aws

Guia do SDK desenvolvedor do Web Client

NICE DCV

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

NICE DCV: Guia do SDK desenvolvedor do Web Client

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o SDK do cliente Web do NICE DCV? ... 1
Pré-requisitos ... 1
Atributos compatíveis ... 2
Suporte ao navegador ... 2
Convenção de versionamento ... 3
Conceitos básicos ... 4
Conecte-se a um servidor do NICE DCV e obtenha o primeiro quadro ... 5
Etapa 1: Preparar a página HTML ... 5
Etapa 2: Autenticar, conectar e obter o primeiro quadro ... 6
Bônus: crie automaticamente um formulário de login em HTML ... 9
Trabalhar com os atributo do NICE DCV ... 10
Entender a função de retorno de chamada featuresUpdate ... 10
Controlar atualizações do atributo ... 11
Usar o SDK da interface de usuário da Web do NICE DCV ... 11
Pré-requisitos ... 12
Etapa 1: Preparar a página HTML ... 13
Etapa 2: Autenticar, conectar e renderizar o componente React do `DCVViewer`. ... 13
Atualizar o AWS-UI para o Cloudscape Design System ... 18
Referência do SDK ... 19
Módulo DCV ... 19
Métodos ... 19
Membros ... 22
Definições de tipo e retorno de chamada ... 27
Classe de conexão ... 68
Métodos ... 19
Classe de autenticação ... 95
Métodos ... 19
Classe de recurso ... 96
Métodos ... 19
SDK da interface de usuário da Web do NICE DCV ... 97
Componentes ... 98
Notas de versão e histórico de documentos ... 106
Notas de versão ... 106
1.5.6 — 9 de novembro de 2023 ... 107

1.4.4 — 29 de junho de 2023 107
1.4.0 — 28 de março de 2023 108
1.3.1 — 9 de dezembro de 2022 109
1.3.0 — 11 de novembro de 2022 110
1.2.1 — 21 de julho de 2022 111
1.2.0 — 29 de junho de 2022 111
1.1.3 — 23 de maio de 2022 112
1.1.2 — 19 de maio de 2022 112
1.1.1 — 23 de março de 2022 113
1.1.0 — 23 de fevereiro de 2022 113
1.0.4 — 20 de dezembro de 2021 114
1.0.3 — 1º de setembro de 2021 115
1.0.2 — 30 de julho de 2021 115
1.0.1 — 31 de maio de 2021 116
1.0.0 — 24 de março de 2021 116
Histórico do documento 117
..... cxix

O que é o SDK do cliente Web do NICE DCV?

O NICE DCV é um protocolo de exibição remota de alto desempenho. Ele permite oferecer desktops remotos e streaming de aplicativos, de forma segura, de qualquer nuvem ou datacenter para qualquer dispositivo, em diferentes condições de rede. Ao usar o NICE DCV com o Amazon EC2, é possível executar aplicativos com alto consumo de gráficos remotamente em instâncias do Amazon EC2. Depois, é possível fazer streaming dos resultados para computadores cliente mais modestos, o que elimina a necessidade de estações de trabalho dedicadas e dispendiosas.

O SDK do Cliente Web do NICE DCV é uma biblioteca JavaScript que você pode usar para desenvolver seus próprios aplicativos cliente do navegador do NICE DCV. Seus usuários finais podem usar esses aplicativos para se conectar e interagir com uma sessão do NICE DCV em execução.

Usando o SDK do Cliente Web do NICE DCV como um componente básico, você pode criar aplicativos Web personalizados que fornecem aos usuários acesso instantâneo a seus desktops ou aplicativos de qualquer lugar, com um desempenho responsivo e fluido que é quase indistinguível de um aplicativo instalado nativamente.

Este guia explica como usar o SDK do Cliente Web do NICE DCV para criar seus aplicativos cliente do navegador da Web personalizados para interagir com as sessões do NICE DCV em seus fluxos de trabalho.

Tópicos

- Pré-requisitos
- Atributos compatíveis
- Suporte ao navegador
- Convenção de versionamento

Pré-requisitos

Antes de começar a trabalhar com o SDK do Cliente Web do NICE DCV, certifique-se de estar familiarizado com o NICE DCV e com as sessões do NICE DCV. Para obter mais informações, consulte o Guia do Administrador do NICE DCV.

O SDK do Cliente Web do NICE DCV é compatível com o servidor do NICE DCV versão 2020 e posterior.

Atributos compatíveis

Você pode criar aplicativos cliente do navegador da Web personalizados compatíveis com os seguintes atributo do NICE DCV:

- Conectar-se aos servidores Windows NICE DCV
- Conectar-se aos servidores Linux NICE DCV
- Gerenciar modos de streaming
- Transferir arquivos
- Imprimir de sessões
- Copiar e colar
- Reprodução de áudio estéreo 2.0
- Gravação de áudio estéreo 2.0 (em servidores do Windows)
- Tela sensível ao toque
- Caneta (em servidores do Linux, Windows 10 e Windows Server 2019)
- Suporte a vários monitores

Para obter informações sobre esses atributo, consulte Atributos compatíveis no Guia do Usuário do NICE DCV.

Suporte ao navegador

O SDK do Cliente Web do NICE DCV é compatível com JavaScript (ES6) e pode ser usado em aplicativos JavaScript ou TypeScript.

O SDK do Cliente Web do NICE DCV é compatível com os seguintes navegadores da Web:

Navegador	Versão
Google Chrome	Últimas três versões principais
Mozilla Firefox	Últimas três versões principais
Microsoft Edge	Últimas três versões principais

Navegador	Versão
Apple Safari para macOS	Últimas três versões principais

Convenção de versionamento

A versão do SDK do Cliente Web do NICE DCV é definida no seguinte formato: *major*.*minor*.*patch*. Geralmente, a convenção de versionamento segue o modelo de versionamento semântico. Uma alteração na versão principal, como de 1.x.x para 2.x.x, indica que foram introduzidas mudanças importantes que podem exigir alterações no código e uma implantação planejada. Uma alteração em uma versão secundária, como de 1.1.x para 1.2.x, é sempre compatível com versões anteriores, mas pode incluir elementos descontinuados.

Conceitos básicos do SDK do Cliente da Web do NICE DCV

O SDK do Cliente da Web do NICE DCV é composto por um arquivo principal `dcv.js` e alguns componentes auxiliares. Todos os arquivos são distribuídos em um arquivo compactado que pode ser baixado do site do NICE.

Para começar a usar o SDK do Cliente da Web do NICE DCV

1. O arquivo SDK do Cliente da Web do NICE DCV é assinado digitalmente com uma assinatura GPG segura. Para verificar a assinatura do arquivo, você deve importar a chave NICE GPG. Para fazer isso, abra uma janela de terminal e importe a chave GPG do NICE.

   ```
   $ wget https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
   ```

   ```
   $ gpg --import NICE-GPG-KEY
   ```

2. Baixe o arquivo do SDK do Cliente da Web do NICE DCV e a assinatura do arquivo do SDK do Cliente da Web do NICE DCV do site do NICE.
3. Verifique a assinatura do arquivo do SDK do Cliente da Web do NICE DCV usando a assinatura.

   ```
   $ gpg --verify
   signature_filename.zip.sign
   archive_filename.zip
   ```

   Por exemplo:

   ```
   $ gpg --verify nice-dcv-web-client-sdk-1.5.6-659.zip.sign nice-dcv-web-client-
   sdk-1.5.6-659.zip
   ```

4. Se a assinatura for verificada com sucesso, extraia o conteúdo do arquivo SDK do Cliente da Web do NICE DCV e coloque o diretório extraído em seu servidor web. Por exemplo:

   ```
   $ unzip
   archive_filename.zip
   -d /
   path_to
   ```

```
/
server_directory
/
```

> **Important**
>
> - Você deve manter a estrutura de pastas ao implantar o SDK do Cliente da Web do NICE DCV em seu servidor web.
> - Ao usar o SDK de Interface do Usuário do NICE DCV, lembre-se de que o componente React do `DCVViewer` espera que os arquivos EULA.txt e third-party-licenses.txt desse pacote estejam presentes no caminho do URL do servidor web incorporado. O arquivo third-party-licenses.txt deve ser modificado para incluir também o conteúdo do arquivo correspondente do pacote SDK do Cliente da Web do NICE DCV e, possivelmente, qualquer outra informação de licença das bibliotecas usadas pelo aplicativo do usuário consumidor.

Conecte-se a um servidor do NICE DCV e obtenha o primeiro quadro

O tutorial a seguir mostra como preparar sua página HTML para seu cliente da web personalizado, como autenticar e se conectar a um servidor do NICE DCV e como receber o primeiro quadro de conteúdo transmitido da sessão do NICE DCV.

Tópicos

- Etapa 1: Preparar a página HTML
- Etapa 2: Autenticar, conectar e obter o primeiro quadro
- Bônus: crie automaticamente um formulário de login em HTML

Etapa 1: Preparar a página HTML

Em sua página da Web, você deve carregar os módulos JavaScript necessários e adicionar um elemento HTML `<div>` com um `id` válido no qual você deseja que o SDK do Cliente da Web do NICE DCV desenhe o fluxo de conteúdo do servidor do NICE DCV remoto.

Por exemplo:

```
<!DOCTYPE html>
<html lang="en" style="height: 100%;">
  <head>
    <title>DCV first connection</title>
  </head>
  <body style="height: 100%;">
    <div id="root" style="height: 100%;"></div>
    <div id="dcv-display"></div>
    <script type="module" src="index.js"></script>
  </body>
</html>
```

Etapa 2: Autenticar, conectar e obter o primeiro quadro

Esta seção mostra como concluir o processo de autenticação do usuário, como conectar o servidor do NICE DCV e como receber o primeiro quadro de conteúdo do servidor do NICE DCV.

Primeiro, a partir do arquivo `index.js`, importe o SDK do Cliente da Web do NICE DCV. Ele pode ser importado como um módulo de Definição de Módulo Universal (UMD), da seguinte forma:

```
import "./dcvjs/dcv.js"
```

Caso contrário, a partir da versão `1.1.0`, ele também pode ser importado como um módulo ECMAScript (ESM) do pacote correspondente, assim:

```
import dcv from "./dcvjs/dcv.js"
```

Defina as variáveis a serem usadas para armazenar o objeto Authentication, o objeto Connection e o URL do servidor do NICE DCV.

```
let auth,
    connection,
    serverUrl;
```

No carregamento do script, registre a versão do SDK do Cliente da Web do NICE DCV e, no carregamento da página, chame a função `main`.

```
console.log("Using NICE DCV Web Client SDK version " + dcv.version.versionStr);
```

```
document.addEventListener('DOMContentLoaded', main);
```

A função `main` define o nível do registro e inicia o processo de autenticação.

```
function main () {
  console.log("Setting log level to INFO");
  dcv.setLogLevel(dcv.LogLevel.INFO);

  serverUrl = "https://your-dcv-server-url:port/";

  console.log("Starting authentication with", serverUrl);

  auth = dcv.authenticate(
    serverUrl,
    {
      promptCredentials: onPromptCredentials,
      error: onError,
      success: onSuccess
    }
  );
}
```

As funções `promptCredentials`, `error` e `success` são funções obrigatórias de retorno de chamada que devem ser definidas no processo de autenticação.

Se o servidor do NICE DCV solicitar credenciais, a função de retorno de chamada `promptCredentials` receberá o desafio de credencial solicitado do servidor do NICE DCV. Se o servidor do NICE DCV estiver configurado para usar a autenticação do sistema, as credenciais de login deverão ser fornecidas. Os exemplos de código a seguir presumem que o nome de usuário é `my_dcv_user` e que a senha é `my_password`.

Se a autenticação falhar, a função de retorno de chamada `error` receberá um objeto de erro do servidor do NICE DCV.

Se a autenticação for bem-sucedida, a função de retorno de chamada `success` receberá uma matriz de pares que inclui o ID da sessão (`sessionId`) e os tokens de autorização (`authToken`) para cada sessão à qual o usuário `my_dcv_user` pode se conectar no servidor do NICE DCV. O exemplo de código a seguir chama a função connect e se conecta à primeira sessão retornada na matriz.

> **Note**
>
> No exemplo de código a seguir, substitua `MY_DCV_USER` por seu próprio nome de usuário e `MY_PASSWORD` por sua própria senha.

```
function onPromptCredentials(auth, challenge) {
  // Let's check if in challege we have a username and password request
  if (challengeHasField(challenge, "username") && challengeHasField(challenge,
 "password")) {
    auth.sendCredentials({username: MY_DCV_USER, password: MY_PASSWORD})
  } else {
    // Challenge is requesting something else...
  }
}

function challengeHasField(challenge, field) {
  return challenge.requiredCredentials.some(credential => credential.name === field);
}

function onError(auth, error) {
  console.log("Error during the authentication: " + error.message);
}

// We connect to the first session returned
function onSuccess(auth, result) {
  let {sessionId, authToken} = {...result[0]};

  connect(sessionId, authToken);
}
```

Conecte-se ao servidor do NICE DCV. O método de retorno de chamada `firstFrame` será chamado quando o primeiro quadro for recebido do servidor do NICE DCV.

```
function connect (sessionId, authToken) {
  console.log(sessionId, authToken);

  dcv.connect({
    url: serverUrl,
    sessionId: sessionId,
    authToken: authToken,
    divId: "dcv-display",
```

```
    callbacks: {
      firstFrame: () => console.log("First frame received")
    }
  }).then(function (conn) {
    console.log("Connection established!");
    connection= conn;
  }).catch(function (error) {
    console.log("Connection failed with error " + error.message);
  });
}
```

Bônus: crie automaticamente um formulário de login em HTML

O objeto `challenge` é retornado quando a função de retorno de chamada `promptCredentials` é chamada. Ele inclui uma propriedade chamada `requiredCredentials` que é uma matriz de objetos - um objeto por credencial que é solicitado pelo servidor do NICE DCV. Cada objeto inclui o nome e o tipo da credencial solicitada. Você pode usar os objetos `challenge` e `requiredCredentials` para criar automaticamente um formulário de login em HTML.

O exemplo de código a seguir mostra como fazer isso.

```
let form,
    fieldSet;

function submitCredentials (e) {
  var credentials = {};
  fieldSet.childNodes.forEach(input => credentials[input.id] = input.value);
  auth.sendCredentials(credentials);
  e.preventDefault();
}

function createLoginForm () {
  var submitButton = document.createElement("button");

  submitButton.type = "submit";
  submitButton.textContent = "Login";

  form = document.createElement("form");
  fieldSet = document.createElement("fieldset");

  form.onsubmit = submitCredentials;
  form.appendChild(fieldSet);
```

```
  form.appendChild(submitButton);

  document.body.appendChild(form);
}

function addInput (name) {
  var type = name === "password" ? "password" : "text";

  var inputField = document.createElement("input");
  inputField.name = name;
  inputField.id = name;
  inputField.placeholder = name;
  inputField.type = type;
  fieldSet.appendChild(inputField);
}

function onPromptCredentials (_, credentialsChallenge) {
  createLoginForm();
  credentialsChallenge.requiredCredentials.forEach(challenge =>
 addInput(challenge.name));
}
```

Trabalhar com os atributo do NICE DCV

A disponibilidade dos atributo do NICE DCV depende das permissões configuradas para a sessão do NICE DCV e dos recursos do navegador da web do cliente.

Os atributo disponíveis em uma sessão do NICE DCV são gerenciados pelas permissões que foram especificadas para a sessão. Isso significa que mesmo que um atributo seja compatível com o SDK do Cliente da Web do NICE DCV, o acesso a esse atributo pode ser impedido com base nas permissões definidas pelo administrador da sessão. Para obter mais informações, consulte Configurar a autorização do NICE DCV no Guia do administrador do NICE DCV.

Entender a função de retorno de chamada featuresUpdate

Quando a disponibilidade de um atributo em uma sessão do NICE DCV muda, o SDK do NICE DCV Web Client notifica você usando a função de retorno de chamada `featuresUpdate` que você especifica no momento do estabelecimento da conexão. Por exemplo:

```
featuresUpdate: function (connection, list) {
  ...
```

```
},
```

A função de retorno de chamada notifica você somente sobre os atributo cuja disponibilidade foi alterada. O parâmetro `list` é uma matriz de strings e inclui somente os nomes dos atributo atualizados. Por exemplo, se a disponibilidade do atributo de entrada de áudio for alterada durante a sessão, o parâmetro incluirá somente `["audio-in"]`. Se, posteriormente, a disponibilidade dos atributo de copiar e colar da área de transferência mudar para a sessão, o parâmetro incluirá somente `["clipboard-copy", "clipboard-paste"]`.

Controlar atualizações do atributo

A função de retorno de chamada `featuresUpdate` notifica você somente sobre os atributos cuja disponibilidade foi alterada. Para saber quais atributos foram atualizados, você deve consultar o atributo usando o método `connection.queryFeature`. Isso pode ser feito a qualquer momento após o recebimento da notificação de alteração. Esse método retorna uma `Promise` que resolve o status atualizado do atributo solicitado. O valor `status` está sempre associado e tem uma propriedade booleana (`true` | `false`) chamada `enabled`. Alguns atributos podem ter propriedades adicionais no valor `status`. Se a disponibilidade do atributo não tiver sido atualizada, ela será rejeitada.

O exemplo de código a seguir mostra como fazer isso.

```
// Connection callback called
function featuresUpdate (_, list) {
  if (list.length > 0) {
    list.forEach((feat) => {
      connection.queryFeature(feat).then(status => console.log(feat, "is",
 status.enabled)));
    });
  }
}
```

Usar o SDK da interface de usuário da Web do NICE DCV

O tutorial a seguir mostra como se autenticar no servidor do NICE DCV, se conectar a ele e renderizar o componente React do `DCVViewer` a partir do SDK da interface de usuário da Web do NICE DCV.

Tópicos

- Pré-requisitos
- Etapa 1: Preparar a página HTML
- Etapa 2: Autenticar, conectar e renderizar o componente React do DCVViewer.
- Atualizar o AWS-UI para o Cloudscape Design System

Pré-requisitos

Você precisa instalar `React`, `ReactDOM`, `Cloudscape Design Components React`, `Cloudscape Design Global Styles` e `Cloudscape Design Design Tokens`.

```
$ npm i react react-dom @cloudscape-design/components @cloudscape-design/global-styles
 @cloudscape-design/design-tokens
```

Você também precisaria baixar de `NICE DCV Web Client SDK`. Consulte Conceitos básicos do SDK do Cliente da Web do NICE DCV para ler o guia passo a passo sobre como fazer isso.

Você deve criar um alias para importar o módulo `dcv`, pois ele é uma dependência externa do SDK da interface de usuário da Web do NICE DCV. Por exemplo, se você estiver usando o webpack para agrupar seu aplicativo da Web, poderá usar a opção resolve.alias da seguinte forma:

```
const path = require('path');

module.exports = {
  //...
  resolve: {
    alias: {
      dcv: path.resolve('path', 'to', 'dcv.js'),
    },
  },
};
```

Se você estiver usando o rollup para empacotamento, poderá instalar o @rollup/plugin-alias e usá-lo assim:

```
import alias from '@rollup/plugin-alias';
const path = require('path');

module.exports = {
```

```
  //...
  plugins: [
    alias({
      entries: [
        { find: 'dcv', replacement: path.resolve('path', 'to', 'dcv.js') },
      ]
    })
  ]
};
```

Etapa 1: Preparar a página HTML

Em sua página da Web, você deve carregar os módulos JavaScript necessários e ter um elemento `<div>` em HTML com um `id` válido no qual o componente de entrada do seu aplicativo será renderizado.

Por exemplo:

```
<!DOCTYPE html>
<html lang="en" style="height: 100%;">
  <head>
    <title>DCV first connection</title>
  </head>
  <body style="height: 100%;">
    <div id="root" style="height: 100%;"></div>
    <script type="module" src="index.js"></script>
  </body>
</html>
```

Etapa 2: Autenticar, conectar e renderizar o componente React do **`DCVViewer`**.

Esta seção mostra como concluir o processo de autenticação do usuário, como conectar o servidor do NICE DCV e como renderizar o componente React do `DCVViewer`.

Primeiro, a partir do arquivo `index.js`, importe `React`, `ReactDOM` e seu componente de nível superior `App`.

```
import React from "react";
import ReactDOM from 'react-dom';
import App from './App';
```

Renderize o nó de contêiner de nível superior do seu aplicativo.

```
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById("root")
);
```

No arquivo `App.js`, importe o SDK do Cliente da Web do NICE DCV como um módulo ESM, o componente React `DCVViewer` do SDK da Interface do Usuário do NICE DCV, `React` e o pacote `Cloudscape Design Global Styles`.

```
import React from "react";
import dcv from "dcv";
import "@cloudscape-design/global-styles/index.css";
import {DCVViewer} from "./dcv-ui/dcv-ui.js";
```

A seguir está um exemplo que mostra como se autenticar no Servidor do NICE DCV e renderizar o componente React do `DCVViewer` a partir do SDK da Interface da Web do NICE DCV, desde que a autenticação tenha sido bem-sucedida.

```
const LOG_LEVEL = dcv.LogLevel.INFO;
const SERVER_URL = "https://your-dcv-server-url:port/";
const BASE_URL = "/static/js/dcvjs";

let auth;

function App() {
  const [authenticated, setAuthenticated] = React.useState(false);
  const [sessionId, setSessionId] = React.useState('');
  const [authToken, setAuthToken] = React.useState('');
  const [credentials, setCredentials] = React.useState({});

  const onSuccess = (_, result) => {
    var { sessionId, authToken } = { ...result[0] };

    console.log("Authentication successful.");

    setSessionId(sessionId);
    setAuthToken(authToken);
    setAuthenticated(true);
```

```
    setCredentials({});
  }

  const onPromptCredentials = (_, credentialsChallenge) => {
    let requestedCredentials = {};

    credentialsChallenge.requiredCredentials.forEach(challenge =>
requestedCredentials[challenge.name] = "");
    setCredentials(requestedCredentials);
  }

  const authenticate = () => {
    dcv.setLogLevel(LOG_LEVEL);

    auth = dcv.authenticate(
      SERVER_URL,
      {
        promptCredentials: onPromptCredentials,
        error: onError,
        success: onSuccess
      }
    );
  }

  const updateCredentials = (e) => {
    const { name, value } = e.target;
    setCredentials({
      ...credentials,
      [name]: value
    });
  }

  const submitCredentials = (e) => {
    auth.sendCredentials(credentials);
    e.preventDefault();
  }

  React.useEffect(() => {
    if (!authenticated) {
      authenticate();
    }
  }, [authenticated]);

  const handleDisconnect = (reason) => {
```

```
    console.log("Disconnected: " + reason.message + " (code: " + reason.code + ")");
    auth.retry();
    setAuthenticated(false);
  }

  return (
    authenticated ?
    <DCVViewer
      dcv={{
        sessionId: sessionId,
        authToken: authToken,
        serverUrl: SERVER_URL,
        baseUrl: BASE_URL,
        onDisconnect: handleDisconnect,
        logLevel: LOG_LEVEL
      }}
      uiConfig={{
        toolbar: {
          visible: true,
          fullscreenButton: true,
          multimonitorButton: true,
        },
      }}
    />
    :
    <div
      style={{
        height: window.innerHeight,
        backgroundColor: "#373737",
        display: 'flex',
        alignItems: 'center',
        justifyContent: 'center',
      }}
    >
      <form>
        <fieldset>
          {Object.keys(credentials).map((cred) => (
            <input
              key={cred}
              name={cred}
              placeholder={cred}
              type={cred === "password" ? "password" : "text"}
              onChange={updateCredentials}
              value={credentials[cred]}
```

```
            />
          ))}
        </fieldset>
        <button
          type="submit"
          onClick={submitCredentials}
        >
          Login
        </button>
      </form>
    </div>
  );
}

const onError = (_, error) => {
  console.log("Error during the authentication: " + error.message);
}

export default App;
```

As funções `promptCredentials`, `error` e `success` são funções obrigatórias de retorno de chamada que devem ser definidas no processo de autenticação.

Se o servidor do NICE DCV solicitar credenciais, a função de retorno de chamada `promptCredentials` receberá o desafio de credencial solicitado do servidor do NICE DCV. Se o servidor do NICE DCV estiver configurado para usar a autenticação do sistema, as credenciais deverão ser fornecidas no formato de um nome de usuário e senha.

Se a autenticação falhar, a função de retorno de chamada `error` receberá um objeto de erro do servidor do NICE DCV.

Se a autenticação for bem-sucedida, a função de retorno de chamada `success` receberá uma matriz de pares que inclui o ID da sessão (`sessionId`) e os tokens de autorização (`authToken`) para cada sessão à qual o usuário pode se conectar no servidor do NICE DCV. O exemplo de código acima atualiza o estado do React para renderizar o componente `DCVViewer` em caso de autenticação bem-sucedida.

Para saber mais sobre as propriedades aceitas por esse componente, consulte a referência do SDK da interface de usuário da Web do NICE DCV.

Para saber mais sobre certificados autoassinados, consulte Esclarecimentos sobre redirecionamento com certificados autoassinados.

Atualizar o AWS-UI para o Cloudscape Design System

A partir da versão 1.3.0 do SDK, atualizamos nosso componente `DCVViewer` do AWS-UI para sua evolução: Cloudscape Design.

O Cloudscape usa um tema visual diferente do AWS-UI, mas a base de código subjacente permanece a mesma. Portanto, migrar seu aplicativo com base no `DCVViewer` deve ser fácil. Para migrar, substitua os pacotes NPM relacionados ao AWS-UI que você instalou pelos pacotes associados do Cloudscape:

Nome do pacote AWS-UI	Nome do pacote Cloudscape
@awsui/components-react	@cloudscape-design/components
@awsui/global-styles	@cloudscape-design/global-styles
@awsui/collection-hooks	@cloudscape-design/collection-hooks
@awsui/design-tokens	@cloudscape-design/design-tokens

Para obter mais detalhes sobre a migração, consulte a página de documentação do AWS-UI no GitHub.

Referência do SDK

Esta seção fornece descrições, sintaxe e exemplos de uso para cada SDK do Cliente Web do NICE DCV.

Tópicos

- Módulo DCV
- Classe de conexão
- Classe de autenticação
- Classe de recurso
- SDK da interface de usuário da Web do NICE DCV

Módulo DCV

Um módulo que implementa o lado do cliente do protocolo DCV.

Expõe

- Métodos
- Membros
- Definições de tipo e retorno de chamada

Métodos

Listar

- authenticate(url, callbacks) → {Autenticação}
- connect(config) → {Promessa.<Conexão>|Promessa.<{code: ConnectionErrorCode, message: string}>}
- setLogHandler(handler) → {void}
- setLogLevel(level) → {void}

authenticate(url, callbacks) → {Autenticação}

Inicia o processo de autenticação para o endpoint do servidor do NICE DCV especificado.

Parâmetros:

Nome	Tipo	Descrição
`url`	string	O nome do host e a porta do servidor do NICE DCV em execução no seguinte formato: `https://dcv_host_address:port` . Por exemplo: `https://my-dcv-server:8443` .
`callbacks`	authenticationCallbacks	Os retornos de chamada que estão disponíveis para serem chamados durante o processo de autenticação.

Retorna:

- O objeto Authentication.

Tipo

Autenticação

connect(config) → {Promessa.<Conexão>|Promessa.<{code: ConnectionErrorCode, message: string}>}

Conecta-se ao endpoint do servidor do NICE DCV especificado. Se a conexão for bem-sucedida, ela retornará um objeto Connection. Se a conexão falhar, ela retornará um objeto de erro.

Parâmetros:

Nome	Tipo	Descrição
`config`	ConnectionConfig	O objeto ConnectionConfig.

Retorna:

- Um objeto Connection ou um objeto de erro.

Tipo

Promessa.<Connection> | Promessa.<{code: ConnectionErrorCode, message: string}>

setLogHandler(handler) → {void}

Define uma função personalizada de manipulador de log. Ao substituir o manipulador de log padrão, a posição original da entrada do log será perdida durante a depuração usando o console do navegador.

Parâmetros:

Nome	Tipo	Descrição
`handler`	função	A função personalizada do manipulador de logs. A função do manipulador contém level (número), levelName (string), domínio (string) e message (string).

Retorna:

Tipo

nulo

setLogLevel(level) → {void}

Define o nível do log. Isso será necessário apenas se o manipulador de logs padrão for usado.

Parâmetros:

Nome	Tipo	Descrição
`level`	LogLevel	O nível de log a ser usado.

Retorna:

Tipo

nulo

Membros

Listar

- (constant) AudioError :AudioErrorCode
- (constant) AuthenticationError :AuthenticationErrorCode
- (constant) ChannelError :ChannelErrorCode
- (constant) ClosingReasonError :ClosingReasonErrorCode
- (constant) ConnectionError :ConnectionErrorCode
- (constant) CustomChannelError :CustomChannelErrorCode
- (constant) DisplayConfigError :DisplayConfigErrorCode
- (constant) FileStorageError :FileStorageErrorCode
- (constant) LogLevel :LogLevel
- (constant) MultiMonitorError :MultiMonitorErrorCode
- (constant) ResolutionError :ResolutionErrorCode
- (constant) TimezoneRedirectionError :TimezoneRedirectionErrorCode
- (constant) TimezoneRedirectionSetting :TimezoneRedirectionSettingCode
- (constant) TimezoneRedirectionStatus :TimezoneRedirectionStatusCode
- (constant) version
- (constant) ScreenshotError :ScreenshotErrorCode
- (constant) WebcamError :WebcamErrorCode

(constant) AudioError :AudioErrorCode

A enumeração dos códigos AudioError.

Tipo:

- AudioErrorCode

(constant) AuthenticationError :AuthenticationErrorCode

A enumeração dos códigos AuthenticationError.

Tipo:

- AuthenticationErrorCode

(constant) ChannelError :ChannelErrorCode

A enumeração dos códigos ChannelError.

Tipo:

- ChannelErrorCode

(constant) ClosingReasonError :ClosingReasonErrorCode

A enumeração dos códigos ClosingReasonError.

Tipo:

- ClosingReasonErrorCode

(constant) ConnectionError :ConnectionErrorCode

A enumeração dos códigos ConnectionError.

Tipo:

- ConnectionErrorCode

(constant) CustomChannelError :CustomChannelErrorCode

A enumeração dos códigos CustomChannelError.

Tipo:

- CustomChannelErrorCode

(constant) DisplayConfigError :DisplayConfigErrorCode

A enumeração dos códigos DisplayConfigError.

Tipo:

- DisplayConfigErrorCode

(constant) FileStorageError :FileStorageErrorCode

A enumeração dos códigos FileStorageError.

Tipo:

- FileStorageErrorCode

(constant) LogLevel :LogLevel

Os níveis de log do SDK disponíveis.

Tipo:

- LogLevel

(constant) MultiMonitorError :MultiMonitorErrorCode

A enumeração dos códigos MultiMonitorError.

Tipo:

- MultiMonitorErrorCode

(constant) ResolutionError :ResolutionErrorCode

A enumeração dos códigos ResolutionError.

Tipo:

- ResolutionErrorCode

(constant) TimezoneRedirectionError :TimezoneRedirectionErrorCode

A enumeração dos códigos TimezoneRedirectionError.

Tipo:

- TimezoneRedirectionErrorCode

(constant) TimezoneRedirectionSetting :TimezoneRedirectionSettingCode

A enumeração dos códigos TimezoneRedirectionSetting.

Tipo:

- TimezoneRedirectionSettingCode

(constant) TimezoneRedirectionStatus :TimezoneRedirectionStatusCode

A enumeração dos códigos TimezoneRedirectionStatus.

Tipo:

- TimezoneRedirectionStatusCode

(constant) version

A versão do NICE DCV com versões principais, secundária, de correção, revisão, estendida e versionStr.

Propriedades:

Nome	Tipo	Descrição
`major`	integer	O número da versão principal.
`minor`	integer	O número da versão secundária.
`patch`	integer	O número da versão de correção.
`revision`	integer	O número da revisão.
`extended`	string	A sequência estendida.
`versionStr`	string	Uma concatenação dos números principais, secundários, de correção e de revisão no formulário `major.minor.patch+build.revision` .

(constant) ScreenshotError :ScreenshotErrorCode

A enumeração dos códigos ScreenshotError.

Tipo:

- ScreenshotErrorCode

(constant) WebcamError :WebcamErrorCode

A enumeração dos códigos WebcamError.

Tipo:

- WebcamErrorCode

Definições de tipo e retorno de chamada

Listar

- AudioErrorCode
- authenticationCallbacks
- AuthenticationErrorCode
- authErrorCallback(authentication, error)
- authPromptCredentialsCallback(authentication, challenge)
- authSuccessCallback(authentication, authenticationData)
- Canal
- ChannelErrorCode
- clipboardEventCallback(event)
- ClosingReasonErrorCode
- Colorspace
- connectionCallbacks
- ConnectionConfig
- ConnectionErrorCode
- createDirectory(path)
- CustomChannelErrorCode
- dataChannelCallback(info)
- deleteFile(path)
- deviceChangeEventCallback()
- disconnectCallback(reason)
- displayAvailabilityCallback(status, displayId)
- DisplayConfigErrorCode
- displayLayoutCallback(serverWidth, serverHeight, heads)
- feature
- featuresUpdateCallback(featuresList)
- fileDownloadCallback(fileResource)
- filePrintedCallback(printResource)

- filestorage
- filestorageEnabledCallback(enabled)
- FileStorageErrorCode
- firstFrameCallback(resizeEnabled, relativeMouseModeEnabled, displayId)
- idleWarningNotificationCallback(disconnectionDateTime)
- collaboratorListCallback(collaborators)
- licenseNotificationCallback(notification)
- list(path)
- LogLevel
- Monitor
- MultiMonitorErrorCode
- qualityIndicatorStateCallback(state)
- renameDirectory(src, dest)
- renameFile(src, dest)
- ResolutionErrorCode
- retrieveFile(path)
- screenshotCallback(screenshot)
- ScreenshotErrorCode
- serverInfo
- stats
- storeFile(file, dir)
- TimezoneRedirectionErrorCode
- TimezoneRedirectionSettingCode
- TimezoneRedirectionStatusCode
- WebcamErrorCode

AudioErrorCode

As enumerações do código AudioError disponíveis no módulo DCV

- `SETTING_AUDIO_FAILED`
- `CHANNEL_NOT_AVAILABLE`

Tipo:

- número

authenticationCallbacks

Retornos de chamada de autenticação

Tipo:

- Objeto

Propriedades:

Nome	Tipo	Descrição
`promptCredentials`	authPromptCredentialsCallback	A função de retorno de chamada a ser chamada quando o usuário é desafiado a fornecer credenciais.
`error`	authErrorCallback	A função de retorno de chamada a ser chamada quando a autenticação falhar.
`success`	authSuccessCallback	A função de retorno de chamada a ser chamada quando a autenticação for bem-sucedida.

AuthenticationErrorCode

As enumerações do código AuthenticationError disponíveis no módulo DCV

- `INVALID_MESSAGE`
- `UNKNOWN_AUTH_MODE`
- `SESSION_NOT_AVAILABLE`
- `NO_SESSIONS`

- `WRONG_CREDENTIALS`
- `SASL_CHALLENGE`
- `SASL_AUTH_MECHANISM`
- `FAILED_COMMUNICATION`
- `AUTHENTICATION_REJECTED`
- `GENERIC_ERROR`
- `WRONG_CREDENTIALS_FORMAT`
- `WRONG_CREDENTIALS_TYPE`
- `UNREQUESTED_CREDENTIALS`
- `MISSING_CREDENTIAL`

Tipo:

- número

authErrorCallback(authentication, error)

A função de retorno de chamada a ser chamada quando a autenticação falhar.

Parâmetros:

Nome	Tipo	Descrição
`authentication`	Autenticação	O objeto Authentication.
`error`	Objeto	O objeto de erro gerado pelo processo de autenticação. Nome / Tipo / Descrição `code` / AuthenticationErrorCode / O código do erro. `message` / string / A mensagem

Nome	Tipo	Descrição
		Nome / Tipo / Descrição de erro.

authPromptCredentialsCallback(authentication, challenge)

A função de retorno de chamada a ser chamada quando o usuário é desafiado a fornecer credenciais. O usuário deve responder ao desafio fornecendo as credenciais solicitadas.

Parâmetros:

Nome	Tipo	Descrição
`authentication`	Autenticação	O objeto Authentication.
`challenge`	Objeto	O desafio. Nome / Tipo / Descrição `require redenti s` / Array.<Object> / Uma matriz de objetos de credenciais solicitados. N / T / Descrição n / s / O nome da

Nome	Tipo	Descrição

Nome	Tipo	Descrição

N	T	D
		Descrição credencial solicitada.
t	s	O tipo da credencial solicitada.

authSuccessCallback(authentication, authenticationData)

A função de retorno de chamada a ser chamada quando a autenticação for bem-sucedida.

Parâmetros:

Nome	Tipo	Descrição
`authentication`	Autenticação	O objeto Authentication.
`authenticationData`	Array.<Object>	Uma matriz de objetos que incluem IDs de sessão do NICE DCV e tokens de autenticação.

Nome	Tipo	Descrição
		Nome: `sessior` — Tipo: string — Descrição: O ID da sessão do NICE DCV. Nome: `authTok` — Tipo: string — Descrição: O token de autenticação para a sessão do NICE DCV.

Canal

Os canais disponíveis que podem ser especificados.

Tipo:

- "clipboard" | "display" | "input" | "audio" | "filestorage"

ChannelErrorCode

As enumerações do código ChannelError disponíveis no módulo DCV

- `ALREADY_OPEN`
- `INITIALIZATION_FAILED`
- `REJECTED`

Tipo:

- número

clipboardEventCallback(event)

A função de retorno de chamada a ser chamada quando a `clipboardEvent` é gerada.

Parâmetros:

Nome	Tipo	Descrição
`event`	Objeto	Informações sobre o evento da área de transferência. Nome: `name` Tipo: established | copy | paste | dataSi lert | autoC one | newDa ailable | autoPa Done | remote or | Atribut: Descrição: Sempre presente. O nome do evento.

Nome	Tipo	Descrição

Nome	Tipo	Atribut	Descrição
	paste/ lableD		
`clipb Data`	Object | string		Os dados na área de transferência.
`autoC`	boolea	<opcic>	Indica se a cópia automática da área de transferência da sessão para a área de transferência do cliente local está ativada.

Nome	Tipo	Descrição
		Nome: `maxDataSize` — Tipo: número — Atributo: <opcional> — Descrição: A quantidade máxima de dados que podem ser colocados na área de transferência. Nome: `error` — Tipo: string — Atributo: <opcional> — Descrição: Informações de erro, se aplicável.

ClosingReasonErrorCode

As enumerações do código ClosingReasonError disponíveis no módulo DCV

- `TRANSPORT_ERROR`
- `NO_ERROR`
- `GENERIC_ERROR`
- `INTERNAL_SERVER_ERROR`
- `PROTOCOL_ERROR`

- `AUTHORIZATION_DENIED`
- `AUTHORIZATION_REVOKED`
- `ACCESS_REJECTED`
- `IDLE_TIMEOUT_EXPIRED`
- `DISCONNECT_BY_OWNER`
- `DISCONNECT_BY_USER`
- `EVICTED`

Tipo:

- número

Colorspace

Os espaços de cores disponíveis que podem ser especificados.

Tipo:

- "RGB" | "YUV_REC601" | "YUV_REC709"

connectionCallbacks

Os retornos de chamada que estão disponíveis para serem chamados no caso de um erro de conexão.

Tipo:

- Objeto

Propriedades:

Nome	Tipo	Descrição
`disconnect`	disconnectCallback	A função de retorno de chamada a ser chamada quando a conexão terminar.

Nome	Tipo	Descrição
`displayLayout`	displayLayoutCallback	A função de retorno de chamada a ser chamada quando o layout ou a resolução da tela são alterados.
`displayAvailability`	displayAvailabilityCallback	A função de retorno de chamada a ser chamada quando a disponibilidade de uma tela mudar.
`firstFrame`	firstFrameCallback	A função de retorno de chamada a ser chamada quando o primeiro quadro for recebido do servidor do NICE DCV.
`filePrinted`	filePrintedCallback	A função de retorno de chamada a ser chamada quando um arquivo for impresso no servidor do NICE DCV.
`fileDownload`	fileDownloadCallback	A função de retorno de chamada a ser chamada quando um arquivo baixado no servidor do NICE DCV.
`dataChannel`	dataChannelCallback	A função de retorno de chamada a ser chamada quando o servidor do NICE DCV enviar uma notificação sobre a disponibilidade de um canal de dados.

Nome	Tipo	Descrição
`licenseNotification`	licenseNotificationCallback	A função de retorno de chamada a ser chamada quando o servidor do NICE DCV enviar uma notificação sobre o estado da licença.
`idleWarningNotification`	idleWarningNotificationCallback	A função de retorno de chamada a ser chamada quando o servidor do NICE DCV envia um aviso de tempo limite de inatividade.
`collaboratorList`	collaboratorListCallback	A função de retorno de chamada a ser chamada quando o servidor do NICE DCV envia a lista de colaboradores (desde o SDK do Cliente Web do NICE DCV versão 1.1.0).
`qualityIndicatorState`	qualityIndicatorStateCallback	A função de retorno de chamada a ser chamada quando o indicador de qualidade da conexão mudar de estado.
`filestorageEnabled`	filestorageEnabledCallback	A função de retorno de chamada a ser chamada quando o armazenamento de arquivos estiver ativado ou desativado.
`featuresUpdate`	featuresUpdateCallback	A função de retorno de chamada a ser chamada quando o status de um atributo for alterado.

Nome	Tipo	Descrição
`clipboardEvent`	clipboardEventCallback	A função de retorno de chamada a ser chamada quando a `clipboardEvent` é gerada.
`deviceChangeEvent`	deviceChangeEventCallback	A função de retorno de chamada a ser chamada quando um evento `deviceChange` for acionado.
`screenshot`	screenshotCallback	A função de retorno de chamada a ser chamada quando a `screenshot` for gerada.

ConnectionConfig

A configuração para uma conexão do NICE DCV.

Tipo:

- Objeto

Propriedades:

Nome	Tipo	Descrição
`url`	string	O nome do host e a porta do servidor do NICE DCV em execução no seguinte formato: `https://dcv_host_address:port`. Por exemplo: `https://my-dcv-server:8443`.

Nome	Tipo	Descrição
`sessionId`	string	O ID da sessão do NICE DCV.
`authToken`	string	O token de autenticação a ser usado ao se conectar ao servidor.
`baseUrl`	string	O URL absoluto ou relativo do qual carregar arquivos do SDK.
`resourceBaseUrl`	string	O URL absoluto ou relativo a partir do qual acessar os recursos do DCV.
`enabledChannels`	Array.<Channel>	Indica a lista de canais que podem ser ativados. Se não for especificado ou se uma matriz vazia for fornecida, o padrão será todos os canais disponíveis.
`losslessColorspace`	Colorspace	Indica o espaço de cores que será usado. Se não for especificado, o padrão será “RGB”.
`divId`	string	O ID do objeto `div` no HTML DOM em que o SDK deve criar a tela com o fluxo remoto.
`volumeLevel`	integer	O nível de volume preferido. O intervalo válido é de 0 a 100.

Nome	Tipo	Descrição
`clipboardAutoSync`	boolean	Indica se a cópia automátic a da área de transferência da sessão do NICE DCV para a área de transferência do cliente local está ativada para navegadores da Web compatíveis.
`dynamicAudioTuning`	boolean	Indica se o áudio deve ser ajustado dinamicamente com base nas configurações de áudio do servidor do NICE DCV quando uma conexão é estabelecida.
`clientHiDpiScaling`	boolean	Indica se a tela deve ser escalada com base no DPI do cliente.
`highColorAccuracy`	boolean	Indica se deve ser usada a alta precisão de cores, se disponível. O padrão é `false`, se não for especificado.
`enableWebCodecs`	Booliano	Indica se WebCodecs deve ser usado, se disponível. O padrão é `false`, se não for especificado.
`observers`	connectionCallbacks	O retorno de chamada funciona para chamar eventos relacionados à conexão.

Nome	Tipo	Descrição
`callbacks`	connectionCallbacks	O mesmo que a proprieda de `observers` , mas cada retorno de chamada inclui o objeto Conexão como primeiro parâmetro.

ConnectionErrorCode

As enumerações do código ConnectionError disponíveis no módulo DCV

- `ALREADY_OPEN`
- `INVALID_CONFIG`
- `INITIALIZATION_FAILED`
- `REJECTED`
- `MAIN_CHANNEL_ALREADY_OPEN`
- `GENERIC_ERROR` (desde o DCV Server 2021.0)
- `INTERNAL_SERVER_ERROR` (desde o DCV Server 2021.0)
- `AUTHENTICATION_FAILED` (desde o DCV Server 2021.0)
- `PROTOCOL_ERROR` (desde o DCV Server 2021.0)
- `INVALID_SESSION_ID` (desde o DCV Server 2021.0)
- `INVALID_CONNECTION_ID` (desde o DCV Server 2021.0)
- `CONNECTION_LIMIT_REACHED` (desde o DCV Server 2021.0)
- `SERVER_UNREACHABLE` (desde o DCV Server 2022.1)

Tipo:

- número

createDirectory(path)

Parâmetros:

Nome	Tipo	Descrição
`path`	string	O caminho absoluto no servidor no qual queremos criar um diretório. Também deve incluir o nome do diretório de destino.

CustomChannelErrorCode

As enumerações do código CustomChannelError disponíveis no módulo DCV

- `TRANSPORT_ERROR`

Tipo:

- número

dataChannelCallback(info)

A função de retorno de chamada a ser chamada quando o servidor do NICE DCV enviar uma notificação sobre a disponibilidade de um canal de dados.

Parâmetros:

Nome	Tipo	Descrição
`info`	Objeto	Informações sobre o canal de dados.

Nome	Tipo	Descrição
		Nome: `name` (string) — O nome do canal de dados. Nome: `token` (string) — O token de autenticação do canal de dados.

deleteFile(path)

Parâmetros:

Nome	Tipo	Descrição
`path`	string	O caminho absoluto no servidor que identifica o arquivo que queremos excluir.

deviceChangeEventCallback()

A função de retorno de chamada a ser chamada quando um evento `deviceChange` for acionado.

disconnectCallback(reason)

A função de retorno de chamada a ser chamada quando a conexão terminar.

Parâmetros:

Nome	Tipo	Descrição
`reason`	Objeto	O motivo da desconexão. Nome / Tipo / Descrição `code` / número / O código do motivo. `message` / string / A mensagem do motivo.

displayAvailabilityCallback(status, displayId)

A função de retorno de chamada a ser chamada quando a disponibilidade de uma tela mudar.

Parâmetros:

Nome	Tipo	Descrição
`status`	Objeto	O status do monitor. Nome / Tipo / Descrição `enablec` / boolean / Indica se o monitor

<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td></td><td></td><td>está ativado.</td></tr>
<tr><td><code>closed</code></td><td>boolean</td><td>Indica se o monitor está fechado.</td></tr>
</table>
</td></tr>
<tr><td><code>displayId</code></td><td>número</td><td>O identificador do monitor.</td></tr>
</table>

DisplayConfigErrorCode

As enumerações do código DisplayConfigError disponíveis no módulo DCV

- `INVALID_ARGUMENT`
- `UNSUPPORTED_OPERATION`
- `NO_CHANNEL`

Tipo:

- número

displayLayoutCallback(serverWidth, serverHeight, heads)

A função de retorno de chamada a ser chamada quando o layout ou a resolução da tela são alterados.

Parâmetros:

Nome	Tipo	Descrição
`serverWidth`	número	A largura (em pixels) do monitor principal.
`serverHeight`	número	A altura (em pixels) do monitor principal.
`heads`	Array.<Monitor>	Os títulos da exibição compatíveis com o servidor do NICE DCV.

feature

Os valores do atributo.

- `display` - Indica a disponibilidade de um streaming de vídeo com um único monitor.
- `display-multi` - Indica a disponibilidade de um streaming de vídeo com vários monitores.
- `high-color-accuracy` - Indica a disponibilidade de alta precisão de cores (desde o SDK do Cliente da Web do NICE DCV versão 1.1.0).
- `mouse` - Indica a disponibilidade da funcionalidade do mouse.
- `keyboard` - Indica a disponibilidade da funcionalidade do teclado.
- `keyboard-sas` - Indica a disponibilidade da funcionalidade de sequência SAS (Control + Alt + Delete).
- `relative-mouse` - Indica a disponibilidade do modo relativo do mouse.
- `clipboard-copy` - Indica a disponibilidade da funcionalidade de cópia da área de transferência do servidor do NICE DCV para o cliente.
- `clipboard-paste` - Indica a disponibilidade da funcionalidade de colagem da área de transferência do servidor do NICE DCV para o cliente.
- `audio-in` - Indica a disponibilidade da funcionalidade de entrada de áudio usando o microfone.
- `audio-out` - Indica a disponibilidade da funcionalidade de reprodução de áudio.
- `webcam` - Indica a disponibilidade da funcionalidade de streaming da webcam.
- `file-download` - Indica a disponibilidade da funcionalidade de download do arquivo do servidor do NICE DCV para o cliente.

- `file-upload` - Indica a disponibilidade da funcionalidade de upload do arquivo do servidor do NICE DCV para o cliente.
- `timezone-redirection` - Indica a disponibilidade da funcionalidade de redirecionamento de fuso horário (desde o SDK do Cliente da Web do NICE DCV versão 1.3.0).

Tipo:

- string

featuresUpdateCallback(featuresList)

A função de retorno de chamada a ser chamada quando o status de um atributo for alterado.

Parâmetros:

Nome	Tipo	Descrição
`featuresList`	Matriz.<feature>	Uma série de atributos que foram alterados.

fileDownloadCallback(fileResource)

A função de retorno de chamada a ser chamada quando um arquivo baixado no servidor do NICE DCV.

Parâmetros:

Nome	Tipo	Descrição
`fileResource`	Objeto	Informações sobre o arquivo que está pronto para ser baixado. Nome / Tipo / Descrição `id` / string / O identific

Nome	Tipo	Descrição
		Nome / Tipo / Descrição ador do arquivo. `url` / string / O URL a ser usado para baixar o arquivo. `domain` / string / O domínio do recurso. `token` / string / O token de autenticação a ser usado para baixar o arquivo. O token também está incluído

Nome	Tipo	Descrição
		Nome: Tipo: string — Descrição: no URL.

filePrintedCallback(printResource)

A função de retorno de chamada a ser chamada quando um arquivo for impresso no servidor do NICE DCV.

Parâmetros:

Nome	Tipo	Descrição
`printResource`	Objeto	Informações sobre o arquivo impresso. Nome: `id` Tipo: string — Descrição: O identificador do arquivo impresso. Nome: `url` Tipo: string — Descrição: O URL a ser usado para baixar o arquivo impresso.

<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td><code>domain</code></td><td>string</td><td>O domínio do recurso. Nesse caso, <code>printer</code>.</td></tr>
<tr><td><code>token</code></td><td>string</td><td>O token de autenticação a ser usado para baixar o arquivo impresso. O token também está incluído no URL.</td></tr>
</table>
</td></tr>
</table>

filestorage

Objeto que permite explorar e executar ações no sistema de arquivos.

Tipo:

- Objeto

Propriedades:

Nome	Tipo	Descrição
`list`	list	Função que permite listar os itens (arquivos e diretórios) presentes no caminho fornecido no servidor.
`createDirectory`	createDirectory	Função que permite criar um diretório no caminho especificado no servidor.
`retrieveFile`	retrieveFile	Função que permite baixar localmente um arquivo no caminho especificado no servidor.
`deleteFile`	deleteFile	Função que permite excluir um arquivo no caminho especificado no servidor.
`renameFile`	renameFile	Função que permite renomear um arquivo do caminho de origem especificado para o caminho de destino especificado.
`renameDirectory`	renameDirectory	Função que permite renomear um diretório do caminho de origem especificado para o caminho de destino especificado.

Nome	Tipo	Descrição
`storeFile`	storeFile	Função que permite fazer o upload de um arquivo local para o caminho fornecido no servidor.

filestorageEnabledCallback(enabled)

A função de retorno de chamada a ser chamada quando o armazenamento de arquivos estiver ativado. Canal lento somente no Internet Explorer 11.

Parâmetros:

Nome	Tipo	Descrição
`enabled`	boolean	Indica se o armazenamento de arquivos está habilitado.

FileStorageErrorCode

As enumerações do código FileStorageError disponíveis no módulo DCV

- `CANCELLED`
- `ABORTED`
- `INVALID_ARGUMENT`
- `NOT_IMPLEMENTED`
- `ERROR`
- `ALREADY_EXIST`
- `NOT_FOUND`

Tipo:

- número

firstFrameCallback(resizeEnabled, relativeMouseModeEnabled, displayId)

A função de retorno de chamada a ser chamada quando o primeiro quadro for recebido do servidor do NICE DCV. Emitido para cada monitor.

Parâmetros:

Nome	Tipo	Descrição
`resizeEnabled`	boolean	Indica se o servidor é compatível com o redimensionamento do layout de exibição do cliente.
`relativeMouseModeEnabled`	boolean	Indica se o servidor é compatível com o modo relativo do mouse.
`displayId`	número	O identificador do monitor.

idleWarningNotificationCallback(disconnectionDateTime)

A função de retorno de chamada a ser chamada quando o servidor do NICE DCV envia um aviso de tempo limite de inatividade.

Parâmetros:

Nome	Tipo	Descrição
`disconnectionDateTime`	Data	A data e a hora da desconexão.

collaboratorListCallback(collaborators)

A função de retorno de chamada a ser chamada quando o servidor do NICE DCV envia a lista de colaboradores.

Parâmetros:

<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td><code>collaborators</code></td><td>Array.<Object></td><td>Uma lista de objetos contendo informações sobre colaboradores.
<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td><code>usernam</code></td><td>string</td><td>O nome de usuário do colaborador.</td></tr>
<tr><td><code>owner</code></td><td>boolean</td><td>Indica se o colaborador é o proprietário da sessão.</td></tr>
<tr><td><code>connectnId</code></td><td>número</td><td>Indica o ID atribuído pelo servidor à conexão.</td></tr>
</table>
</td></tr>
</table>

licenseNotificationCallback(notification)

A função de retorno de chamada a ser chamada quando o servidor do NICE DCV enviar uma notificação sobre o estado da licença.

Parâmetros:

Nome	Tipo	Descrição
`notification`	Objeto	A notificação.

Nome	Tipo	Descrição
`product`	string	O produto do DCV.
`status`	string	O status da licença.
`message`	string	Uma mensagem.
`leftDay`	número	O número de dias antes que a licença expire.
`isDemo`	boolean	Indica se a licença é uma licença

Nome	Tipo	Descrição
		Nome / Tipo / Descrição de demonstração. `numUnlicensed` / número / O número de conexões não licenciadas. `licensingMode` / string / O modelo de licença. `documentationUrl` / string / O URL da documentação.

list(path)

Parâmetros:

Nome	Tipo	Descrição
`path`	string	O caminho absoluto no servidor do qual queremos listar o conteúdo.

LogLevel

Os níveis de log do SDK disponíveis.

Tipo:

- TRACE | DEBUG | INFO | WARN | ERROR | SILENT

Monitor

Tipo:

- Objeto

Propriedades:

<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td>name</td><td>string</td><td>O nome do título da exibição.</td></tr>
<tr><td>rect</td><td>Objeto</td><td>Informações sobre o título da exibição.

<table><tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr><tr><td>x</td><td>número</td><td>A coordenada inicial x do título da exibição.</td></tr><tr><td>y</td><td>número</td><td>A coordenada inicial</td></tr></table></td></tr>
</table>

<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td></td><td></td><td>y do título da exibição.</td></tr>
<tr><td><code>width</code></td><td>número</td><td>A largura (em pixels) do título da exibição.</td></tr>
<tr><td><code>height</code></td><td>número</td><td>A altura (em pixels) do título da exibição.</td></tr>
</table>
</td></tr>
<tr><td><code>primary</code></td><td>boolean</td><td>Indica se o título da exibição é o título da exibição principal . Isso é determinado pelo sistema operacional remoto, se disponível.</td></tr>
<tr><td><code>dpi</code></td><td>número</td><td>O DPI do título da exibição.</td></tr>
</table>

MultiMonitorErrorCode

As enumerações do código MultiMonitorError disponíveis no módulo DCV

- `NO_DISPLAY_CHANNEL`
- `MAX_DISPLAY_NUMBER_REACHED`
- `INVALID_ARGUMENT`
- `DISPLAY_NOT_OPENED_BY_SERVER`
- `REQUEST_TIMEOUT`
- `GENERIC_ERROR`
- `NO_ERROR`

Tipo:

- número

qualityIndicatorStateCallback(state)

A função de retorno de chamada a ser chamada quando o indicador de qualidade da conexão mudar de estado.

Parâmetros:

Nome	Tipo	Descrição
`state`	Array.<Object>	Informações sobre a qualidade da conexão. Nome / Tipo / Descrição `name` / string / O nome do indicador. `status` / NORMAL | WARNIN / Uma descrição do status.

Nome	Tipo	Descrição

Nome	Tipo	Descrição
	| CRITICA	
`changec`	boolean	Indica se o status foi alterado.

renameDirectory(src, dest)

Parâmetros:

Nome	Tipo	Descrição
`src`	string	O caminho de origem absoluto no servidor que identifica o diretório que queremos renomear.
`dest`	string	O caminho de destino absoluto no servidor que especifica o caminho de destino e o nome do diretório.

renameFile(src, dest)

Parâmetros:

Nome	Tipo	Descrição
`src`	string	O caminho de origem absoluto no servidor que identifica

Nome	Tipo	Descrição
		o arquivo que queremos renomear.
`dest`	string	O caminho de destino absoluto no servidor que especifica o caminho de destino e o nome do arquivo.

ResolutionErrorCode

As enumerações do código ResolutionError disponíveis no módulo DCV

- `INVALID_ARGUMENT`
- `NO_CHANNEL`
- `NOT_IMPLEMENTED`

Tipo:

- número

retrieveFile(path)

Parâmetros:

Nome	Tipo	Descrição
`path`	string	O caminho absoluto no servidor que identifica o arquivo que queremos baixar localmente.

screenshotCallback(screenshot)

A função de retorno de chamada a ser chamada quando a uma captura de tela estiver disponível.

Parâmetros:

Nome	Tipo	Descrição
`screenshot`	byte[]	Buffer da captura de tela no formato PNG ou `null` se a recuperação da captura de tela falhar.

ScreenshotErrorCode

As enumerações do código ScreenshotError disponíveis no módulo DCV

- `NO_CHANNEL`
- `GENERIC_ERROR`

Tipo:

- número

serverInfo

Tipo:

- Objeto

Propriedades:

Nome	Tipo	Descrição
`name`	string	O nome do software.
`version`	Objeto	O número da versão do software.

<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td><code>major</code></td><td>número</td><td>O número da versão principal.</td></tr>
<tr><td><code>minor</code></td><td>número</td><td>O número da versão secundária.</td></tr>
<tr><td><code>revisic</code></td><td>número</td><td>O número da versão da revisão.</td></tr>
</table>
</td></tr>
<tr><td><code>os</code></td><td>string</td><td>O sistema operacional.</td></tr>
<tr><td><code>arch</code></td><td>string</td><td>A arquitetura.</td></tr>
<tr><td><code>hostname</code></td><td>string</td><td>O nome do host.</td></tr>
</table>

stats

Tipo:

- Objeto

Propriedades:

Nome	Tipo	Descrição
`fps`	número	Os quadros atuais por segundo.
`traffic`	número	O tráfego atual em bit/s.
`peakTraffic`	número	O pico de tráfego em bit/s desde que a conexão foi estabelecida.
`latency`	número	A latência atual em ms.
`currentChannels`	número	O número de canais que foram abertos desde que a conexão foi estabelecida.
`openedChannels`	número	O número de canais abertos atualmente.
`channelErrors`	número	O número de canais que relataram um erro.

storeFile(file, dir)

Parâmetros:

Nome	Tipo	Descrição
`file`	Arquivo	O objeto de arquivo (para obter mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/API/File) que queremos carregar no servidor.

Nome	Tipo	Descrição
`dir`	string	O caminho absoluto no servidor no qual queremos fazer o upload do arquivo.

TimezoneRedirectionErrorCode

As enumerações do código TimezoneRedirectionError disponíveis no módulo DCV

- `INVALID_ARGUMENT`
- `NO_CHANNEL`
- `USER_CANNOT_CHANGE`

Tipo:

- número

TimezoneRedirectionSettingCode

As enumerações do código TimezoneRedirectionSetting disponíveis no módulo DCV

- `ALWAYS_OFF`
- `ALWAYS_ON`
- `CLIENT_DECIDES`

Tipo:

- número

TimezoneRedirectionStatusCode

As enumerações do código TimezoneRedirectionStatus disponíveis no módulo DCV

- `SUCCESS`
- `PERMISSION_ERROR`
- `GENERIC_ERROR`

Tipo:

- número

WebcamErrorCode

As enumerações do código WebcamError disponíveis no módulo DCV

- `SETTING_WEBCAM_FAILED`
- `CHANNEL_NOT_AVAILABLE`

Tipo:

- número

Classe de conexão

A Classe de conexão obtida chamando o método `connect` do modulo `dcv`. Para ver um exemplo de como usá-lo, consulte a seção Introdução.

Expõe

- Métodos

Métodos

Listar

- attachDisplay(win, displayConf) → {Promessa.<number>|Promessa.<{code: MultiMonitorErrorCode, message: string}>}
- captureClipboardEvents(enabled, win, displayId) → {void}
- detachDisplay(displayId) → {void}
- disconnect() → {void}
- disconnectCollaborator(connectionId) → {void}
- enableDisplayQualityUpdates(enable) → {void}
- enableTimezoneRedirection(enable) → {Promessa|Promessa.<{code: TimezoneRedirectionErrorCode, message: string}>}

- enterRelativeMouseMode() → {void}
- getConnectedDevices() → {Promise.<Array.<MediaDeviceInfo>>|Promise.<{message: string}>}
- getFileExplorer() → {Promessa.<filestorage>|Promessa.<{code: ChannelErrorCode, message: string}>}
- getServerInfo() → {serverInfo}
- getScreenshot() → {Promessa|Promessa.<{code: ScreenshotErrorCode, message: string}>}
- getStats() → {stats}
- latchModifierKey(key, location, isDown) → {boolean}
- openChannel(name, authToken, callbacks, namespace) → {Promessa|Promessa.<{code: ChannelErrorCode, message: string}>}
- queryFeature(featureName) → {Promise.<{enabled: boolean, remote?: string, autoCopy?: boolean, autoPaste?: boolean, serviceStatus?: string, available?: boolean}>|Promise.<{message: string}>}
- registerKeyboardShortcuts(shortcuts) → {void}
- requestDisplayConfig(highColorAccuracy) → {Promessa|Promessa.<{code: DisplayConfigErrorCode, message: string}>}
- requestDisplayLayout(layout) → {Promessa|Promessa.<{code: ResolutionErrorCode, message: string}>}
- requestResolution(width, height) → {Promessa|Promessa.<{code: ResolutionErrorCode, message: string}>}
- sendKeyboardEvent(event) → {boolean}
- sendKeyboardShortcut(shortcut) → {void}
- setDisplayQuality(min, maxopt) → {void}
- setDisplayScale(scaleRatio, displayId) → {Promessa|Promessa.<{code: ResolutionErrorCode, message: string}>} (DEPRECATED)
- setKeyboardQuirks(quirks) → {void}
- setMaxDisplayResolution(maxWidth, maxHeight) → {void}
- setMicrophone(enable) → {Promessa|Promessa.<{code: AudioErrorCode, message: string}>}
- setMinDisplayResolution(minWidth, minHeight) → {void}
- setUploadBandwidth(value) → {number}
- setVolume(volume) → {void}

- setWebcam(enable, deviceId) → {Promessa|Promessa.<{code: WebcamErrorCode, message: string}>}
- syncClipboards() → {boolean}

attachDisplay(win, displayConf) → {Promessa.<number>|Promessa.<{code: MultiMonitorErrorCode, message: string}>}

Anexa uma tela específica em uma janela. Você não pode anexar a tela principal. Se bem-sucedida, a função retornará `displayId`.

Parâmetros:

Nome	Tipo	Descrição
`win`	Objeto	A janela à qual a exibição deve ser anexada.
`displayConf`	Objeto	A configuração da exibição. Nome: `displ` — Tipo: númer — Atribut: <opcic> — Descrição: O ID da exibição. Nome: `displ vName` — Tipo: — Atribut: — Descrição: O nome do título da div. da exibição.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa.<number> | Promessa.<{code: MultiMonitorErrorCode, message: string}>

captureClipboardEvents(enabled, win, displayId) → {void}

Inicia ou para de ouvir eventos de copiar e colar. No caso de áreas de transferências interativas (sempre no caso de colar), precisamos começar a ouvir os eventos de copiar/colar. Pode ser útil iniciar e parar de ouvir somente quando necessário, por exemplo, quando um modal é exibido.

Parâmetros:

Nome	Tipo	Atributos	Descrição
`enabled`	boolean		Para começar a ouvir eventos, especifiq ue `true`. Para parar de ouvir eventos, especifique `false`.
`win`	Objeto	<opcional>	A janela na qual se deve ouvir os eventos. Se omitido, a janela padrão será usada.
`displayId`	número	<opcional>	O ID do monitor que deve ouvir os eventos. Se omitido, o monitor padrão da janela será usada.

Retorna:

Tipo

nulo

detachDisplay(displayId) → {void}

Desconecta um monitor específico. O monitor principal não pode ser desconectado.

Parâmetros:

Nome	Tipo	Descrição
`displayId`	número	O ID da tela a ser desconectado.

Retorna:

Tipo

nulo

disconnect() → {void}

Desconecta-se do servidor do NICE DCV e fecha a conexão.

Retorna:

Tipo

nulo

disconnectCollaborator(connectionId) → {void}

Solicita a desconexão do colaborador conectado com o ID de conexão fornecido (desde o SDK do Cliente Web do NICE DCV versão 1.1.0).

Parâmetros:

Nome	Tipo	Descrição
`connectionId`	boolean	O ID da conexão que será desconectada.

Retorna:

Tipo

nulo

enableDisplayQualityUpdates(enable) → {void}

Ativa ou desativa as atualizações de qualidade de exibição para áreas de streaming que não recebem atualizações. Desativar as atualizações da qualidade de exibição reduz o uso da largura de banda, mas também diminui a qualidade da tela.

Parâmetros:

Nome	Tipo	Descrição
`enable`	boolean	Para ativar as atualizações da qualidade de exibição, especifique `true`. Para desativar as atualizações da qualidade de exibição, especifique `false`.

Retorna:

Tipo

nulo

enableTimezoneRedirection(enable) → {Promessa|Promessa.<{code: TimezoneRedirectionErrorCode, message: string}>}

Ativa ou desativa o redirecionamento de fuso horário. Depois de ativado, o cliente solicita que o servidor faça com que o fuso horário da área de trabalho do servidor corresponda ao fuso horário do cliente.

Parâmetros:

Nome	Tipo	Descrição
`enable`	boolean	Para ativar o redirecionamento de fuso horário, especifiq ue `true`. Para desativar o redirecionamento de fuso horário, especifique `false`.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa.<number> | Promessa.<{code: TimezoneRedirectionErrorCode, message: string}>

enterRelativeMouseMode() → {void}

Ativa o modo relativo do mouse.

Retorna:

Tipo

nulo

getConnectedDevices() → {Promise.<Array.<MediaDeviceInfo>>|Promise.<{message: string}>}

Solicita uma lista dos dispositivos de mídia conectados ao computador cliente.

Retorna:

Se for bem-sucedido, ele retornará uma Promessa que é resolvida em uma matriz de objetos MediaDeviceInfo. Para obter mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/API/MediaDeviceInfo. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promise.<Array.<MediaDeviceInfo>> | Promise.<{message: string}>

getFileExplorer() → {Promessa.<filestorage>|Promessa.<{code: ChannelErrorCode, message: string}>}

Obtém um objeto para gerenciar o armazenamento de arquivos do servidor do NICE DCV.

Retorna:

Promise. Resolve para o objeto explorador de arquivos, se preenchido, ou para um objeto de erro, se rejeitado.

Tipo

Promessa.<filestorage> | Promessa.<{code: ChannelErrorCode, message: string}>

getServerInfo() → {serverInfo}

Obtém informações sobre o servidor do NICE DCV.

Retorna:

Informações sobre o software do servidor.

Tipo

serverInfo

getScreenshot() → {Promessa|Promessa.<{code: ScreenshotErrorCode, message: string}>}

Recupera a captura de tela da área de trabalho remota no formato PNG. A captura de tela será retornada no observador screenshotCallback. Em vez disso, `null` será retornado em caso de falhas.

Retorna:

Promessa que será resolvida se a solicitação for processada. Se rejeitado, recebemos um objeto de erro.

Tipo

Promessa | Promessa.<{code: ScreenshotErrorCode, message: string}>

getStats() → {stats}

Obtém estatísticas sobre o servidor do NICE DCV.

Retorna:

Informações sobre as estatísticas de streaming.

Tipo

stats

latchModifierKey(key, location, isDown) → {boolean}

Envia um único evento `keydown` ou `keyup` do teclado para um modificador permitido.

Parâmetros:

Nome	Tipo	Descrição
`key`	Control | Alt | AltGraph | Meta | OS | Shift	A tecla para enviar.
`location`	KeyboardEvent.location	O local da tecla. Para obter mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/location.
`isDown`	boolean	Se o evento chave a ser injetado for um keydown (`true`) ou um keyup (`false`).

Retorna:

Se a combinação solicitada for válida, a função retornará `true`, caso contrário, retornará `false`.

Tipo

boolean

openChannel(name, authToken, callbacks, namespace) → {Promessa| Promessa.<{code: ChannelErrorCode, message: string}>}

Abre um canal de dados personalizado na conexão se ela tiver sido criada no Servidor do NICE DCV.

Parâmetros:

Nome	Tipo	Descrição
`name`	string	O nome do canal.
`authToken`	string	O token de autenticação a ser usado ao se conectar ao servidor.
`callbacks`	Objeto	Os retornos de chamada onMessage e onClose funcionam para chamar.
`namespace`	string	O namespace do modelo. Disponível desde o SDK do Cliente da Web do NICE DCV 1.2.0 e o Servidor do NICE DCV 2022.1.

Retorna:

Promise. Se rejeitado, recebemos um objeto de erro.

Tipo

Promessa | Promessa.<{code: ChannelErrorCode, message: string}>

queryFeature(featureName) → {Promise.<{enabled: boolean, remote?: string, autoCopy?: boolean, autoPaste?: boolean, serviceStatus?: string, available?: boolean}>|Promise.<{message: string}>}

Consulta o status de um atributo específico do servidor do NICE DCV.

Parâmetros:

Nome	Tipo	Descrição
`featureName`	feature	O nome do atributo a ser consultado.

Retorna:

Promise. Se resolvida, a função retorna um objeto `status` que sempre contém uma propriedade `enabled` e, possivelmente, também outras propriedades. Se rejeitada, a função retornará um objeto `error`.

Tipo

{Promise.<{enabled: boolean, remote?: string, autoCopy?: boolean, autoPaste?: boolean, serviceStatus?: string, available?: boolean}> | Promise.<{message: string}>

registerKeyboardShortcuts(shortcuts) → {void}

Registra os atalhos de teclado.

Parâmetros:

Nome	Tipo	Descrição
`shortcuts`	Array.<Object>	A matriz de chaves e mapeamentos a serem registrados.

<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td>sequenc</td><td>Array.<Object></td><td>O atalho do teclado para se registrar.
<table>
<tr><th>N</th><th>T</th><th>Descrição</th></tr>
<tr><td>k
v</td><td>K
v</td><td>O valor da tecla pressionada pelo usuário. Para obter mais informações, consulte https://developer.mozilla.org/en-US/</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

Nome	Tipo	Descrição

Nome	Tipo	Descrição

N	T	D
		Descrição docs/Web/API/KeyboardEvent/key.
1	K v ti	Abcon matriz de chaves a serem enviadas. A localização da tecla no teclado. Para obter mais informações, consulte https://developer.mozilla.org/

<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Nome</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>N</th><th>T</th><th>Descrição</th></tr>
<tr><td></td><td></td><td>en-US/docs/Web/API/KeyboardEvent/location.</td></tr>
</table>
</td></tr>
<tr><td><code>output</code></td><td>Array.<Object></td><td>A ação que deve ser executada pelo atalho.
<table>
<tr><th>N</th><th>T</th><th>Descrição</th></tr>
<tr><td>k</td><td>k
v</td><td>KeyboardEvent.key value da tecla pressionada pelo usuário. Para obter mais informaçõ</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

Nome	Tipo	Descrição

Nome	Tipo	Descrição

N	T	Descrição
		es, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key.
1	K v ti	AbcardEtatriz de chaves a serem enviadas. A localização da tecla no teclado. Para obter

Nome	Tipo	Descrição
		Nome Tipo Descrição N T Descrição mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/location.

Retorna:

Tipo

nulo

requestDisplayConfig(highColorAccuracy) → {Promessa|Promessa.<{code: DisplayConfigErrorCode, message: string}>}

Solicita uma configuração de exibição atualizada do Servidor do NICE DCV. Disponível desde o SDK do Cliente da Web do NICE DCV 1.1.0 e o Servidor do NICE DCV 2022.0.

Parâmetros:

Nome	Tipo	Descrição
`highColorAccuracy`	boolean	Se a alta precisão de cores deve ou não ser solicitada.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa | Promessa.<{code: DisplayConfigErrorCode, message: string}>

requestDisplayLayout(layout) → {Promessa|Promessa.<{code: ResolutionErrorCode, message: string}>}

Solicita um layout de exibição atualizado para a conexão.

Parâmetros:

Nome	Tipo	Descrição
`layout`	Array.<Monitor>	O solicitado é exibido no layout.

Retorna:

Promise. Se rejeitado, recebemos um objeto de erro.

Tipo

Promessa | Promessa.<{code: ResolutionErrorCode, message: string}>

requestResolution(width, height) → {Promessa|Promessa.<{code: ResolutionErrorCode, message: string}>}

Solicita uma resolução de exibição atualizada do servidor do NICE DCV.

Parâmetros:

Nome	Tipo	Descrição
`width`	número	A largura a ser solicitada em pixels. O valor mínimo permitido é `0`.
`height`	número	A altura a ser solicitada em pixels. O valor mínimo permitido é `0`.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa | Promessa.<{code: ResolutionErrorCode, message: string}>

sendKeyboardEvent(event) → {boolean}

Envia um evento de atalho de teclado. Para obter mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent. Os eventos de teclado válidos incluem: `keydown`, `keypress` e `keyup`. Para obter mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent#events.

Parâmetros:

Nome	Tipo	Descrição
`event`	KeyboardEvent	O evento de teclado a ser enviado.

Retorna:

Se o evento não for válido, a função retornará `false`. Se o evento for válido, a função retornará `true`.

Tipo

boolean

sendKeyboardShortcut(shortcut) → {void}

Envia um atalho de teclado. Use essa função para enviar uma sequência completa de `keydown` ou `keyup`. Por exemplo, enviar Ctrl + Alt + Del envia os eventos `keydown` para todas as teclas seguidas pelos eventos `keyup`. Use essa função mesmo se quiser enviar uma única chave.

Parâmetros:

Nome	Tipo	Descrição
`shortcut`	Array.<Object>	A matriz de chaves a serem enviadas.

Nome	Tipo	Descrição
`key`	Keyboard vent.key	O valor da tecla pressionada pelo usuário. Para obter mais informações, consulte https://

Nome	Tipo	Descrição
		Nome Tipo Descrição d eveloper. mozilla.o rg/en- US/ docs/ Web/ API/ Keybo ardEvent/ key. `locatic` Keyboarc vent.loca tion A matriz de chaves a serem enviadas. A localizaç ão da tecla no teclado. Para obter mais informaçõ es, consulte https:// d

Nome	Tipo	Descrição
		Nome / Tipo / Descrição eveloper.mozilla.org/en-US/docs/Web/API/KeyboardEvent/location.

Retorna:

Tipo

nulo

setDisplayQuality(min, maxopt) → {void}

Define a qualidade da imagem a ser usada para a conexão. O intervalo válido é `0` a `100`, sendo `1` a menor qualidade de imagem e `100` a mais alta qualidade de imagem. Especifique `0` para reter o valor atual.

Parâmetros:

Nome	Tipo	Atributos	Descrição
`min`	número		A qualidade mínima da imagem.
`max`	número	<opcional>	A qualidade máxima da imagem.

Retorna:

Tipo

nulo

setDisplayScale(scaleRatio, displayId) → {Promessa|Promessa.<{code: ResolutionErrorCode, message: string}>} (DEPRECATED)

Obsoleto desde a versão 1.3.0. Não há mais necessidade de definir a escala de exibição. As coordenadas do mouse serão gerenciadas internamente de forma automática.

Notifica o NICE DCV de que a tela está escalada no lado do cliente. Use isso para notificar o servidor de que ele precisa escalar os eventos do mouse para corresponder à proporção de exibição do cliente.

Parâmetros:

Nome	Tipo	Descrição
`scaleRatio`	flutuante	A taxa de escala a ser usada. Deve ser um número estritamente positivo.
`displayId`	número	O ID da tela a ser dimension ada.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa | Promessa.<{code: ResolutionErrorCode, message: string}>

setKeyboardQuirks(quirks) → {void}

Define peculiaridades do teclado para o computador cliente.

Parâmetros:

Nome	Tipo	Descrição
`quirks`	Objeto	As peculiaridades do teclado a serem ativadas ou desativadas. Nome: `macOptionToAlt` Tipo: boolean Descrição: Para mapear a tecla Option como Alt no macOS, especifique `true`. Caso contrário, especifique `false`. Nome: `macCommandToCtrl` Tipo: boolean Descrição: Para mapear a tecla Command como Ctrl no macOS, especifique `true`.

Nome	Tipo	Descrição
		Nome / Tipo / Descrição Caso contrário, especifique `false`.

Retorna:

Tipo

nulo

setMaxDisplayResolution(maxWidth, maxHeight) → {void}

Define a resolução máxima da tela a ser usada para a conexão.

Parâmetros:

Nome	Tipo	Descrição
`maxWidth`	número	A largura máxima da tela em pixels. O valor mínimo permitido é `0`.
`maxHeight`	número	A altura máxima da tela em pixels. O valor mínimo permitido é `0`.

Retorna:

Tipo

nulo

setMicrophone(enable) → {Promessa|Promessa.<{code: AudioErrorCode, message: string}>}

Ativa ou desativa o microfone.

Parâmetros:

Nome	Tipo	Descrição
`enable`	boolean	Para ativar o microfone, especifique `true`. Para desativar o microfone, especifique `false`.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa | Promessa.<{code: AudioErrorCode, message: string}>

setMinDisplayResolution(minWidth, minHeight) → {void}

Define a resolução mínima da tela a ser usada para a conexão. Alguns aplicativos podem exigir uma resolução mínima da tela. Se a resolução mínima exigida for maior que a resolução máxima suportada pelo cliente, será usada uma estratégia de redimensionamento. Use essa função com cuidado. A estratégia de redimensionamento pode fazer com que o sistema de entrada do mouse e do teclado fique menos preciso.

Parâmetros:

Nome	Tipo	Descrição
`minWidth`	número	A largura mínima da tela em pixels. O valor mínimo permitido é `0`.

Nome	Tipo	Descrição
`minHeight`	número	A largura mínima da tela em pixels. O valor mínimo permitido é `0`.

Retorna:

Tipo

nulo

setUploadBandwidth(value) → {number}

Define a largura de banda máxima a ser usada para carregar arquivos no servidor do NICE DCV.

Parâmetros:

Nome	Tipo	Descrição
`value`	número	O limite máxima da largura de banda em Kbps. O intervalo válido é de 1.024 kbps a 102.400 kbps.

Retorna:

- O limite de largura de banda definido. `null` se o atributo de armazenamento de arquivos estiver desativado no servidor.

Tipo

número

setVolume(volume) → {void}

Define o nível de volume a ser usado para áudio. O intervalo válido é de 0 a 100, sendo 0 o volume mais baixo e 100 sendo o volume mais alto.

Parâmetros:

Nome	Tipo	Descrição
`volume`	número	O nível de volume a ser usado.

Retorna:

Tipo

nulo

setWebcam(enable, deviceId) → {Promessa|Promessa.<{code: WebcamErrorCode, message: string}>}

Ativa ou desativa a webcam.

Parâmetros:

Nome	Tipo	Descrição
`enable`	boolean	Para ativar a webcam, especifique `true`. Para desativar a webcam, especifiq ue `false`.
`deviceId`	string	O ID do dispositivo da webcam.

Retorna:

Prometa que, se for bem-sucedida, será resolvida para o deviceId da webcam anexado/desanexado. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa | Promessa.<{code: WebcamErrorCode, message: string}>

syncClipboards() → {boolean}

Sincroniza a área de transferência do cliente local com a área de transferência remota do servidor do NICE DCV. A cópia automática deve ser compatível com o navegador.

Retorna:

Se as áreas de transferência tiverem sido sincronizadas, a função retornará `true`. Se as áreas de transferência não tiverem sido sincronizadas ou se o navegador não for compatível com a cópia automática, a função retornará `false`.

Tipo

boolean

Classe de autenticação

A classe de autenticação deve ser usada para obter um token de autenticação chamando o `authenticate`método do módulo `dcv`. Para ver um exemplo de como usá-lo, consulte a seção Introdução.

Expõe

- Métodos

Métodos

Listar

- retry() → {void}
- sendCredentials(credentials) → {void}

retry() → {void}

Recupera o processo de autenticação.

Retorna:

Tipo

nulo

sendCredentials(credentials) → {void}

Envia as credenciais de autenticação fornecidas pelo cliente para o servidor do NICE DCV.

Parâmetros:

Nome	Tipo	Descrição
`credentials`	Objeto	O objeto que contém as credenciais fornecidas. As credenciais devem ter o mesmo nome e ser do mesmo tipo especificado no desafio.

Retorna:

Tipo

nulo

Classe de recurso

A classe de recurso pode buscar ou descartar o arquivo correspondente que acabou de ser impresso ou baixado. Ao realizar essas ações, as funções do observador correspondentes `filePrinted` e `fileDownload`, respectivamente, seriam invocadas com o objeto de recurso como único argumento. Esse recurso pode ser aceito ou recusado para buscar ou descartar o arquivo ao qual eles fazem referência.

Expõe

- Métodos

Métodos

Listar

- accept(urlParameters) → {void}
- decline() → {void}

accept(urlParameters) → {void}

Baixa localmente o recurso.

Parâmetros:

Nome	Tipo	Descrição
`urlParameters`	Objeto	O objeto opcional que contém os pares de chave/valor dos parâmetros de pesquisa de URL transmitidos para a solicitação de busca do recurso.

Retorna:

Tipo

nulo

decline() → {void}

Descarta o recurso.

Retorna:

Tipo

nulo

SDK da interface de usuário da Web do NICE DCV

Uma biblioteca de componentes JavaScript React, atualmente exportando um único componente React chamado `DCVViewer` que se conecta ao Servidor do NICE DCV e renderiza a barra de ferramentas para interagir com o fluxo remoto.

Expõe

- Componentes

Componentes

Listar

- DCVViewer

DCVViewer

O componente React que renderiza a barra de ferramentas com todas as suas funcionalidades úteis para interagir com o fluxo remoto.

Propriedades:

Listar

- dcv
- uiConfig

dcv

Nome	Tipo	Obrigatório	Descrição
`dcv`	Objeto	Sim	O objeto que define as propriedades necessárias para estabelecer a conexão com o Servidor do NICE DCV, definindo o nível do log e a URL de onde carregar os ativos do SDK do Cliente Web do NICE DCV e acessar os recursos do DCV.

Nome	Tipo	Obrigatório	Descrição
			Nom Tipo Obri io Descrição ses Strir Sim O ID da sessão do NICE DCV. aut Strir Sim O token de autentica ção a ser usado ao se conectar ao servidor. ser Strir Sim O nome do host e a porta do servidor do

Nome	Tipo	Obrigatório	Descrição
			Nom Tipo Obrigatório Descrição NICE DCV em execução no seguinte formato: https://dcv_host_address:port. Por exemplo: https://my-dcv-server:8443. bas Strir Sim O URL absoluto ou relativo do qual carregar arquivos do SDK.

Nome	Tipo	Obrigatório	Descrição
			Nom Tipo Obri io Descrição res ase Strir Não (pad "") O URL absoluto ou relativo a partir do qual acessar os recursos do DCV. onD ect func Não (pad () => {}) A função de retorno de chamada é invocada ao se desconectar do servidor do NICE DCV

Nome	Tipo	Obrigatório	Descrição
			Nom Tipo Obri io Descrição e a conexão é fechada. `log` Logl Não (pad Log INF O nível de l. log a ser usado no visualiza dor.

uiConfig

Nome	Tipo	Obrigatório	Descrição
`uiConfig`	Objeto	Não (padrão: `{}`)	O objeto que define as propriedades para configurar se a barra de ferramentas está visível e se os botões de tela cheia e de vários monitores devem ser exibidos nela.

Nome	Tipo	Obrigatório	Descrição

Nom	Tipo	Obri io	Descrição
`too`	Obje	Não (pad `{}`)	O objeto que define as opções de configura ção da barra de ferrament as.

Nome	Tipo	Obrigatório	Descrição

Nome	Tipo	Obrigatório	Descrição

Nome	Tipo	Obrigatório	Descrição
			Nom Tipo Obri io Descrição
			ferramen as.
			(opção para definir se deseja mostrar ou ocultar o botão de tela cheia na barra de ferramen as.
			(opção para definir se deseja mostrar

Nome	Tipo	Obrigatório	Descrição
			Nom Tipo Obri io Descrição Obrigatório ou ocultar o botão de vários monitores na barra de ferramentas.

Notas de versão e histórico de documentos do SDK do Cliente da Web do NICE DCV

Esta página traz as notas da versão e o histórico de documentos do SDK do Cliente da Web do NICE DCV.

Tópicos

- Notas de versão do SDK do Cliente da Web do NICE DCV
- Histórico do documento

Notas de versão do SDK do Cliente da Web do NICE DCV

Esta seção fornece notas de versão do SDK do Cliente da Web do NICE DCV por data de versão.

Tópicos

- 1.5.6 — 9 de novembro de 2023
- 1.4.4 — 29 de junho de 2023
- 1.4.0 — 28 de março de 2023
- 1.3.1 — 9 de dezembro de 2022
- 1.3.0 — 11 de novembro de 2022
- 1.2.1 — 21 de julho de 2022
- 1.2.0 — 29 de junho de 2022
- 1.1.3 — 23 de maio de 2022
- 1.1.2 — 19 de maio de 2022
- 1.1.1 — 23 de março de 2022
- 1.1.0 — 23 de fevereiro de 2022
- 1.0.4 — 20 de dezembro de 2021
- 1.0.3 — 1º de setembro de 2021
- 1.0.2 — 30 de julho de 2021
- 1.0.1 — 31 de maio de 2021
- 1.0.0 — 24 de março de 2021

1.5.6 — 9 de novembro de 2023

Versão	Notas de versão
• Versão semântica: 1.5.6 • Compilação: 659	Alterações e correções de bugs • Melhorias de desempenho na decodificação e renderização do streaming • Removido o suporte para Internet Explorer 11

1.4.4 — 29 de junho de 2023

Versão	Notas de versão
• Versão semântica: 1.4.4 • Compilação: 573	Alterações e correções de bugs • O componente de interface do usuário do visualizador agora usa a API `navigator.keyboard.lock` em navegadores compatíveis com ela para lidar com teclas especiais em tela cheia. • Corrigido um problema que poderia apresentar cores erradas no Chrome 114 ou mais recente. • Detecção aprimorada de WebCodecs. • Corrigido um problema com o estado do botão do mouse ao entrar na janela. •

Versão	Notas de versão
	Corrigido um problema que poderia fazer com que as teclas modificadoras permanecessem pressionadas no macOS. • Robustez de áudio aprimorada para condições de rede degradadas. • Corrigido um vazamento de memória. • Logs aprimorados para incluir tempo e nível.

1.4.0 — 28 de março de 2023

Versão	Notas de versão
• Versão semântica: 1.4.0 • Compilação: 476	Novos recursos • Foi adicionado um novo método `uploadFiles` ao objeto `FileStorage` para carregar vários arquivos. • O componente de interface do usuário do visualizador agora é compatível com arrastar e soltar para iniciar o upload do arquivo. • A API do navegador WebCodecs agora é usada também para áudio e webcam. Alterações e correções de bugs •

Versão	Notas de versão
	Corrigidos os vazamentos de memória relacionados a conexões repetidas da mesma página. • `setUploadBandwidth` agora permite valores de até 1 Gbps. • Renderização otimizada dos componentes da interface do usuário. • Corrigida a compatibilidade com cursores animados no Windows. • Corrigido um problema com a compatibi lidade da área de transferência quando dados de texto e imagem estão presentes na mesma operação. • Maior robustez da API da Webcam: as configurações não podem ser alteradas enquanto uma solicitação já está em andamento. `webcam.setEnabled` agora acompanha o ID do dispositivo quando a solicitação está em andamento e retorna uma Promessa. O componente de interface do usuário do visualizador exibe a notificação em caso de erro.

1.3.1 — 9 de dezembro de 2022

Versão	Notas de versão
• Versão semântica: 1.3.1	Alterações e correções de bugs

Versão	Notas de versão
• Compilação: 413	• Corrigido um problema que poderia fazer com que a interface do usuário de redirecio namento de fuso horário perdesse a sincronização com o servidor. • Corrigido um vazamento de memória após várias reconexões. • Corrigido um problema que fazia com que uma página ficasse em branco na desconexã o. • Corrigido um bug que fazia com que os avisos do console no decodificador de áudio fechassem.

1.3.0 — 11 de novembro de 2022

Versão	Notas de versão
• Versão semântica: 1.3.0 • Compilação: 407	Novos recursos • Adotado o Cloudscape (https://cloudscap e.design) para o componente do Visualizador da Interface do Usuário. • Adicionada compatibilidade para redirecio namento de fuso horário. Alterações e correções de bugs •

Versão	Notas de versão
	Corrigida a falta de atualização na área de transferência assíncrona quando o visualizador do DCV está focado. • A função `setDisplayScale` não é mais necessária ao escalar a tela no lado do cliente. • O componente `DCVViewer` agora chama `disconnect()` automaticamente quando é desmontado.

1.2.1 — 21 de julho de 2022

Versão	Notas de versão
• Versão semântica: 1.2.1 • Compilação: 358	Alterações e correções de bugs • Corrigido um problema que resultava em uma falha na conexão com o servidor do NICE DCV 2019.1 e versões anteriores.

1.2.0 — 29 de junho de 2022

Versão	Notas de versão
• Versão semântica: 1.2.0 • Compilação: 352	Alterações e correções de bugs • Corrigido o erro de falha quando os quadros recebidos são maiores do que a resolução máxima suportada (4.096 x 2.160).

Versão	Notas de versão
	• Os objetos de atributo (transmitidos como argumentos para os observadores `fileDownload` e `filePrinted`) agora têm os métodos `accept` e `decline` que podem ser chamados no objeto para baixar e descartar o recurso, respectivamente. • Pequena correção de bug na sincronização automática da área de transferência ao desconectar.

1.1.3 — 23 de maio de 2022

Versão	Notas de versão
• Versão semântica: 1.1.3 • Compilação: 329	Alterações e correções de bugs • Corrigido um problema que impedia a conexão quando a opção web-url-path era especificada.

1.1.2 — 19 de maio de 2022

Versão	Notas de versão
• Versão semântica: 1.1.2 • Compilação: 322	Alterações e correções de bugs • Corrigido um problema que poderia fazer com que a entrada não funcionasse corretamente após a conexão.

Versão	Notas de versão
	• Corrigidas as coordenadas do mouse quando a proporção da escala é maior que 1.

1.1.1 — 23 de março de 2022

Versão	Notas de versão
• Versão semântica: 1.1.1 • Compilação: 309	Alterações e correções de bugs • Relate `Transport Error` quando a comunicação com o servidor atingir o tempo limite. • Corrigido um erro de decodificação recorrent e ao transmitir grandes resoluções.

1.1.0 — 23 de fevereiro de 2022

Versão	Notas de versão
• Versão semântica: 1.1.0 • Compilação: 295	Novos recursos • Inicie a biblioteca do SDK da interface do usuário da Web do NICE DCV com o componente do React`DCVViewer` . • Exporte o SDK do Cliente da Web do NICE DCV como módulos UMD e ES. • Adicionada compatibilidade de alta precisão de cores.

Versão	Notas de versão
	• Adicionada a capacidade de listar e interagir com clientes conectados a uma sessão. Adicionadas notificações para conexão e desconexão. Alterações e correções de bugs • Compatibilidade aprimorada de decodific ação de webcodecs. • Várias melhorias no teclado. • Corrigido um bug que estava impedindo a abertura de uma segunda tela quando a área de transferência estava desativada.

1.0.4 — 20 de dezembro de 2021

Versão	Notas de versão
• Versão semântica: 1.0.4 • Compilação: 249	Novos recursos • Compatibilidade para abrir várias conexões na mesma página. • Compatibilidade para carregar o SDK a partir de uma CDN.

1.0.3 — 1º de setembro de 2021

Versão	Notas de versão
• Versão semântica: 1.0.3 • Compilação: 202	Novos recursos • Compatibilidade experimental para WebCodecs. Isso está desativado por padrão e deve ser ativado por meio do objeto `ConnectionConfig` usando a nova propriedade `enableWebCodecs` . • Área de transferência: adicionada compatibilidade para tipos de dados `image/png` em navegadores baseados em Chromium. • Adicionado observador/retorno de chamada para obter a captura de tela do servidor como uma imagem PNG (requer o servidor do NICE DCV 2021.2). Alterações e correções de bugs • Manipulação aprimorada dos modificadores de teclado.

1.0.2 — 30 de julho de 2021

Versão	Notas de versão
• Versão semântica: 1.0.2 • Compilação: 167	• Corrigimos a detecção de pressão para eventos com caneta.

Versão	Notas de versão
	• Compatibilidade aprimorada para layout de teclado coreano no Chrome.

1.0.1 — 31 de maio de 2021

Versão	Notas de versão
• Versão semântica: 1.0.1 • Compilação: 141	• Corrigida a propagação de erros de conexão e motivos de fechamento • Corrigida a atualização do progresso do fragmento de armazenamento de arquivos • Manuseio aprimorado da webcam • Processamento aprimorado de entrada de áudio

1.0.0 — 24 de março de 2021

Versão	Notas de versão
• Versão semântica: 1.0.0 • Compilação: 81	Versão inicial do SDK do Cliente da Web do NICE DCV.

Histórico do documento

A tabela a seguir descreve a documentação desta versão do SDK do Cliente da Web do NICE DCV.

Alteração	Descrição	Data
Versão 1.5.6 do SDK do Cliente da Web do NICE DCV	O SDK do Cliente da Web do NICE DCV 1.5.6 já está disponível. Para obter mais informações, consulte SDK v.1.5.6.	9 de novembro de 2023
Versão 1.4.4 do SDK do Cliente da Web do NICE DCV	O SDK do Cliente da Web do NICE DCV 1.4.4 já está disponível. Para obter mais informações, consulte SDK v.1.4.4.	29 de junho de 2023
Versão 1.4.0 do SDK do Cliente da Web do NICE DCV	O SDK do Cliente da Web do NICE DCV 1.4.0 já está disponível. Para obter mais informações, consulte SDK v.1.4.0.	28 de março de 2023
Versão 1.3.1 do SDK do Cliente da Web do NICE DCV	O SDK do Cliente da Web do NICE DCV 1.3.1 já está disponível. Para obter mais informações, consulte SDK v.1.3.1.	9 de dezembro de 2022
Versão 1.3.0 do SDK do Cliente da Web do NICE DCV	O SDK do Cliente da Web do NICE DCV 1.3.0 já está disponível. Para obter mais informações, consulte SDK v.1.3.0.	11 de novembro de 2022
Versão 1.2.0 do SDK do Cliente da Web do NICE DCV	O SDK do Cliente da Web do NICE DCV 1.2.0 já está	29 de junho de 2022

Alteração	Descrição	Data
	disponível. Para obter mais informações, consulte SDK v.1.2.0.	
Versão 1.1.0 do SDK do Cliente da Web do NICE DCV	O SDK do Cliente da Web do NICE DCV 1.1.0 já está disponível. Para obter mais informações, consulte SDK v.1.1.0.	23 de fevereiro de 2022
Versão 1.0.1 do SDK do Cliente da Web do NICE DCV	Corrigidos alguns erros de digitação. Corrigidos pequenos bugs, consulte SDK v.1.0.1.	31 de maio de 2021
Versão inicial	Primeira publicação deste conteúdo.	24 de março de 2021

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.