }
    ]
  },
  {
    "navn": "rød",
    "aktiv": true,
    "parametre": [
      {
        "parameternavn": "farve",
        "parameterværdi": "red"
      }
    ]
  }
]
```

En variant kan skjules ved at ændre aktiv til false.

2.2.4 Side navigation i leverandør-applikation

Nogle gange er der brug for at vise forskellige sider i leverandør-applikationen ved fx. en bruger som klikker sig igennem forskellige trin, og derved får vist indhold der passer til trinnet. Det er leverandør-applikationens eget ansvar at simulere denne form for side navigation hvilket ofte kan implementeres vha. Vues *v-show* og *v-if* direktiver.

Har slutbrugeren påbegyndt side navigation inde i leverandør-applikationen, er det vigtigt browserens tilbage-knap kan klikkes uden slutbrugeren forlader leverandør-applikationen blot fordi URL'en skifter. For at undgå dette kan leverandør-applikationen sørge for der angives et fragment (hash #) i slutningen af URL'en hver gang der foretages side navigation.

Klikker slutbrugeren på tilbage-knappen når URL'en indeholder et fragment, så vil den enten skifte til forrige fragment eller helt fjerne det afhængig af navigations-historikken. Resultatet er det samme, slutbrugeren bliver i leverandør-applikationen og navigerer ikke væk fra den.

Ved at registrere en event listener på *hashchange* kan leverandør-applikationen selv styre hvad der skal ske. Der er ingen krav til selve værdien af fragmentet så længe den er unik, og værdien bestemmes naturligvis af leverandør-applikationen selv.

```
created() {
  window.addEventListener('hashchange',
    funktionDerHåndtererFragmentÆndring);
},
destroyed() {
  window.removeEventListener('hashchange',
    funktionDerHåndtererFragmentÆndring);
}
```

Se koden i sandkasse-miljøet som viser et simpelt eksempel på trin navigation.

2.2.5 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.

Det er her leverandøren angiver URL'en (Review URL) til den leverandør-applikation, der skal importeres i VG. Formatet på URL'en afhænger af hvilken Git host der anvendes (se afsnit 2.4.1 eller 2.4.2). Det er tilladt at importere flere gange for at se hvordan evt. ændringer opfører sig i samspil med VG.

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 i VG, og dette bør afprøves løbende og inden den sendes til godkendelse. Ved preview vælges den første variant som default, men der er mulighed for at skifte variant vha. en dropdown menu med de mulige varianter.

For login henvises til Afsnit 4.

2.3 Godkendelse af ny applikation

Når leverandøren er færdig med at udvikle og teste sin applikation, så skal der angives en endelig 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}/${PROJECT}/commit/${COMMIT_SHA}
```

Eksempel:

```
REPO: Erhvervsfremmeplatformen  
PROJECT: tredjepartsapps  
ACCESS_TOKEN: stzJfj1Rp5TgW1Jc5T3  
COMMIT_SHA: 408c2ea58421e9c1335d5953f50d6caed11af258
```

Leverandør skal angive denne URL som review URL:

<https://github.com/Erhvervsfremmeplatformen/tredjepartsapps/commit/408c2ea58421e9c1335d5953f50d6caed11af258>

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

<https://stzJfj1Rp5TgW1Jc5T3@github.com/Erhvervsfremmeplatformen/tredjepartsapps/commit/408c2ea58421e9c1335d5953f50d6caed11af258>

Bemærk URL'en skal pege på et eksisterende commit SHA (fx. 408c2ea58421e9c1335d5953f50d6caed11af258), 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://${USERNAME?:ACCESS_TOKEN+@}${GITLAB_HOST}/git/${REPO}/${PROJECT}/-/commit/${COMMIT_SHA}
```

Eksempel:

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

Leverandør skal angive denne URL som review URL:

<https://gitlab.com/git/vgapplication/app1/-/commit/1f7f1b71c5961df99e5131572d87a31f4d9edc98/>

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

<https://username:stzJfj1Rp5TgW1Jc5T3@gitlab.com/git/vgapplication/app1/-/commit/1f7f1b71c5961df99e5131572d87a31f4d9edc98/>

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

2.4.3 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/Applikation.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

Sandkasse-miljøet 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 Designsystem (DKFDS - <https://designsystem.dk/>). Det Fælles Designsystem 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@8.0.1** 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.

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/Applikation.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, der skal være fixed dvs. der må ikke bruges ^, ~ eller lignende
- Typescript declaration file (hvis nødvendigt)

kompatibilitet

Bemærk, det kan ikke garanteres et NPM modul vil virke i VG selvom det virker korrekt i sandkasse-miljøet. Derfor en anbefaling til at teste det tidligt i forløbet, så der kan findes et alternativ.

Vær desuden opmærksom på, at for hvert NPM modul der anvendes, vokser størrelsen på leverandør-applikationen, og vil derfor påvirke performance ved visning i VG. Det skal derfor grundigt overvejes om alle NPM moduler er nødvendige.

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.

Bemærk følgende regler er ikke udtømmende, og review processen kan derfor afsløre behov for nye regler, der kan forårsage at et review afvises.

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.$root`, `this.$parent` eller anden metode til at tilgå forældre komponenter i Vue udenfor leverandør-applikationen.

3.4.3 Ingen DOM operationer udenfor leverandør-applikationen

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`
- `window.addEventListener`
- `window.removeEventListener`

3.4.4 Alle eksterne links skal bruge target="_blank"

Alle eksterne 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 Styles skal være scoped

`<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>

```

Se evt. kode i sandkasse-miljøet til håndtering af NPM styles, som ikke umiddelbart virker med scoped attributten.

```

>>> .my-multiselect {
  @import 'vue-multiselect/dist/vue-multiselect.min';
}

```

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" />

```

3.4.10 Event listeners

Alle event listeners, der registreres i leverandør-applikationen skal af-registreres når leverandør-applikationen ikke længere skal vises, hvilket passende kan gøres i life cycle metoden *destroyed()*

3.4.11 Logning

Der bør ikke være nogen logning til konsollen med mindre der er en god grund.

3.4.12 Miljøvariabler

Det er ikke teknisk muligt at benytte miljøvariabler, da *.env filer ikke inkluderes i den færdige leverandør-applikation der vises i VG.

3.4.13 Vue plugins

Det er ikke tilladt at bruge Vue plugins, da det kan påvirke VG og andre leverandør-applikationer. Der er dog undtagelser, som angives i dette afsnit. Bemærk, det kan være muligt andre Vue plugins kan godkendes senere, men det vil blive vurderet for hvert plugin.

3.4.12.1 Vuex

VG bruger vuex (<https://vuex.vuejs.org/>), så derfor er Vuex også tilladt at bruge i leverandør-applikationer til state management. Bemærk, der er et uløst problem mht. reaktivitet når leverandør-applikation afvikles i VG. Se koden i sandkasse-miljøet for detaljer om problemet og hvordan det evt. kan håndteres.

4 Login

VG tilbyder, at leverandør-applikationer nemt kan implementere login-funktionalitet, og på den måde personalisere bruger-oplevelsen.

Når en bruger er logget ind i VG, kan en leverandør-applikation få adgang til denne bruger samt en access token, som leverandøren kan bruge til at identificere brugeren ved kald til egne API'er.

Udover nedenstående henvises der til sandkasse miljøet som beskrives i afsnit 2.1.4 for information om login løsningen, hvor der både findes mere dokumentation og kode eksempel på håndtering af login flowet.

Følgende er en liste af de props der bruges, og sendes ind fra VG til leverandør-applikationen.

4.1 Props

Prop erty	Beskrivelse	Eksempel
isLoggedIn	Boolean flag, som angiver om brugeren er logget ind i VG eller ej.	true
bruger	JavaScript objekt med information om brugeren som fx. navn, entitets-ID etc.	<pre>{ "navn": "Jens Hansen", "organisation": "Demo Nation", "virksomhedsnavn": "Business Demo", "cvr": 12345678, "entityId": "eid-CVR:12345678- RID:e4f13c3b-3c5a-459d-90a9- 847ab9596157" }</pre>
token	En <i>access token</i> fra JWT som leverandør-applikationen bla. kan bruges til at kalde beskyttede API endpoints i Erhvervsstyrelsen eller som access control i leverandørens egne API'er. Den skal angives som <i>bearer token</i> i <i>Authorization</i> headeren i API requests.	<pre>eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJBQWVhZG91WUUG0Zko3MHVsb2YyVzNDZEHqVGsiLCJleHAiOiJlE2NTY1OQ0NTQsImhhdCI6MTY1NjU5MDg1Nn0.fTsRzum7cs ejJF6KLh8Zh23h3voQj9t6_LiT8K_VlStGk4lh6UnapH1q gxpNjHo9hO4laeFk9GUIHMYEKNjElmHm81XMrluZg SgZ947AY3sY3WLBbbZfp6Hhk1U8uwq3- r3MzxebgoaLD8UYDtXXiH0k6G52HrgYO1yXqqXxxzI8 ChpwUf2XlBl- 9hllX8WYtllS2SveljXHjxb5B_2co41mbZiwq- _1fpoTyhaj7S6jRlxDLQ4brMPuLVVmibJ9eqcwluKW8- pUs1Uym84fR5wE9KwOJjzWNbY-_1JkEl- Ubn0JUeTXQY7gmqMy7YracrLla7GayMcfpJogQlg</pre>

Prop tilgængelighed

Uanset om brugeren er logget ind eller ej (isLoggedIn), så er *token* og *bruger* prop aldrig udfyldt til at starte med. De bliver først tilgængelig når leverandør-applikationen eksplicit anmoder om token.

4.2 Events

I forbindelse med login, så lytter VG parent komponent for leverandør-applikation efter følgende custom events

Navn	Beskrivelse	Eksempel
requestToken	leverandør-applikation anmoder om access token, VG udfylder token prop eller starter login flow hvis brugeren ikke allerede er logget ind	<code>this.\$emit('requestToken')</code>

4.3 Konfiguration

Før en leverandør-applikation kan anvende login funktionalitet, skal den først aktiveres og konfigureres i administration af leverandør-applikationer.

Login aktiveres ved at krydse af i checkbox med titlen *Applikationen anvender bruger login*, som nu folder 2 nye indtastningsfelter ud, der skal udfyldes.

4.3.1 Sletning af brugerdata

Feltet *URL til kald for sletning af brugerdata* skal udfyldes med en valid URL. Denne URL ejes af leverandøren og er et API endpoint, som VG kalder når en bruger ønsker at få fjernet alt data der er registreret på brugeren i leverandør-applikationen. Endpoint skal kunne modtage et **POST** request med følgende parametre:

Parameter	Beskrivelse	Eksempel
entitetsId	Entitets ID på brugeren, som er logget ind i VG	<code>eid-CVR:12345678-RID:e4f13c3b-3c5a-459d-90a9-847ab9596157</code>
apiKey	API key, som leverandøren selv bestemmer. Denne kan bruges til at kontrollere API requests, så API endpoint ikke nemt kan misbruges	<code>mYV3ryS3cr3T4P1K3y</code>

API endpoint skal svare med **2xx** success HTTP status codes (fx. 200 OK) som kvittering. Asynkron processering er derfor også tilladt.

Hvis VG ikke modtager en 2xx HTTP status code fra API endpoint, så vil VG kalde API endpoint periodisk de næste 48 timer indtil der kommer et 2xx svar.

4.3.2 API key

Feltet *API-key til ovenstående* skal udfyldes med en værdi, som leverandøren selv bestemmer. Denne API key sendes med som POST parametre til leverandørens API endpoint. API key kan bruges til at begrænse API adgang eller personalisere, men det er leverandørens eget ansvar at implementere.

4.4 Forside

Når en leverandør-applikation, som har aktiveret login, vises i VG, skal brugeren først se en forside der introducerer leverandør-applikationen samt minimum have en knap som starter den del der kræver login. Det er et krav, at forsiden på leverandør-applikationen skal kunne vises og fungere uden brugeren er logget ind.

4.4.1 Logget ind

Hvis brugeren allerede er logget ind i VG, så bliver token udfyldt når brugeren klikker på start knappen der anmoder om token via event. Se afsnit 4.2

4.4.2 Ikke logget ind

Hvis brugeren endnu ikke er logget ind i VG når der klikkes på knappen på forsiden, initieres der et login flow igennem VG. Efter succesfuldt login vises leverandør-applikationen igen, hvor token nu er udfyldt. Det er leverandør-applikationens ansvar at sørge for forsiden med start knappen ikke længere vises, men skal i stedet vise den første side det ligger bag login.