

## EUWA Wrapper Interface

**EUWA Wrapper Interface** (også kalt *wrapper* nedenfor) gir et enkelt inngangspunkt og kommunikasjon for sluttbrukerens webapplikasjoner (EUWA). Det lar utviklerne starte EUWA raskt og kommunisere med dem, uten å administrere innlastingen av applikasjonene, for eksempel hvis du vil starte chat med en egendefinert knapp eller legge til konfigurasjon / variabler (kjøretid) fra nettstedet ditt.

Pakken er publisert i vårt offentlige NPM-register som `@puzzel/euwa-wrapper`

## Grensesnitt

```
declare interface Config {
  customerKey: string,
  configId: string
}

declare interface Options {
  settings: ApplicationSettings,
  hooks: Hooks
}

declare interface ApplicationList {
  [app: string]: string
}

declare interface Hooks {
  [hook: string]: Function
}

declare interface ApplicationBridge {
  api: ApplicationAPI,
  publish: (event: string, ...data: any) => void,
  subscribe: (event: string, callback: Function) => void,
}

declare interface ApplicationSettings {
  [app: string]: object
}

declare interface ApplicationAPI {
  [method: string]: Function
}

declare class EUWA {
  static APPS: ApplicationList

  constructor({customerKey, configId}: Config, {settings, hooks}: Options);
  getApplication(id: string): Promise<ApplicationBridge>;
  getApplicationBeforeLoad(id: string): ApplicationBridge;
}
```

## Koble til NPM-registeret

En `.npmrc` fil skal opprettes enten i prosjektet eller på brukernivå. Les mer om `.npmrc` på <https://docs.npmjs.com/cli/v6/configuring-npm/npmrc>

Følgende linjer skal legges til:

```
@puzzel:registry=https://puzzel.pkgs.visualstudio.com/public/_packaging/main/npm/registry/ always-auth=true
```

## Grunnleggende bruk

Den grunnleggende bruken vil laste EUWA , med konfigurasjon satt fra Puzzels administrasjonsportal.

```
import { EUWA } from '@puzzel/euwa-wrapper';  
new EUWA({ configId: 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx', customerKey: 123456 });
```

## API-bruk

Instansering av EUWA-klassen vil gi en enkel API, som gjør det mulig å få tilgang til alle sluttbrukerens webapplikasjonskontekster.

```
import {EUWA} from '@puzzel/euwa-wrapper';  
const euwa = new EUWA({  
  configId: 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx',  
  customerKey: 123456  
});  
  
// Subscribe to click event on your start button  
document.querySelector('#your-start-button').addEventListener('click', async () => {  
  // Get the chat application context  
  const chat = await euwa.getApplication(EUWA.APPS.CHAT);  
  
  // Use the Chat's API to retrieve it's state  
  const state = chat.api.getState();  
  
  // Start a chat, if the user is not already in session  
  if (!state.isConnected)  
    { chat.api.startChat();  
  }  
});
```

## Bruke kroker

onBeforeLoad

Dette lar deg abonnere på hendelser eller gjøre andre handlinger før applikasjonene lastes inn. Men siden de forskjellige applikasjonens API-er er definert av selve applikasjonen, vil de ikke være tilgjengelige - bare publiser / abonner-grensesnitt er tilgjengelig.

### Note

EUWAs `getApplicationBeforeLoad`-metode er spesielt designet for å brukes med spesifikt denne kroken. Det vil ikke vente på applikasjonens belastning og returnere det grunnleggende kommunikasjonsgrensesnittet - hendelser.

```
import {EUWA} from '@puzzel/euwa-wrapper';  
  
const euwa = new EUWA({  
  configId: 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx',  
  customerKey: 123456  
}), {  
  hooks: {  
    // All hooks accept functions  
    onBeforeLoad: subscribeToChatInit  
  }  
});  
  
function subscribeToChatInit() {  
  // Get the Chat's event interface  
  
  const chat = euwa.getApplicationBeforeLoad(EUWA.APPS.CHAT);  
  // Subscribe to chatInit* event  
  chat.subscribe('chatInit', data => {  
    console.log('Chat Init Data:', data);  
  });  
}
```

\* En komplett liste over hendelser finner du i vår API-artikkel om Chat Front End.

## Overstyrende innstillinger

```
import {EUWA} from '@puzzel/euwa-wrapper';

const euwa = new EUWA({
  configId: 'xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx',
  customerKey: 123456
}, {
  settings: {
    // Application name as first-level property
    [EUWA.APPS.CHAT]: {
      // Properties names as listed in Chat Admin
      showForm: false
    }
  }
});
```