

Remote module

Assets in the remote module

Version 8.0



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Table of Contents

Assets in the remote module	4
Use a custom font	4
Use a custom icon	6
Use a custom image	9
Use a custom background image	11

Assets in the remote module



Creatio lets you use assets in the remote module.

Assets include the following:

- fonts
- icons (*.svg files)
- images (*.png, *.jpg, *.jpeg, *.gif files)
- background images (images connected via the `background-image` CSS property)

Do not use assets in the remote module in the following **situations**:

- You do not have access to the source code of the remote module.
- Configuring the module properties via incoming properties is not possible because the developer of the remote module does not provide this capability.

You can use assets in the remote module in other situations. Connect assets either directly (`CDN` or `Base64`) or via an incoming property. The asset format depends on the Angular project file where the user adds the asset. If you connect the required asset in the `Base64` format, the user can use the remote module without access to the required asset on the Internet. View detailed instructions on how to use the assets below.

You can customize assets connected via an incoming property more flexibly than `CDN` or `Base64` assets connected directly . If incoming properties are provided, interact with them via the `CrtInput` decorator. In this case, set the property value via an attribute or in the Freedom UI page schema. View an example below.

Example that connects the asset via an incoming property

```
<usr-view-element></usr-view-element>
...
const viewElement = document.querySelector('usr-view-element');
const base64Image = 'data:image/png;base64,...';
viewElement.setAttribute('image', base64Image)
```

Use a custom font

You can connect a custom font to a remote module in the following **ways**:

- `CDN` or `Base64` directly
- `CDN` or `Base64` via the incoming property

To use a custom font in a remote module (`CDN`):

1. Connect the font in the `index.html` component file.

Example of the `index.html` file

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Dancing+Script&display=s
```

2. Apply the font.

To use a custom font in a remote module (`Base64`):

1. Connect the font in the `src` property of the `*.component.scss` file.

Example of the `*.component.scss` file

```
@font-face {
  font-family: "Menlo";
  src: url("data:font/ttf;base64, ...")
}
```

2. Apply the font in the `*.component.html` file.

Example of the `*.component.html` file

```
p {
  font-family: "Menlo", cursive;
  color: green;
  font-size: 36px;
}
```

To use a custom font in a remote module (`CDN` or `Base64` via the incoming property):

1. Flag the incoming field using the `CrtInput` decorator.
2. Implement the business logic that installs and loads a custom font.

Example that implements the business logic

```
private _fontFamily!: string;
private _fontSrc!: string;

@CrtInput()
@Input()
public set fontConfig(config: { fontFamily: string; fontSrc: string }) {
  if (Boolean(config)) {
    this._fontFamily = config.fontFamily;
    this._fontSrc = config.fontSrc;
  }
}
```

```

const font = new FontFace(this._fontFamily, 'url(${this._fontSrc})');
font.load().then(() => {
  this._renderrer.setStyle(
    document.querySelector('p'),
    'fontFamily',
    this._fontFamily
  );
});
}
}

```

3. Add the needed configuration object that contains the custom font to the Freedom UI page schema. If the Freedom UI page schema of other component uses the remote module, set up the properties via the component attributes.

Example of a configuration object (CDN)

```

"type": "usr.viewElement",
"fontConfig": {
  "fontFamily": "FontFamily Style Bitter",
  "fontSrc": "https://fonts.gstatic.com/s/bitter/v7/HEpP8tJXlWaYHimsnXgfC0vvDin1pK8aKtelpeZ"
}

```

Example of a configuration object (Base64)

```

"type": "usr.viewElement",
"fontConfig": {
  "fontFamily": "FontFamily Style Bitter",
  "fontSrc": "data:application/octet-stream;base64, ..."
}

```

Use a custom icon

You can connect a custom icon to a remote module in the following **ways**:

- `CDN` or `Base64` directly
- `CDN` or `Base64` via the incoming property

To use a custom icon in a remote module (`CDN` **or** `Base64`):

1. Add the `DomSanitizer` abstract class to the component constructor. Learn more about the `DomSanitizer` abstract class in the official [Angular documentation](#).

Example of a constructor

```
constructor(
  private readonly _renderrer: Renderer2,
  private readonly _domSanitizer: DomSanitizer
) {}
```

In this case, it can display an error about the unsafe URL address of the custom icon.

2. Add the URL of the needed `CDN` or `Base64` custom icon to the `bypassSecurityTrustResourceUrl()` method.

Example of the `bypassSecurityTrustResourceUrl()` method (CDN)

```
public svgImage: string = this._domSanitizer.bypassSecurityTrustResourceUrl(
  'https://www.svgrepo.com/show/303233/icon.svg'
) as string;
```

Example of the `bypassSecurityTrustResourceUrl()` method (Base64)

```
public svgImage: string = this._domSanitizer.bypassSecurityTrustResourceUrl(
  'data:image/svg+xml;base64,...'
) as string;
```

3. Connect the icon.

You can connect the icon in the following **ways**:

- via the `` tag
- via the `<object>` tag

Example that connects an icon (`` tag)

```
<img [src]="svgImage" alt="svgImage" />
```

Example that connects an icon (`<cke:object>` tag)

```
<object type="image/svg+xml" [data]="svgImage">svg image</object>
```

To use a custom icon in a remote module (`CDN` or `Base64` **via the incoming property**):

1. Add the `DomSanitizer` abstract class to the component constructor.

Example of a constructor

```
constructor(
  private readonly _rendererr: Renderer2,
  private readonly _domSanitizer: DomSanitizer
) {}
```

In this case, can display an error about the unsafe URL address of the custom icon.

2. Flag the field that contains the custom icon using the `CrtInput` decorator.
3. Add the URL of the needed `CDN` or `Base64` custom icon to the `bypassSecurityTrustResourceUrl()` method in.

Example of the `bypassSecurityTrustResourceUrl()` method

```
@CrtInput()
@Input()
public set svgImage(value: string){
  if (Boolean(value)) {
    this._svgImage = this._domSanitizer.bypassSecurityTrustResourceUrl(
      value
    ) as string;
  }
}

public get svgImage(): string {
  return this._svgImage;
}
```

4. Connect the icon.

You can connect the icon in the following **ways**:

- via the `` tag
- via the `<object>` tag

Example that connects an icon (`` tag)

```
<ng-container *ngIf="svgImage">
  <img [src]="svgImage" alt="svgImage" />
</ng-container>
```

Example that connects an icon (`<cke:object>` tag)


```
<ng-container *ngIf="svgImage">
  <object type="image/svg+xml" [data]="svgImage" >svg image</object>
</ng-container>
```

5. Add the needed configuration object that contains a custom icon to the Freedom UI page schema. Set up the properties via the component attributes using a wrapper.

Example of a configuration object

```
"type": "usr.viewElement",
"svgImage": "https://www.svgrepo.com/show/303233/icon.svg"
```

Use a custom image

You can connect a custom image to a remote module in the following **ways**:

- `CDN` or `Base64` directly
- `CDN` or `Base64` via the incoming property

To use a custom image in a remote module (`CDN` **or** `Base64`):

1. Add an image to the `imageUrl` attribute.

Example of the `imageUrl` attribute (CDN)

```
export class ImageComponent {
  public imageUrl = 'https://upload.wikimedia.org/wikipedia/commons/thumb/4/47/picture.png'
}
```

Example of the `imageUrl` attribute (Base64)

```
export class ImageComponent {
  public imageUrl = 'data:image/png;base64,...'
}
```

2. Connect the image to the `[src]` attribute of the `` tag.

Example of the `` tag

```
<img [src]="imageUrl" alt="image" />
```

To use a custom image in a remote module (`CDN` or `Base64` **via the incoming property**):

1. Flag the incoming parameter using the `CrtInput` decorator.

Example of an incoming parameter

```
export class ImageComponent {
  @CrtInput()
  @Input()
  public imageUrl!: string;
}
```

2. Set the `imageUrl` property value in the Freedom UI page schema. Creatio lets you set a property value from the page schema and the component wrapper.

Example of the imageUrl property (CDN)

```
"values": {
  "type": "usr.viewElement",
  "imageUrl": "https://upload.wikimedia.org/wikipedia/commons/thumb/4/47/picture.png"
}
```

Example of the imageUrl property (Base64)

```
"values": {
  "type": "usr.viewElement",
  "imageUrl": "data:image/png;base64,..."
}
```

Use the attribute that contains the property name to set the `Base64` image in the wrapper in .

Example that sets an image (the `Base64` property)

```
<usr-view-element></usr-view-element>
...
const viewElement = document.querySelector('usr-view-element');
const base64Image = 'data:image/png;base64,...';
viewElement.setAttribute('image', base64Image)
```

Use a custom background image

You can connect a custom background image in a remote module in the following **ways**:

- `CDN` or `Base64` directly
- `CDN` or `Base64` via the incoming property

To use a custom background image in a remote module (`CDN` **or** `Base64`):

1. Find the needed element after the `view` model initialization.
2. Connect the background image in the remote module.

Example that connects image (CDN)

```
public ngAfterViewInit(): void {
  this._renderer.setStyle(
    document.querySelector('p'),
    'background-image',
    'url(https://upload.wikimedia.org/wikipedia/commons/thumb/4/47/picture.png)'
  );
}
```

Example that connects image (Base64)

```
public ngAfterViewInit(): void {
  this._renderer.setStyle(
    document.querySelector('p'),
    'background-image',
    'url(data:image/png;base64,...)'
  );
}
```

To use a custom background image in a remote module (`CDN` **or** `Base64` **via the incoming property**):

1. Flag the incoming parameter using the `CrtInput` decorator.
2. Set the CSS style when receiving the incoming parameter value.

Example that sets a CSS style

```
@CrtInput()
@Input()
public set imageUrl(value) {
  if (Boolean(value)) {
    this._renderer.setStyle(
```

```
        document.querySelector('p'),  
        'background-image',  
        'url(${value})'  
    );  
    }  
}
```

3. Specify the background image URL in the Freedom UI page schema.

Example of a background image URL (CDN)

```
"type": "usr.viewElement",  
"imageUrl": "https://upload.wikimedia.org/wikipedia/commons/thumb/4/47/picture.png"
```

Example of a background image URL (Base64)

```
"type": "usr.viewElement",  
"imageUrl": "data:image/png;base64,..."
```