aws

Guía para desarrolladores de SDK para clientes web

Amazon DCV

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon DCV: Guía para desarrolladores de SDK para clientes web

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es el Amazon DCV Web Client SDK? 1
Requisitos previos 2
Características admitidas 2
Compatibilidad con navegadores 2
Convención de control de versiones 3
Introducción 4
Conéctese a un servidor Amazon DCV y obtenga el primer fotograma 5
Paso 1: Preparar la página HTML 5
Paso 2: Autenticar, conectar y obtener el primer fotograma 6
Extra: crear automáticamente un formulario de inicio de sesión HTML 9
Trabajo con características de Amazon DCV 10
Comprender la función de devolución de llamada de featuresUpdate 10
Gestión de la actualización de características 11
Uso del Amazon DCV Web UI SDK 11
Requisitos previos 12
Paso 1: Preparar la página HTML 13
Paso 2: Autenticar, conectar y representar el componente React de `DCVViewer`. 13
Actualizar de AWS-UI a Cloudscape Design System 18
Referencia del SDK 19
Módulo DCV 19
Métodos 19
Miembros 22
Definiciones de tipo y devolución de llamada 27
Clase de conexión 68
Métodos 19
Clase de autenticación 97
Métodos 19
Clase de recurso 99
Métodos 19
Amazon DCV Web UI SDK 100
Componentes 100
Notas de la versión e historial de revisión 108
Notas de la versión 108
1.8.7 — 31 de octubre de 2024 109

1.8.4: 1 de octubre de 2024 109
1.5.10: 19 de diciembre de 2023 110
1.5.6: 9 de noviembre de 2023 110
1.4.4: 29 de junio de 2023 111
1.4.0: 28 de marzo de 2023 112
1.3.1: 9 de diciembre de 2022 113
1.3.0: 11 de noviembre de 2022 114
1.2.1 : 21 de julio de 2022 115
1.2.0: 29 de junio de 2022 115
1.1.3: 23 de mayo de 2022 116
1.1.2: 19 de mayo de 2022 116
1.1.1: 23 de marzo de 2022 117
1.1.0: 23 de febrero de 2022 117
1.0.4: 20 de diciembre de 2021 118
1.0.3: 1 de septiembre de 2021 118
1.0.2: 30 de julio de 2021 119
1.0.1: 31 de mayo de 2021 120
1.0.0: 24 de marzo de 2021 120
Historial de documentos 120
..... cxxiii

¿Qué es el Amazon DCV Web Client SDK?

> **Note**
>
> Amazon DCV se conocía anteriormente como NICE DCV.

Amazon DCV es un protocolo de visualización remota de alto rendimiento. Le permite entregar de forma segura escritorios remotos y transmisiones de aplicaciones desde cualquier nube o centro de datos a cualquier dispositivo, en diferentes condiciones de red. Al utilizar Amazon DCV con Amazon EC2, puede ejecutar aplicaciones con uso intensivo de gráficos de forma remota en las instancias de Amazon. EC2 A continuación, puede transmitir los resultados a máquinas cliente más modestas, lo que elimina la necesidad de costosas estaciones de trabajo específicas.

El SDK de Amazon DCV Web Client es una JavaScript biblioteca que puede utilizar para desarrollar sus propias aplicaciones cliente para navegadores web Amazon DCV. Los usuarios finales pueden utilizar estas aplicaciones para conectarse e interactuar con una sesión de Amazon DCV en ejecución.

Con el Amazon DCV Web Client SDK como elemento básico, puede crear aplicaciones web personalizadas que proporcionen a los usuarios acceso instantáneo a sus escritorios o aplicaciones desde cualquier lugar, con un rendimiento fluido y una capacidad de respuesta casi indistinguibles de los de una aplicación instalada de forma nativa.

Esta guía explica cómo utilizar el Amazon DCV Web Client SDK para crear aplicaciones cliente de navegador web personalizadas para interactuar con las sesiones de Amazon DCV dentro de sus flujos de trabajo.

Temas

- Requisitos previos
- Características admitidas
- Compatibilidad con navegadores
- Convención de control de versiones

Requisitos previos

Antes de empezar a trabajar con el Amazon DCV Web Client SDK, asegúrese de estar familiarizado con Amazon DCV y con las sesiones de Amazon DCV. Para obtener más información, consulte la Guía del administrador de Amazon DCV.

El Amazon DCV Web Client SDK es compatible con la versión 2020 y posteriores del servidor Amazon DCV.

Características admitidas

Puede crear aplicaciones cliente de navegador web personalizadas que admitan las siguientes características de Amazon DCV:

- Conexión con servidores Amazon DCV de Windows
- Conexión con servidores Amazon DCV de Linux
- Administrar modos de transmisión
- Transferencia de archivos
- Imprimir desde sesiones
- Copiar y pegar
- Reproducción de audio estéreo 2.0
- Grabación de audio estéreo 2.0 (en servidores Windows)
- Touchscreen
- Lápiz (en Linux, Windows 10 y servidores Windows Server 2019)
- Compatibilidad con varios monitores

Para obtener más información sobre estas características, consulte Características compatibles en la Guía del usuario de Amazon DCV.

Compatibilidad con navegadores

El SDK de Amazon DCV Web Client admite JavaScript (ES6) y se puede utilizar desde nuestras JavaScript TypeScript aplicaciones.

El Amazon DCV Web Client SDK es compatible con los siguientes navegadores web:

Navegador	Versión
Google Chrome	Tres últimas versiones importantes
Mozilla Firefox	Tres últimas versiones importantes
Microsoft Edge	Tres últimas versiones importantes
Apple Safari para macOS	Tres últimas versiones principales

Convención de control de versiones

La versión del Amazon DCV Web Client SDK se define con el siguiente formato: *`major`*`.`*`minor`*`.`*`patch`*. La convención de control de versiones generalmente se ajusta al modelo de control de versiones semántico. Un cambio en la versión principal, por ejemplo, de `1.x.x` a `2.x.x`, indica que se han introducido cambios importantes que podrían requerir cambios de código y una implementación planificada. Un cambio en una versión secundaria, por ejemplo, de `1.1.x` a `1.2.x`, es compatible con versiones anteriores, pero puede incluir elementos obsoletos.

Introducción a Amazon DCV Web Client SDK

El Amazon DCV Web Client SDK consta de un archivo `dcv.js` principal y algunos componentes auxiliares. Todos los archivos se distribuyen dentro de un archivo comprimido que se puede descargar del sitio web de Amazon DCV.

Para comenzar a trabajar con Amazon DCV Web Client SDK

1. El archivo Amazon DCV Web Client SDK está firmado digitalmente con una firma GPG segura. Para verificar la firma del archivo debe importar la clave NICE GPG. Para ello, abra una ventana de terminal e importe la clave NICE GPG.

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

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

2. Descargue el archivo de Amazon DCV Web Client SDK y la firma de archivo de Amazon DCV Web Client SDK del sitio web de Amazon DCV.
3. Verifique la firma del archivo de Amazon DCV Web Client SDK mediante la firma.

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

   Por ejemplo:

   ```
   $ gpg --verify nice-dcv-web-client-sdk-1.8.7-858.zip.sign nice-dcv-web-client-
   sdk-1.8.7-858.zip
   ```

4. Si la firma se verifica correctamente, extraiga el contenido del archivo Amazon DCV Web Client SDK y coloque el directorio extraído en su servidor web. Por ejemplo:

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

```
/
server_directory
/
```

> **Important**
>
> - Debe retener la estructura de carpetas al implementar el Amazon DCV Web Client SDK en el servidor web.
> - Cuando utilice el SDK de interfaz de usuario web Amazon DCV, tenga en cuenta que el componente `DCVViewer` React espera que los archivos EULA.txt y third-party-licenses .txt de este paquete estén presentes en la ruta URL del servidor web integrado. El third-party-licenses archivo.txt debe modificarse para incluir también el contenido del archivo correspondiente del paquete SDK de Amazon DCV Web Client y, posiblemente, cualquier otra información de licencia de las bibliotecas utilizadas por la aplicación de usuario consumidora.

Conéctese a un servidor Amazon DCV y obtenga el primer fotograma

El siguiente tutorial le muestra cómo preparar la página HTML para el cliente web personalizado, cómo autenticarse y conectarse a un servidor Amazon DCV y cómo recibir el primer fotograma de contenido en streaming de la sesión de Amazon DCV.

Temas

- Paso 1: Preparar la página HTML
- Paso 2: Autenticar, conectar y obtener el primer fotograma
- Extra: crear automáticamente un formulario de inicio de sesión HTML

Paso 1: Preparar la página HTML

En su página web, debe cargar los JavaScript módulos necesarios y añadir un elemento `<div>` HTML con un `id` lugar válido en el que desee que el SDK de Amazon DCV Web Client dibuje el flujo de contenido del servidor remoto de Amazon DCV.

Por ejemplo:

```
<!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>
```

Paso 2: Autenticar, conectar y obtener el primer fotograma

En esta sección se muestra cómo completar el proceso de autenticación del usuario, cómo conectar el servidor Amazon DCV y cómo recibir el primer fotograma de contenido desde el servidor Amazon DCV.

Primero, desde el archivo `index.js`, importe el Amazon DCV Web Client SDK. Se puede importar como módulo de Definición de módulo universal (UMD) del siguiente modo:

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

De lo contrario, a partir de la versión`1.1.0`, también se puede importar como un ECMAScript módulo (ESM) desde el paquete correspondiente, de la siguiente manera:

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

Defina las variables que se utilizarán para almacenar el objeto de autenticación, el objeto de conexión y la URL del servidor Amazon DCV.

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

Al cargar el script, registre la versión del Amazon DCV Web Client SDK y, al cargar la página, llame a la función `main`.

```
console.log("Using Amazon DCV Web Client SDK version " + dcv.version.versionStr);
document.addEventListener('DOMContentLoaded', main);
```

La función `main` establece el nivel de registro e inicia el proceso de autenticación.

```
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
    }
  );
}
```

Las funciones `promptCredentials`, `error` y `success` son funciones de devolución de llamada obligatorias que deben definirse en el proceso de autenticación.

Si el servidor Amazon DCV solicita credenciales, la función de devolución de llamada `promptCredentials` recibe la impugnación de credenciales solicitada del servidor Amazon DCV. Si el servidor Amazon DCV está configurado para utilizar la autenticación del sistema, se deben proporcionar las credenciales de inicio de sesión. En los siguientes ejemplos de código se da por sentado que el nombre de usuario es `my_dcv_user` y que la contraseña es `my_password`.

Si la autenticación produce error, la función de devolución de llamada `error` recibe un objeto de error del servidor Amazon DCV.

Si la autenticación se realiza correctamente, la función de devolución de llamada `success` recibe una matriz de pares que incluye el identificador de sesión (`sessionId`) y los tokens de autorización (`authToken`) para cada sesión a la que el usuario de `my_dcv_user` esté autorizado a conectarse en el servidor Amazon DCV. El siguiente ejemplo de código llama a la función connect y se conecta a la primera sesión devuelta en la matriz.

> **Note**
>
> En el siguiente ejemplo de código, sustituya `MY_DCV_USER` por su propio nombre de usuario y `MY_PASSWORD` por su propia contraseña.

```
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);
}
```

Conéctese al servidor Amazon DCV. Se llama al método de devolución de llamada `firstFrame` cuando se recibe el primer fotograma del servidor Amazon 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);
  });
}
```

Extra: crear automáticamente un formulario de inicio de sesión HTML

El objeto `challenge` se devuelve cuando se llama a la función de devolución de llamada `promptCredentials`. Incluye una propiedad denominada `requiredCredentials` que es una matriz de objetos, un objeto por credencial solicitada por el servidor Amazon DCV. Cada objeto incluye el nombre y el tipo de la credencial solicitada. Puede utilizar los objetos `challenge` y `requiredCredentials` para crear automáticamente un formulario de inicio de sesión HTML.

El siguiente ejemplo de código muestra cómo hacerlo.

```
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));
}
```

Trabajo con características de Amazon DCV

La disponibilidad de las características de Amazon DCV depende de los permisos configurados para la sesión de Amazon DCV y de las capacidades del navegador web del cliente.

Las características que están disponibles en una sesión de Amazon DCV se administran mediante los permisos que se han especificado para la sesión. Esto significa que, incluso si una característica es compatible con el Amazon DCV Web Client SDK, es posible que se impida el acceso a esa característica en función de los permisos definidos por el administrador de la sesión. Para obtener más información, consulte Configuración de la autorización de Amazon DCV en la Guía del administrador de Amazon DCV.

Comprender la función de devolución de llamada de featuresUpdate

Cuando cambia la disponibilidad de una característica en una sesión de Amazon DCV, el Amazon DCV Web Client SDK se lo notifica mediante la función de devolución de llamada `featuresUpdate` que especifique al establecer la conexión. Por ejemplo:

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

```
  ...
},
```

La función de devolución de llamada solo le notifica las características cuya disponibilidad ha cambiado. El parámetro `list` es una matriz de cadenas e incluye solo los nombres de las características actualizadas. Por ejemplo, si la disponibilidad de la característica de entrada de audio cambia para la sesión, el parámetro solo incluye `["audio-in"]`. Si en un momento posterior, cambia la disponibilidad de las características de copiar y pegar del portapapeles durante la sesión, el parámetro solo incluye `["clipboard-copy", "clipboard-paste"]`.

Gestión de la actualización de características

La función de devolución de llamada `featuresUpdate` solo le notifica que la disponibilidad de una o varias características ha cambiado. Para saber qué características se actualizaron, debe consultarlas mediante el método `connection.queryFeature`. Esto se puede hacer en cualquier momento después de recibir la notificación de cambio. Este método devuelve una `Promise` que pasa al estado actualizado de la característica solicitada. El valor `status` siempre está asociado y tiene una propiedad booleana (`true` | `false`) llamada `enabled`. Es posible que algunas características tengan propiedades adicionales en el valor `status`. Si la disponibilidad de la característica no se ha actualizado, se rechaza.

El siguiente ejemplo de código muestra cómo hacerlo.

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

Uso del Amazon DCV Web UI SDK

El siguiente tutorial muestra cómo autenticarse en el servidor Amazon DCV, conectarse a él y representar el componente React de `DCVViewer` desde el Amazon DCV Web UI SDK.

Temas

- Requisitos previos
- Paso 1: Preparar la página HTML
- Paso 2: Autenticar, conectar y representar el componente React de DCVViewer.
- Actualizar de AWS-UI a Cloudscape Design System

Requisitos previos

Debe instalar `React` , `ReactDOM` , `Cloudscape Design Components React` , `Cloudscape Design Global Styles` y `Cloudscape Design Design Tokens` .

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

También debería descargar `Amazon DCV Web Client SDK` . Consulte Introducción a Amazon DCV Web Client SDK la step-by-step guía sobre cómo hacerlo.

Debe crear un alias para importar el módulo `dcv`, ya que es una dependencia externa del Amazon DCV Web UI SDK. Por ejemplo, si está utilizando una paquete web para agrupar su aplicación web, puede utilizar la opción resolve.alias del siguiente modo:

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

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

Si está utilizando acumulación para agrupar, puede instalar @rollup/plugin-alias y utilizarlo de este modo:

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

module.exports = {
  //...
```

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

Paso 1: Preparar la página HTML

En su página web, debe cargar los JavaScript módulos necesarios y debe tener un elemento `<div>` HTML con un valor válido en el que `id` se renderizará el componente de entrada de su aplicación.

Por ejemplo:

```
<!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>
```

Paso 2: Autenticar, conectar y representar el componente React de **`DCVViewer`**.

En esta sección se muestra cómo completar el proceso de autenticación del usuario, cómo conectar el servidor Amazon DCV y cómo representar el componente React de `DCVViewer`.

En primer lugar, desde el archivo `index.js`, importe `React` , `ReactDOM` y el `App` componente de nivel superior .

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

Represente el nodo de contenedor de nivel superior de la aplicación.

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

En el archivo `App.js`, importe el Amazon DCV Web Client SDK como módulo ESM, el componente React de `DCVViewer` desde el Amazon DCV Web UI SDK, `React` y el paquete `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 continuación se muestra un ejemplo que muestra cómo autenticarse en el servidor Amazon DCV y representar el componente React de `DCVViewer` desde el Amazon DCV Web UI SDK, siempre que la autenticación se haya realizado correctamente.

```
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;
```

Las funciones `promptCredentials` , `error` y `success` son funciones de devolución de llamada obligatorias que deben definirse en el proceso de autenticación.

Si el servidor Amazon DCV solicita credenciales, la función de devolución de llamada `promptCredentials` recibe la impugnación de credenciales solicitada del servidor Amazon DCV. Si el servidor Amazon DCV está configurado para utilizar la autenticación del sistema, se deben proporcionar las credenciales en forma de nombre de usuario y contraseña.

Si la autenticación produce error, la función de devolución de llamada `error` recibe un objeto de error del servidor Amazon DCV.

Si la autenticación se realiza correctamente, la función de devolución de llamada `success` recibe una matriz de pares que incluye el identificador de sesión (`sessionId`) y los tokens de autorización (`authToken`) para cada sesión a la que el usuario esté autorizado a conectarse en el servidor Amazon DCV. El ejemplo de código anterior actualiza el estado React para representar el componente `DCVViewer` si la autenticación se realiza correctamente.

Para obtener más información sobre las propiedades que acepta este componente, consulte la referencia del Amazon DCV Web UI SDK.

Para obtener más información sobre certificados autofirmados, consulte las Redirection clarifications with self-signed certificates.

Actualizar de AWS-UI a Cloudscape Design System

A partir del SDK versión 1.3.0 actualizamos nuestro componente `DCVViewer` desde AWS-UI hasta su evolución: Cloudscape Design.

Cloudscape utiliza un tema visual diferente al de AWS-UI, pero la base de código subyacente sigue siendo la misma. Por tanto, debería resultar fácil la migración de una aplicación basada en el `DCVViewer`. Para migrar, sustituya los paquetes NPM relacionados con AWS-UI que haya instalado por los paquetes de Cloudscape asociados:

Nombre del paquete de AWS-UI	Nombre del paquete de 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 obtener más información sobre la migración, consulte la página de documentación de AWS-UI GitHub .

Referencia del SDK

En esta sección se proporcionan descripciones, sintaxis y ejemplos de uso del Amazon DCV Web Client SDK.

Temas

- Módulo DCV
- Clase de conexión
- Clase de autenticación
- Clase de recurso
- Amazon DCV Web UI SDK

Módulo DCV

Módulo que implementa el lado del cliente del protocolo DCV.

Expone

- Métodos
- Miembros
- Definiciones de tipo y devolución de llamada

Métodos

Enumeración

- authenticate(url, callbacks) → {Authentication}
- connect (config) → {Promesa. < Conexión >|Promesa. < {código: ConnectionErrorCode, mensaje: cadena} >}
- setLogHandler(controlador) → {vacío}
- setLogLevel(nivel) → {vacío}

authenticate(url, callbacks) → {Authentication}

Inicia el proceso de autenticación para el punto de conexión del servidor Amazon DCV especificado.

Parámetros:

Nombre	Tipo	Descripción
`url`	cadena	Nombre y puerto de host del servidor Amazon DCV en ejecución con el siguiente formato: `https://dcv_host_address:port` . Por ejemplo: `https://my-dcv-server:8443` .
`callbacks`	authenticationCallbacks	Devoluciones de llamada que están disponibles para invocarse durante el proceso de autenticación.

Devuelve:

- El objeto de autenticación.

Tipo

Autenticación

connect (config) → {Promesa. < Conexión >|Promesa. < {código: ConnectionErrorCode, mensaje: cadena} >}

Se conecta al punto de conexión del servidor Amazon DCV especificado. Si la conexión se realiza correctamente, devuelve un objeto Connection. Si se produce un error de conexión, devuelve un objeto de error.

Parámetros:

Nombre	Tipo	Descripción
`config`	ConnectionConfig	El ConnectionConfig objeto.

Devuelve:

- Un objeto Connection o un objeto de error.

Tipo

Lo prometo. < Conexión > | Promesa. < {código: ConnectionErrorCode, mensaje: cadena} >

setLogHandler(controlador) → {vacío}

Establece una función de controlador de registros personalizada. Cuando se sustituye el controlador de registros predeterminado, la posición original de la entrada de registro se perderá al depurar con la consola del navegador.

Parámetros:

Nombre	Tipo	Descripción
`handler`	función	La función de controlador de registros personalizada. La función de controlador contiene level (número), levelName (cadena), domain (cadena) y message (cadena).

Devuelve:

Tipo

void

setLogLevel(nivel) → {vacío}

Establece el nivel de registro. Solo es necesario si se utiliza el controlador de registro predeterminado.

Parámetros:

Nombre	Tipo	Descripción
`level`	LogLevel	Nivel de registro que se debe utilizar.

Devuelve:

Tipo

void

Miembros

Enumeración

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

(constante) AudioError: AudioErrorCode

La AudioError enumeración de códigos.

Tipo:

- AudioErrorCode

(constante) AuthenticationError: AuthenticationErrorCode

La AuthenticationError enumeración de códigos.

Tipo:

- AuthenticationErrorCode

(constante) ChannelError: ChannelErrorCode

La ChannelError enumeración de códigos.

Tipo:

- ChannelErrorCode

(constante) ClosingReasonError: ClosingReasonErrorCode

La ClosingReasonError enumeración de códigos.

Tipo:

- ClosingReasonErrorCode

(constante) ConnectionError: ConnectionErrorCode

La ConnectionError enumeración de códigos.

Tipo:

- ConnectionErrorCode

(constante) CustomChannelError: CustomChannelErrorCode

La CustomChannelError enumeración de códigos.

Tipo:

- CustomChannelErrorCode

(constante) DisplayConfigError: DisplayConfigErrorCode

La DisplayConfigError enumeración de códigos.

Tipo:

- DisplayConfigErrorCode

(constante) FileStorageError: FileStorageErrorCode

La FileStorageError enumeración de códigos.

Tipo:

- FileStorageErrorCode

(constante) LogLevel: LogLevel

Niveles de registro del SDK disponibles.

Tipo:

- LogLevel

(constante) MultiMonitorError: MultiMonitorErrorCode

La MultiMonitorError enumeración de códigos.

Tipo:

- MultiMonitorErrorCode

(constante) ResolutionError: ResolutionErrorCode

La ResolutionError enumeración de códigos.

Tipo:

- ResolutionErrorCode

(constante) TimezoneRedirectionError: TimezoneRedirectionErrorCode

La TimezoneRedirectionError enumeración de códigos.

Tipo:

- TimezoneRedirectionErrorCode

(constante) TimezoneRedirectionSetting: TimezoneRedirectionSettingCode

La TimezoneRedirectionSetting enumeración de códigos.

Tipo:

- TimezoneRedirectionSettingCode

(constante) TimezoneRedirectionStatus: TimezoneRedirectionStatusCode

La TimezoneRedirectionStatus enumeración de códigos.

Tipo:

- TimezoneRedirectionStatusCode

(constant) version

Versión de Amazon DCV con las opciones major, minor, patch, revision, extended y versionStr.

Propiedades:

Nombre	Tipo	Descripción
`major`	entero	Número de la versión principal .
`minor`	entero	Número de la versión secundaria.
`patch`	entero	Número de versión del parche.
`revision`	entero	Número de revisión.
`extended`	cadena	Cadena ampliada.
`versionStr`	cadena	Concatenación de los números de versión principal , secundaria, de parche y de revisión con el formato `major.minor.patch+build.revision` .

(constante) ScreenshotError: ScreenshotErrorCode

La ScreenshotError enumeración de códigos.

Tipo:

- ScreenshotErrorCode

(constante) WebcamError: WebcamErrorCode

La WebcamError enumeración de códigos.

Tipo:

- WebcamErrorCode

Definiciones de tipo y devolución de llamada

Enumeración

- AudioErrorCode
- authenticationCallbacks
- AuthenticationErrorCode
- authErrorCallback(autenticación, error)
- authPromptCredentialsDevolución de llamada (autenticación, impugnación)
- authSuccessCallback(autenticación, datos de autenticación)
- Canal
- ChannelErrorCode
- clipboardEventCallback(evento)
- ClosingReasonErrorCode
- Colorspace
- connectionCallbacks
- ConnectionConfig
- ConnectionErrorCode
- createDirectory(path)
- CustomChannelErrorCode
- dataChannelCallback(información)
- deleteFile(path)
- deviceChangeEventLlamada ()
- disconnectCallback(reason)
- displayAvailabilityCallback(estado, displayID)
- DisplayConfigErrorCode
- displayLayoutCallback(ServerWidth, ServerHeight, heads)
- característica
- featuresUpdateCallback(Lista de características)
- fileDownloadCallback(Recurso de archivo)
- filePrintedCallback(Recurso de impresión)

- filestorage
- filestorageEnabledCallback(habilitado)
- FileStorageErrorCode
- firstFrameCallback(Resize Enabled, Enabled, DisplayID relativeMouseMode)
- idleWarningNotificationLlamada de vuelta () disconnectionDateTime
- collaboratorListCallback(colaboradores)
- licenseNotificationCallback(notificación)
- list(path)
- LogLevel
- Supervisión
- MultiMonitorErrorCode
- qualityIndicatorStateLlamada de vuelta (estado)
- renameDirectory(src, dest)
- renameFile(src, dest)
- ResolutionErrorCode
- retrieveFile(path)
- screenshotCallback(screenshot)
- ScreenshotErrorCode
- serverInfo
- stats
- storeFile(file, dir)
- TimezoneRedirectionErrorCode
- TimezoneRedirectionSettingCode
- TimezoneRedirectionStatusCode
- WebcamErrorCode

AudioErrorCode

Las enumeraciones AudioError de códigos disponibles en el módulo DCV

- `SETTING_AUDIO_FAILED`

- `CHANNEL_NOT_AVAILABLE`

Tipo:

- número

authenticationCallbacks

Devoluciones de llamada de autenticación

Tipo:

- Objeto

Propiedades:

Nombre	Tipo	Descripción
`promptCredentials`	authPromptCredentialsLlamada de vuelta	Función de devolución de llamada que se invoca cuando se solicitan credenciales al usuario.
`error`	authErrorCallback	Función de devolución de llamada que se invoca cuando se produce un error en la autenticación.
`success`	authSuccessCallback	Función de devolución de llamada que se invoca cuando la autenticación es correcta.

AuthenticationErrorCode

Las enumeraciones AuthenticationError de códigos disponibles en el 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(autenticación, error)

Función de devolución de llamada que se invoca cuando se produce un error en la autenticación.

Parámetros:

Nombre	Tipo	Descripción
authentication	Autenticación	El objeto de autenticación.
error	Objeto	Objeto de error generado por el proceso de autenticación. Nombre: code — Tipo: AuthenticationErrorCode — Descripción: Código de error.

Nombre	Tipo	Descripción
		Nombre: `message` — Tipo: cadena — Descripción: Mensaje de error.

authPromptCredentialsDevolución de llamada (autenticación, impugnación)

Función de devolución de llamada que se invoca cuando se solicitan credenciales al usuario. El usuario debe responder a la solicitud proporcionando las credenciales requeridas.

Parámetros:

Nombre	Tipo	Descripción
`authentication`	Autenticación	El objeto de autenticación.
`challenge`	Objeto	La solicitud. Nombre: `requiredCredentials` — Tipo: Array.<Object> — Descripción: Matriz de objetos de credenciales solicitados.

Nombre	Tipo	Descripción
		Nombre: Tipo: Descripción n: c: Nombre de la credencial solicitada. t: c: Tipo de la credencial solicitada.

authSuccessCallback(autenticación, datos de autenticación)

Función de devolución de llamada que se invoca cuando la autenticación es correcta.

Parámetros:

Nombre	Tipo	Descripción
`authentication`	Autenticación	El objeto de autenticación.
`authenticationData`	Array.<Object>	Conjunto de objetos que incluyen tokens de sesión IDs

<table>
<tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td></td><td></td><td>y autenticación de Amazon DCV.

<table><tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr><tr><td>sessior</td><td>cadena</td><td>ID de la sesión de Amazon DCV.</td></tr><tr><td>authTok</td><td>cadena</td><td>Token de autenticación de la sesión de Amazon DCV.</td></tr></table></td></tr>
</table>

Canal

Canales disponibles que se pueden especificar.

Tipo:

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

ChannelErrorCode

Las enumeraciones ChannelError de códigos están disponibles en el módulo DCV

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

Tipo:

- número

clipboardEventCallback(evento)

Función de devolución de llamada que se invoca cuando se genera un `clipboardEvent`.

Parámetros:

<table>
<tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td><code>event</code></td><td>Objeto</td><td>Información sobre el evento del portapapeles.
<table>
<tr><th>Nombr</th><th>Tipo</th><th>Atribut</th><th>Descripción</th></tr>
<tr><td><code>name</code></td><td>establ do | copiar | pegar | | dataSi lert | autoC one newD ailable autoP Done |</td><td></td><td>Siempre están presentes . El nombre del evento.</td></tr>
</table>
</td></tr>
</table>

Nombre	Tipo	Descripción
		Nombr Tipo Atribut Descripción Remot or | pasteA lableD clipb Data Objeto | caden Datos del portapapeles. autoC boolea <optio> Indica si está habilitada la copia automática del portapapeles de la sesión en el portapapeles del cliente local.

Nombre	Tipo	Descripción
		Nombr Tipo Atribut Descripción maxDa número <optio> Cantidad máxima de datos que se pueden colocar en el portapapeles. error cadena <optio> Información de error, en su caso.

ClosingReasonErrorCode

Las enumeraciones ClosingReasonError de códigos disponibles en el 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`
- `EXTERNAL_PROTOCOL_CONNECTION_EVICTED`
- `DISCONNECTION_REQUESTED`

Tipo:

- número

Colorspace

Espacios de color disponibles que se pueden especificar.

Tipo:

- «RGB» | «YUV_ REC6 01" | «YUV_ 09" REC7

connectionCallbacks

Devoluciones de llamada disponibles para invocar en caso de un error de conexión.

Tipo:

- Objeto

Propiedades:

Nombre	Tipo	Descripción
`disconnect`	disconnectCallback	Función de devolución de llamada que se invoca cuando finaliza la conexión.
`displayLayout`	displayLayoutCallback	Función de devolución de llamada que se invoca cuando

Nombre	Tipo	Descripción
		cambia el diseño o la resolución de pantalla.
`displayAvailability`	displayAvailabilityCallback	Función de devolución de llamada que se invoca cuando cambia la disponibilidad de una pantalla.
`firstFrame`	firstFrameCallback	Función de devolución de llamada que se invoca cuando se recibe el primer fotograma del servidor Amazon DCV.
`filePrinted`	filePrintedCallback	Función de devolución de llamada que se invoca cuando se imprime un archivo en el servidor Amazon DCV.
`fileDownload`	fileDownloadCallback	Función de devolución de llamada que se invoca cuando hay un archivo listo para descargar desde el servidor Amazon DCV.
`dataChannel`	dataChannelCallback	Función de devolución de llamada que se invoca cuando el servidor Amazon DCV envía una notificación sobre la disponibilidad de un canal de datos.
`licenseNotification`	licenseNotificationCallback	Función de devolución de llamada que se invoca cuando el servidor Amazon DCV envía una notificación sobre el estado de licencia.

Nombre	Tipo	Descripción
`idleWarningNotification`	idleWarningNotificationLlamada de vuelta	Función de devolución de llamada que se invoca cuando el servidor Amazon DCV envía una advertencia de tiempo de espera de inactividad.
`collaboratorList`	collaboratorListCallback	Función de devolución de llamada que se invoca cuando el servidor Amazon DCV envía la lista de colaboradores (desde Amazon DCV Web Client SDK versión 1.1.0).
`qualityIndicatorState`	qualityIndicatorStateLlamada de vuelta	Función de devolución de llamada que se invoca cuando el indicador de calidad de la conexión cambia de estado.
`filestorageEnabled`	filestorageEnabledCallback	Función de devolución de llamada que se invoca cuando el almacenamiento de archivos está habilitado o deshabilitado.
`featuresUpdate`	featuresUpdateCallback	Función de devolución de llamada que se invoca cuando cambia el estado de una característica.
`clipboardEvent`	clipboardEventCallback	Función de devolución de llamada que se invoca cuando se genera un `clipboardEvent` .

Nombre	Tipo	Descripción
`deviceChangeEvent`	deviceChangeEventLlamada de vuelta	Función de devolución de llamada que se invoca cuando se desencadena un evento `deviceChange` .
`screenshot`	screenshotCallback	Función de devolución de llamada que se invoca cuando a está disponible una `screenshot` .

ConnectionConfig

Configuración de una conexión Amazon DCV.

Tipo:

- Objeto

Propiedades:

Nombre	Tipo	Descripción
`url`	cadena	Nombre y puerto de host del servidor Amazon DCV en ejecución con el siguiente formato: `https://dcv_host_address:port` . Por ejemplo: `https://my-dcv-server:8443` .
`sessionId`	cadena	ID de la sesión de Amazon DCV.

Nombre	Tipo	Descripción
`authToken`	cadena	Token de autenticación que se utilizará al conectarse al servidor.
`baseUrl`	cadena	URL absoluta o relativa desde la que se cargarán los archivos del SDK.
`resourceBaseUrl`	cadena	URL absoluta o relativa desde la que se accederá a los recursos de DCV.
`enabledChannels`	Array.<Channel>	Indica la lista de canales que se pueden habilitar. Si no se especifica o se especific a una matriz vacía, el valor predeterminado es todos los canales disponibles.
`losslessColorspace`	Colorspace	Indica el espacio de color que se utilizará. Si no se especific a, toma el valor "RGB" como predeterminado.
`divId`	cadena	ID del objeto `div` en el HTML DOM donde el SDK debe crear el lienzo con el flujo remoto.
`volumeLevel`	entero	Nivel de volumen preferido. El intervalo válido es de 0 a 100.

Nombre	Tipo	Descripción
`clipboardAutoSync`	booleano	Indica si la copia automática del portapapeles de la sesión de Amazon DCV al portapapeles del cliente local está habilitada para los navegadores web compatibles.
`dynamicAudioTuning`	booleano	Indica si se debe sintonizar dinámicamente el audio en función de la configuración de audio del servidor Amazon DCV cuando se establece una conexión.
`clientHiDpiScaling`	booleano	Indica si se debe escalar el lienzo en función del DPI del cliente.
`highColorAccuracy`	booleano	Indica si se debe utilizar una alta precisión de color, si está disponible. Si no se especifica, toma el valor `false` como predeterminado.
`enableWebCodecs`	Booleano	Indica si se WebCodecs debe utilizar si está disponible. Toma `false` como valor predeterminado si no se especifica.
`observers`	connectionCallbacks	Funciones de devolución de llamada que se invocan para eventos relacionados con la conexión.

Nombre	Tipo	Descripción
`callbacks`	connectionCallbacks	Igual que la propiedad `observers` , pero cada devolución de llamada incluye el objeto Connection como primer parámetro.

ConnectionErrorCode

Las enumeraciones de ConnectionError códigos disponibles en el módulo DCV

- `ALREADY_OPEN`
- `INVALID_CONFIG`
- `INITIALIZATION_FAILED`
- `REJECTED`
- `MAIN_CHANNEL_ALREADY_OPEN`
- `GENERIC_ERROR` (desde DCV Server 2021.0)
- `INTERNAL_SERVER_ERROR` (desde DCV Server 2021.0)
- `AUTHENTICATION_FAILED` (desde DCV Server 2021.0)
- `PROTOCOL_ERROR` (desde DCV Server 2021.0)
- `INVALID_SESSION_ID` (desde DCV Server 2021.0)
- `INVALID_CONNECTION_ID` (desde DCV Server 2021.0)
- `CONNECTION_LIMIT_REACHED` (desde DCV Server 2021.0)
- `SERVER_UNREACHABLE` (desde DCV Server 2022.1)
- `GATEWAY_BUSY`
- `UNSUPPORTED_CREDENTIAL` (desde DCV Server 2022.2)
- `TRANSPORT_ERROR`

Tipo:

- número

createDirectory(path)

Parámetros:

Nombre	Tipo	Descripción
`path`	cadena	Ruta absoluta del servidor en el que queremos crear un directorio. También debe incluir el nombre del directorio de destino.

CustomChannelErrorCode

Las enumeraciones CustomChannelError de códigos disponibles en el módulo DCV

- `TRANSPORT_ERROR`

Tipo:

- número

dataChannelCallback(información)

Función de devolución de llamada que se invoca cuando el servidor Amazon DCV envía una notificación sobre la disponibilidad de un canal de datos.

Parámetros:

Nombre	Tipo	Descripción
`info`	Objeto	Información sobre el canal de datos.

Nombre	Tipo	Descripción
		<table><tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr><tr><td><code>name</code></td><td>cadena</td><td>Nombre del canal de datos.</td></tr><tr><td><code>token</code></td><td>cadena</td><td>Token de autenticación del canal de datos.</td></tr></table>

deleteFile(path)

Parámetros:

Nombre	Tipo	Descripción
`path`	cadena	Ruta absoluta del servidor que identifica el archivo que queremos eliminar.

deviceChangeEventLlamada ()

Función de devolución de llamada que se invoca cuando se desencadena un evento `deviceChange`.

disconnectCallback(reason)

Función de devolución de llamada que se invoca cuando finaliza la conexión.

Parámetros:

Nombre	Tipo	Descripción
`reason`	Objeto	Motivo de la desconexión. Nombre: `code` — Tipo: número — Descripción: Código de región. Nombre: `message` — Tipo: cadena — Descripción: Mensaje del motivo.

displayAvailabilityCallback(estado, displayID)

Función de devolución de llamada que se invoca cuando cambia la disponibilidad de una pantalla.

Parámetros:

Nombre	Tipo	Descripción
`status`	Objeto	Estado de la pantalla. Nombre: `enablec` — Tipo: booleanc — Descripción: Indica si la pantalla

<table>
<tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td></td><td></td><td>está habilitada.</td></tr>
<tr><td><code>closed</code></td><td>booleanc</td><td>Indica si la pantalla está habilitada.</td></tr>
</table>
</td></tr>
<tr><td><code>displayId</code></td><td>número</td><td>Identificador de la pantalla.</td></tr>
</table>

DisplayConfigErrorCode

Las enumeraciones DisplayConfigError de códigos disponibles en el módulo DCV

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

Tipo:

- número

displayLayoutCallback(ServerWidth, ServerHeight, heads)

Función de devolución de llamada que se invoca cuando cambia el diseño o la resolución de pantalla.

Parámetros:

Nombre	Tipo	Descripción
`serverWidth`	número	Ancho (en píxeles) de la pantalla principal.
`serverHeight`	número	Alto (en píxeles) de la pantalla principal.
`heads`	Array.<Monitor>	Cabezales de pantalla compatibles con el servidor Amazon DCV.

característica

Valores de características.

- `display`: indica la disponibilidad de una transmisión de vídeo de una sola pantalla.
- `display-multi`: indica la disponibilidad de una transmisión de vídeo de varias pantallas.
- `high-color-accuracy`: indica la disponibilidad de una alta precisión de color (desde Amazon DCV Web Client SDK versión 1.1.0).
- `mouse`: indica la disponibilidad de la funcionalidad del ratón.
- `keyboard`: indica la disponibilidad de la funcionalidad del teclado.
- `keyboard-sas`: indica la disponibilidad de la funcionalidad de secuencia SAS (Control + Alt + Eliminar).
- `relative-mouse`: indica la disponibilidad del modo de ratón relativo.
- `clipboard-copy`: indica la disponibilidad de la funcionalidad de copia del portapapeles desde el servidor Amazon DCV hasta el cliente.
- `clipboard-paste`: indica la disponibilidad de la funcionalidad de pegado del portapapeles desde el cliente hasta el servidor Amazon DCV.
- `audio-in`: indica la disponibilidad de la funcionalidad de entrada de audio utilizando el micrófono.
- `audio-out`: indica la disponibilidad de la funcionalidad de reproducción de audio.
- `webcam`: indica la disponibilidad de la funcionalidad de streaming de la cámara web.
- `file-download`: indica la disponibilidad de la funcionalidad de descarga desde el servidor Amazon DCV hasta el cliente.

- `file-upload`: indica la disponibilidad de la funcionalidad de carga desde el servidor Amazon DCV hasta el cliente.
- `timezone-redirection`: indica la disponibilidad de la funcionalidad de redireccionamiento de zona horaria (desde Amazon DCV Web Client SDK versión 1.3.0).

Tipo:

- cadena

featuresUpdateCallback(Lista de características)

Función de devolución de llamada que se invoca cuando cambia el estado de una característica.

Parámetros:

Nombre	Tipo	Descripción
`featuresList`	Array.<feature>	Matriz de características que han cambiado.

fileDownloadCallback(Recurso de archivo)

Función de devolución de llamada que se invoca cuando hay un archivo listo para descargar desde el servidor Amazon DCV.

Parámetros:

Nombre	Tipo	Descripción
`fileResource`	Objeto	Información sobre el archivo que está listo para su descarga.

Nombre	Tipo	Descripción
		<table><tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr><tr><td><code>id</code></td><td>cadena</td><td>Identificador del archivo.</td></tr><tr><td><code>url</code></td><td>cadena</td><td>URL que se utilizará para descargar el archivo.</td></tr><tr><td><code>domain</code></td><td>cadena</td><td>Dominio del recurso.</td></tr><tr><td><code>token</code></td><td>cadena</td><td>Token de autenticación que se utilizará para descargar el archivo. El token también se incluye</td></tr></table>

Nombre	Tipo	Descripción
		Nombre / Tipo / Descripción en la URL.

filePrintedCallback(Recurso de impresión)

Función de devolución de llamada que se invoca cuando se imprime un archivo en el servidor Amazon DCV.

Parámetros:

Nombre	Tipo	Descripción
`printResource`	Objeto	Información sobre el archivo impreso. `id` — cadena — Identificador del archivo impreso. `url` — cadena — URL que se utilizará para descargar el archivo impreso.

Nombre	Tipo	Descripción

Nombre	Tipo	Descripción
`domain`	cadena	Dominio del recurso. En este caso, `printer`.
`token`	cadena	Token de autenticación que se utilizará para descargar el archivo. El token también se incluye en la URL.

filestorage

Objeto que permite explorar y realizar acciones en el sistema de archivos.

Tipo:

- Objeto

Propiedades:

Nombre	Tipo	Descripción
`list`	list	Función que permite enumerar los elementos (archivos y directorios) presentes en la ruta proporcionada en el servidor.
`createDirectory`	createDirectory	Función que permite crear un directorio en la ruta especific ada en el servidor.
`retrieveFile`	retrieveFile	Función que permite descargar localmente un archivo en la ruta especificada en el servidor.
`deleteFile`	deleteFile	Función que permite descargar localmente un archivo en la ruta especificada en el servidor.
`renameFile`	renameFile	Función que permite cambiar el nombre de un archivo de la ruta de origen especificada a la ruta de destino especific ada.
`renameDirectory`	renameDirectory	Función que permite cambiar el nombre de un directorio de la ruta de origen especificada a la ruta de destino absoluta.

Nombre	Tipo	Descripción
`storeFile`	storeFile	Función que permite cargar un archivo local en la ruta proporcionada en el servidor.

filestorageEnabledCallback(habilitado)

Función de devolución de llamada que se invoca cuando el almacenamiento de archivos está habilitado. Canal en diferido solo en Internet Explorer 11.

Parámetros:

Nombre	Tipo	Descripción
`enabled`	booleano	Indica si el almacenamiento de archivos está habilitado.

FileStorageErrorCode

Las enumeraciones de FileStorageError códigos disponibles en el módulo DCV

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

Tipo:

- número

firstFrameCallback(Resize Enabled, Enabled, DisplayID relativeMouseMode)

Función de devolución de llamada que se invoca cuando se recibe el primer fotograma del servidor Amazon DCV. Se emite para cada pantalla.

Parámetros:

Nombre	Tipo	Descripción
`resizeEnabled`	booleano	Indica si el servidor admite el cambio de tamaño del diseño de la pantalla del cliente.
`relativeMouseModeEnabled`	booleano	Indica si el servidor admite el modo de ratón relativo.
`displayId`	número	Identificador de la pantalla.

idleWarningNotificationLlamada de vuelta () disconnectionDateTime

Función de devolución de llamada que se invoca cuando el servidor Amazon DCV envía una advertencia de tiempo de espera de inactividad.

Parámetros:

Nombre	Tipo	Descripción
`disconnectionDateTime`	Fecha	Fecha y hora de la desconexión.

collaboratorListCallback(colaboradores)

Función de devolución de llamada que se invoca cuando el servidor Amazon DCV envía la lista de colaboradores.

Parámetros:

Nombre	Tipo	Descripción
`collaborators`	Array.<Object>	Lista de objetos que contiene información sobre los colaboradores.

Nombre	Tipo	Descripción
`username`	cadena	Nombre de usuario del colaborador.
`owner`	booleano	Indica si el colaborador es el propietario de la sesión.
`connectionId`	número	Indica el ID asignado por el servidor a la conexión.

licenseNotificationCallback(notificación)

Función de devolución de llamada que se invoca cuando el servidor Amazon DCV envía una notificación sobre el estado de licencia.

Parámetros:

Nombre	Tipo	Descripción
`notification`	Objeto	Notificación.

Nombre	Tipo	Descripción
`product`	cadena	Producto DCV.
`status`	cadena	Estado de la licencia.
`message`	cadena	Mensaje.
`leftDay`	número	Número de días para que caduque la licencia.
`isDemo`	booleanc	Indica si la licencia es de demostración.

Nombre	Tipo	Descripción
		Nombre / Tipo / Descripción `numUnlicensed` / número / Número de conexiones sin licencia. `licensingMode` / cadena / Modelo de licencia. `documentationUrl` / cadena / URL de la documentación.

list(path)

Parámetros:

Nombre	Tipo	Descripción
`path`	cadena	Ruta absoluta del servidor cuyo contenido queremos enumerar.

LogLevel

Niveles de registro del SDK disponibles.

Tipo:

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

Supervisión

Tipo:

- Objeto

Propiedades:

Nombre	Tipo	Descripción
`name`	cadena	Nombre del cabezal de pantalla.
`rect`	Objeto	Información sobre el cabezal de pantalla. Nombre: `x` — Tipo: número — Descripción: Coordenada x inicial del cabezal de pantalla. Nombre: `y` — Tipo: número — Descripción: Coordenada y inicial del cabezal de pantalla. Nombre: `width` — Tipo: número — Descripción: Ancho (en píxeles)

<table>
<tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td></td><td></td><td>del cabezal de pantalla.</td></tr>
<tr><td><code>height</code></td><td>número</td><td>Alto (en píxeles) del cabezal de pantalla.</td></tr>
</table>
</td></tr>
<tr><td><code>primary</code></td><td>booleano</td><td>Indica si el cabezal de pantalla es el cabezal de la pantalla principal. Esto se determina desde el sistema operativo remoto, si está disponible.</td></tr>
<tr><td><code>dpi</code></td><td>número</td><td>DPI del cabezal de pantalla.</td></tr>
</table>

MultiMonitorErrorCode

Las enumeraciones de MultiMonitorError códigos disponibles en el módulo DCV

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

Tipo:

- número

qualityIndicatorStateLlamada de vuelta (estado)

Función de devolución de llamada que se invoca cuando el indicador de calidad de la conexión cambia de estado.

Parámetros:

Nombre	Tipo	Descripción
`state`	Array.<Object>	Información sobre la calidad de la conexión.

Nombre	Tipo	Descripción
`name`	cadena	Nombre del indicador.
`status`	NORMAL | WARNIN | CRITICA	Descripción del estado.
`changed`	booleanc	Indica si el estado ha cambiado.

renameDirectory(src, dest)

Parámetros:

Nombre	Tipo	Descripción
`src`	cadena	Ruta de origen absoluta en el servidor que identifica el directorio cuyo nombre queremos cambiar.
`dest`	cadena	Ruta de destino absoluta en el servidor que especifica la ruta de destino y el nombre del directorio.

renameFile(src, dest)

Parámetros:

Nombre	Tipo	Descripción
`src`	cadena	Ruta de origen absoluta en el servidor que identific a el archivo cuyo nombre queremos cambiar.
`dest`	cadena	Ruta de destino absoluta en el servidor que especifica la ruta de destino y el nombre del archivo.

ResolutionErrorCode

Las enumeraciones ResolutionError de códigos disponibles en el módulo DCV

- `INVALID_ARGUMENT`

- `NO_CHANNEL`
- `NOT_IMPLEMENTED`

Tipo:

- número

retrieveFile(path)

Parámetros:

Nombre	Tipo	Descripción
`path`	cadena	Ruta absoluta del servidor que identifica el archivo que queremos descargar localmente.

screenshotCallback(screenshot)

Función de devolución de llamada que se invoca cuando a está disponible una captura de pantalla.

Parámetros:

Nombre	Tipo	Descripción
`screenshot`	byte[]	Búfer de captura de pantalla en formato PNG o `null` si la recuperación de la captura de pantalla ha fallado.

ScreenshotErrorCode

Las enumeraciones ScreenshotError de códigos disponibles en el módulo DCV

- `NO_CHANNEL`
- `GENERIC_ERROR`

Tipo:

- número

serverInfo

Tipo:

- Objeto

Propiedades:

Nombre	Tipo	Descripción
`name`	cadena	Nombre del software.
`version`	Objeto	Número de versión del software. Nombre: `major` — Tipo: número — Descripción: Número de la versión principal. Nombre: `minor` — Tipo: número — Descripción: Número de la versión secundaria. Nombre: `revisic` — Tipo: número — Descripción: Número de la versión

Nombre	Tipo	Descripción
		Nombre: Tipo: Descripción: de revisión.
`os`	cadena	Sistema operativo.
`arch`	cadena	Arquitectura.
`hostname`	cadena	Nombre del host.

stats

Tipo:

- Objeto

Propiedades:

Nombre	Tipo	Descripción
`fps`	número	Fotogramas actuales por segundo.
`traffic`	número	Tráfico actual en bit/s.
`peakTraffic`	número	Pico de tráfico en bit/s desde que se estableció la conexión.
`latency`	número	Latencia actual en ms.
`currentChannels`	número	Número de canales que se han abierto desde que se estableció la conexión.

Nombre	Tipo	Descripción
`openedChannels`	número	Número de canales abiertos actualmente.
`channelErrors`	número	Número de canales que han informado de un error.

storeFile(file, dir)

Parámetros:

Nombre	Tipo	Descripción
`file`	Archivos	El objeto de archivo (para obtener más información, consulte https://developer.mozilla). org/en-US/docs/Web/API/File) que queremos subir al servidor.
`dir`	cadena	Ruta absoluta en el servidor en el que queremos cargar el archivo.

TimezoneRedirectionErrorCode

Las enumeraciones de TimezoneRedirectionError códigos disponibles en el módulo DCV

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

Tipo:

- número

TimezoneRedirectionSettingCode

Las enumeraciones TimezoneRedirectionSetting de códigos disponibles en el módulo DCV

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

Tipo:

- número

TimezoneRedirectionStatusCode

Las enumeraciones TimezoneRedirectionStatus de códigos disponibles en el módulo DCV

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

Tipo:

- número

WebcamErrorCode

Las enumeraciones WebcamError de códigos disponibles en el módulo DCV

- `SETTING_WEBCAM_FAILED`
- `CHANNEL_NOT_AVAILABLE`

Tipo:

- número

Clase de conexión

Clase de conexión obtenida al llamar al método `connect` del módulo `dcv`. Para ver un ejemplo de cómo utilizarlo, consulte la sección Introducción.

Expone

- Métodos

Métodos

Enumeración

- attachDisplay (win, DisplayConf) → {Promise. <number>|Promesa. < {código: MultiMonitorErrorCode, mensaje: cadena} >}
- captureClipboardEvents(activado, win, displayID) → {void}
- detachDisplay(displayId) → {void}
- disconnect() → {void}
- disconnectCollaborator(connectionId) → {void}
- enableDisplayQualityActualizaciones (activar) → {void}
- enableHighPixelDensidad (activar) → {void}
- enableTimezoneRedirection(habilitar) → {Promise|Promesa. < {código: TimezoneRedirectionErrorCode, mensaje: cadena} >}
- enterRelativeMouseModo () → {vacío}
- getConnectedDevices() → {Promesa. <Matriz. < MediaDeviceInfo >>|Promesa. < {message: string} >}
- getFileExplorer() → {Promesa. < almacenamiento de archivos >|Promesa. < {código: ChannelErrorCode, mensaje: cadena} >}
- getServerInfo() → {serverInfo}
- getScreenshot () → {Promise|Promesa. < {código: ScreenshotErrorCode, mensaje: cadena} >}
- getStats() → {stats}
- latchModifierKey(clave, ubicación, isDown) → {boolean}
- openChannel (nombre, token de autenticación, devolución de llamadas, espacio de nombres) → {Promise|Promise. < {código:, mensaje: cadena} >} ChannelErrorCode

- queryFeature(featureName) → {Promise.<{enabled: boolean, remote?: string, autoCopy?: boolean, autoPaste?: boolean, serviceStatus?: string, available?: boolean}>|Promise.<{message: string}>}
- registerKeyboardShortcuts(atajos) → {vacío}
- requestDisplayConfig(highColorAccuracy) → {Promesión|Promesa. < {código: DisplayConfigErrorCode, mensaje: cadena} >}
- requestDisplayLayout(diseño) → {Promise|Promesa. < {código: ResolutionErrorCode, mensaje: cadena} >}
- requestResolution (ancho, alto) → {Promise|Promesa. < {código: ResolutionErrorCode, mensaje: cadena} >}
- sendKeyboardEvent(evento) → {boolean}
- sendKeyboardShortcut(acceso directo) → {vacío}
- setDisplayQuality(min, maxopt) → {vacío}
- setDisplayScale(scaleRatio, displayID) → {Promise|Promesa. < {código: ResolutionErrorCode, mensaje: cadena} >} (OBSOLETO)
- setKeyboardQuirks(peculiaridades) → {vacío}
- setMaxDisplayResolución (maxWidth, maxHeight) → {void}
- SetMicrophone (activar) → {Promise|Promise. < {código: AudioErrorCode, mensaje: cadena} >}
- setMinDisplayResolución (minWidth, minHeight) → {void}
- setUploadBandwidth(valor) → {número}
- setVolume(volume) → {void}
- setMicrophone (habilitar, DeviceID) → {Promise|Promise. < {código:, mensaje: AudioErrorCodecadena} >}
- setWebcam (habilitar, DeviceID) → {Promise|Promise. < {código:, mensaje: WebcamErrorCodecadena} >}
- syncClipboards() → {boolean}

attachDisplay (win, DisplayConf) → {Promise. <number>|Promesa. < {código: MultiMonitorErrorCode, mensaje: cadena} >}

Asocia una pantalla específica a una ventana. No se puede asociar la pantalla principal. Si se realiza correctamente, la función devuelve el `displayId`.

Parámetros:

Nombre	Tipo	Descripción
`win`	Objeto	Ventana a la que se debe asociar la pantalla.
`displayConf`	Objeto	Configuración de la pantalla. Nombr Tipo Atribut Descripción displ número <option> ID de la pantalla. displ vName Nombre de div. de pantalla.

Devuelve:

Promesa. Si se rechaza, la promesa devuelve un objeto de error.

Tipo

Promesa. <number>| Promesa. < {código: MultiMonitorErrorCode, mensaje: cadena} >

captureClipboardEvents(activado, win, displayID) → {void}

Inicia o detiene la escucha de eventos de copiar y pegar. En el caso de los portapapeles interactivos (siempre en el caso de pegar), debemos empezar a escuchar los eventos de copiar y pegar. Podría ser útil iniciar y detener la escucha solo cuando sea necesario, por ejemplo, cuando se muestre un modal.

Parámetros:

Nombre	Tipo	Atributos	Descripción
`enabled`	booleano		Para empezar a escuchar eventos, especifique `true`. Para empezar a escuchar eventos, especifique `false`.
`win`	Objeto	<optional>	Ventana en la que se van a escuchar eventos. Si se omite, se utiliza la ventana predeterminada.
`displayId`	número	<optional>	ID de la pantalla que debe escuchar los eventos. Si se omite, se utiliza la pantalla predeterminada de la ventana.

Devuelve:

Tipo

void

detachDisplay(displayId) → {void}

Desasocia una pantalla específica. La pantalla principal no se puede desasociar.

Parámetros:

Nombre	Tipo	Descripción
`displayId`	número	ID de la pantalla que se va a desasociar.

Devuelve:

Tipo

void

disconnect() → {void}

Se desconecta del servidor Amazon DCV y cierra la conexión.

Devuelve:

Tipo

void

disconnectCollaborator(connectionId) → {void}

Solicita la desconexión del colaborador conectado con el identificador de conexión proporcionado (desde Amazon DCV Web Client SDK versión 1.1.0).

Parámetros:

Nombre	Tipo	Descripción
`connectionId`	booleano	Identificador de la conexión que se desconectará.

Devuelve:

Tipo

void

enableDisplayQualityActualizaciones (activar) → {void}

Habilita o deshabilita las actualizaciones de calidad de pantalla para las áreas de streaming que no reciben actualizaciones. Al deshabilitar las actualizaciones de calidad de pantalla se reduce el uso de ancho de banda, pero también la calidad de pantalla.

Parámetros:

Nombre	Tipo	Descripción
`enable`	booleano	Para habilitar las actualizaciones de la calidad de pantalla, especifique `true`. Para deshabilitar las actualizaciones de la calidad de pantalla, especifique `false`.

Devuelve:

Tipo

void

enableHighPixelDensidad (activar) → {void}

Habilita o deshabilita la alta densidad de píxeles en el cliente.

Parámetros:

Nombre	Tipo	Descripción
`enable`	booleano	Si se debe habilitar o no la alta densidad de píxeles.

Devuelve:

Tipo

void

enableTimezoneRedirection(habilitar) → {Promise|Promesa. < {código: TimezoneRedirectionErrorCode, mensaje: cadena} >}

Habilita o deshabilita el redireccionamiento de zona horaria. Si se habilita, el cliente solicita al servidor que ajuste la zona horaria del escritorio del servidor coincida a la zona horaria del cliente.

Parámetros:

Nombre	Tipo	Descripción
`enable`	booleano	Para habilitar el redirecci onamiento de zona horaria, especifique `true`. Para deshabilitar el redirecci onamiento de zona horaria, especifique `false`.

Devuelve:

Promesa. Si se rechaza, la promesa devuelve un objeto de error.

Tipo

Promesa. <number>| Promesa. < {código: TimezoneRedirectionErrorCode, mensaje: cadena} >

enterRelativeMouseModo () → {vacío}

Habilita el modo de ratón relativo.

Devuelve:

Tipo

void

getConnectedDevices() → {Promesa. <Matriz. < MediaDeviceInfo >>|Promesa. < {message: string} >}

Solicita una lista de los dispositivos multimedia conectados al ordenador cliente.

Devuelve:

Si tiene éxito, devuelve una promesa que se convierte en una matriz de MediaDeviceInfo objetos. Para obtener más información, consulte https://developer.mozilla. org/en-US/docs/Web/API/MediaDeviceInfo. Si se rechaza, la promesa devuelve un objeto de error.

Tipo

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

getFileExplorer() → {Promesa. < almacenamiento de archivos >|Promesa. < {código: ChannelErrorCode, mensaje: cadena} >}

Obtiene un objeto para administrar el almacenamiento de archivos del servidor Amazon DCV.

Devuelve:

Promesa. Se resuelve en el objeto del explorador de archivos si se cumple o en un objeto de error si se rechaza.

Tipo

Promesa. < filestorage > | Promesa. < {código: ChannelErrorCode, mensaje: cadena} >

getServerInfo() → {serverInfo}

Obtiene información acerca del servidor Amazon DCV.

Devuelve:

Información sobre el software del servidor.

Tipo

serverInfo

getScreenshot () → {Promise|Promesa. < {código: ScreenshotErrorCode, mensaje: cadena} >}

Recupera la captura de pantalla del escritorio remoto en formato PNG. La captura de pantalla se devolverá en el observador screenshotCallback. `null` se devolverá en su lugar en caso de error.

Devuelve:

Promesa que se resuelve si se procesa la solicitud. Si se rechaza, recibimos un objeto de error.

Tipo

Promesa | Promesa. < {código: ScreenshotErrorCode, mensaje: cadena} >

getStats() → {stats}

Obtiene estadísticas sobre el servidor Amazon DCV.

Devuelve:

Información sobre las estadísticas de streaming.

Tipo

stats

latchModifierKey(clave, ubicación, isDown) → {boolean}

Envía un solo evento `keydown` o `keyup` de teclado para un modificador permitido.

Parámetros:

Nombre	Tipo	Descripción
`key`	Control | Alt | | Meta AltGraph | OS | Shift	Tecla que se va a enviar.
`location`	KeyboardEvent.ubicación	Nombre de la tecla. Para obtener más información, consulte https://developer. mozilla. org/en-US/docs/Web/API/KeyboardEvent/location.
`isDown`	booleano	Si el evento de tecla que se va a inyectar es keydown (`true`) o keyup (`false`).

Devuelve:

Si la combinación solicitada es válida, la función devuelve `true`; de lo contrario, devuelve `false`.

Tipo

booleano

openChannel (nombre, token de autenticación, devolución de llamadas, espacio de nombres) → {Promise|Promise. < {código:, mensaje: cadena} >} ChannelErrorCode

Abre un canal de datos personalizado en la conexión si se creó en el servidor Amazon DCV.

Parámetros:

Nombre	Tipo	Descripción
`name`	cadena	El nombre del canal.
`authToken`	cadena	Token de autenticación que se utilizará al conectarse al canal.

Nombre	Tipo	Descripción
`callbacks`	Objeto	Funciones de devolución de llamada onMessage y onClose que se invocan.
`namespace`	cadena	Espacio de nombres del canal. Disponible desde Amazon DCV Web Client SDK 1.2.0 y Amazon DCV Server 2022.1.

Devuelve:

Promesa. Si se rechaza, recibimos un objeto de error.

Tipo

Promesa | Promesa. < {código: ChannelErrorCode, mensaje: cadena} >

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

Consulta el estado de una característica específica del servidor Amazon DCV.

Parámetros:

Nombre	Tipo	Descripción
`featureName`	feature	Nombre de la característica que se va a consultar.

Devuelve:

Promesa. Si se resuelve, la función devuelve un objeto `status` que siempre contiene una propiedad `enabled` y, posiblemente, también otras propiedades. Si se rechaza, la función devuelve un objeto `error`.

Tipo

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

registerKeyboardShortcuts(atajos) → {vacío}

Registra los métodos abreviados de teclado.

Parámetros:

<table>
<tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td><code>shortcuts</code></td><td>Array.<Object></td><td>Matriz de teclas y asignaciones que se van a registrar.
<table>
<tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td><code>sequenc</code></td><td>Array.<Object></td><td>Método abreviado de teclado que se va a registrar.
<table>
<tr><th>N</th><th>T</th><th>Descripción</th></tr>
<tr><td>k</td><td>Kve</td><td>Valor de la tecla presionada</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

Nombre	Tipo	Descripción
		Nombre: Tipo: Descripción N T Descripción por el usuario. Para obtener más información, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key. l k v a Matriz de ubicaciones en las que

Nombre	Tipo	Descripción
		Nombre: Tipo: Descripción N T Descripción se van a enviar. Ubicación de la tecla en el teclado. Para obtener más información, consulte https://developer.mozilla.org/en-US/docs/Web/API/

<table>
<tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Nombre</th><th>Tipo</th><th>Descripción</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>N</th><th>T</th><th>Descripción</th></tr>
<tr><td></td><td></td><td>KeyboardEvent/location.</td></tr>
</table>
</td></tr>
<tr><td><code>output</code></td><td>Array.<Object></td><td>La acción que se pretende realizar con el acceso directo.
<table>
<tr><th>N</th><th>T</th><th>Descripción</th></tr>
<tr><td>k</td><td>K v</td><td>Valor de la tecla presionada por el usuario. Para obtener</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

Nombre	Tipo	Descripción
		Nombre: Tipo: Descripción N T Descripción más información, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key. l K Matriz de ubicación en las que se van a enviar. Ubicación

Nombre	Tipo	Descripción
		Nombre: Tipo: Descripción: Nombre: Tipo: Descripción: N T Descripción de la tecla en el teclado. Para obtener más información, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/

Nombre	Tipo	Descripción
		Nombre Tipo Descripción N T Descripción location.

Devuelve:

Tipo

void

requestDisplayConfig(highColorAccuracy) → {Promesa|Promesa. < {código: DisplayConfigErrorCode, mensaje: cadena} >}

Solicita una configuración de pantalla actualizada al servidor Amazon DCV. Disponible desde Amazon DCV Web Client SDK 1.1.0 y Amazon DCV Server 2022.0.

Parámetros:

Nombre	Tipo	Descripción
`highColorAccuracy`	booleano	Si se debe solicitar alta precisión de color.

Devuelve:

Promesa. Si se rechaza, la promesa devuelve un objeto de error.

Tipo

Promesa | Promesa. < {código: DisplayConfigErrorCode, mensaje: cadena} >

requestDisplayLayout(diseño) → {Promise|Promesa. < {código: ResolutionErrorCode, mensaje: cadena} >}

Solicita un diseño de pantalla actualizado para la conexión.

Parámetros:

Nombre	Tipo	Descripción
`layout`	Array.<Monitor>	Pantallas solicitadas en el diseño.

Devuelve:

Promesa. Si se rechaza, recibimos un objeto de error.

Tipo

Promesa | Promesa. < {código: ResolutionErrorCode, mensaje: cadena} >

requestResolution (ancho, alto) → {Promise|Promesa. < {código: ResolutionErrorCode, mensaje: cadena} >}

Solicita una resolución de pantalla actualizada al servidor Amazon DCV.

Parámetros:

Nombre	Tipo	Descripción
`width`	número	Ancho que se va a solicitar en píxeles. El valor mínimo permitido es `0`.
`height`	número	Ancho que se va a solicitar en píxeles. El valor mínimo permitido es `0`.

Devuelve:

Promesa. Si se rechaza, la promesa devuelve un objeto de error.

Tipo

Promesa | Promesa. < {código: ResolutionErrorCode, mensaje: cadena} >

sendKeyboardEvent(evento) → {boolean}

Envía un evento de método abreviado de teclado. Para obtener más información sobre los eventos del teclado, consulte https://developer.mozilla. org/en-US/docs/Web/API/KeyboardEvent. Los eventos de teclado válidos incluyen: `keydown`, `keypress` y `keyup`. Para obtener más información sobre estos eventos, consulte https://developer.mozilla. org/en-US/docs/Web/API/KeyboardEvent#events.

Parámetros:

Nombre	Tipo	Descripción
`event`	KeyboardEvent	Evento de teclado que se va a enviar.

Devuelve:

Si el evento no es válido, la función devuelve `false`. Si el evento es válido, la función devuelve `true`.

Tipo

booleano

sendKeyboardShortcut(acceso directo) → {vacío}

Envía un método abreviado de teclado. Utilice esta función para enviar una secuencia de `keydown` o `keyup` completa. Por ejemplo, si se envía Ctrl + Alt + Supr, se envían los eventos `keydown` para todas las teclas seguidos de los eventos `keyup`. Utilice esta función incluso si desea enviar una sola tecla.

Parámetros:

Nombre	Tipo	Descripción
`shortcut`	Array.<Object>	Matriz de teclas que se van a enviar. Nombre: `key` Tipo: KeyboardEvent.clave Descripción: Valor de la tecla presionada por el usuario. Para obtener más información, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key.

Nombre	Tipo	Descripción

Nombre	Tipo	Descripción
`locatic`	Keyboarc vent.ubic ación	Matriz de teclas que se van a enviar. Ubicación de la tecla en el teclado. Para obtener más informaci ón, consulte https:// d eveloper. mozilla. org/ en-US /docs/ Web /API/ Keyb oardEvent / location .

Devuelve:

Tipo

void

setDisplayQuality(min, maxopt) → {vacío}

Establece la calidad de imagen que se utilizará en la conexión. El intervalo válido es de `0` a `100`, siendo 1 la calidad de imagen más baja y `100` la calidad de imagen más alta. Especifique `0` para retener el valor actual.

Parámetros:

Nombre	Tipo	Atributos	Descripción
`min`	número		Calidad de imagen mínima.
`max`	número	<optional>	Calidad de imagen máxima.

Devuelve:

Tipo

void

setDisplayScale(scaleRatio, displayID) → {Promise|Promesa. < {código: ResolutionErrorCode, mensaje: cadena} >} (OBSOLETO)

Obsoleto a partir de la versión 1.3.0. Ya no es necesario configurar la escala de pantalla. Las coordenadas del ratón se gestionarán automáticamente de forma interna.

Notifica a Amazon DCV que la pantalla está escalada en el lado del cliente. Utilice esta opción para notificar al servidor que necesita escalar los eventos del ratón para que coincidan con la relación de pantalla del cliente.

Parámetros:

Nombre	Tipo	Descripción
`scaleRatio`	float	Relación de escala que se va a utilizar. Debe ser un número estrictamente positivo.
`displayId`	número	ID de la pantalla que se va a escalar.

Devuelve:

Promesa. Si se rechaza, la promesa devuelve un objeto de error.

Tipo

Promesa | Promesa. **<** {código: ResolutionErrorCode, mensaje: cadena} **>**

setKeyboardQuirks(peculiaridades) → {vacío}

Establece las peculiaridades del teclado del ordenador cliente.

Parámetros:

Nombre	Tipo	Descripción
`quirks`	Objeto	Peculiaridades del teclado que se habilitarán o deshabilitarán. Nombre: `macOptiToAlt` — Tipo: boolean — Descripción: Para asignar la tecla Opción a Alt

Nombre	Tipo	Descripción		
		Nombre	Tipo	Descripción
				para macOS, especifique `true`. De lo contrario, especifique `false`.
		`macCommandToCtrl`	boolean	Para asignar la tecla Comando a Ctrl para macOS, especifique `true`. De lo contrario, especifique `false`.

Devuelve:

Tipo

void

setMaxDisplayResolución (maxWidth, maxHeight) → {void}

Establece la resolución máxima de pantalla que se utilizará en la conexión.

Parámetros:

Nombre	Tipo	Descripción
`maxWidth`	número	Ancho máximo de pantalla en píxeles. El valor mínimo permitido es `0`.
`maxHeight`	número	Alto máximo de pantalla en píxeles. El valor mínimo permitido es `0`.

Devuelve:

Tipo

void

SetMicrophone (activar) → {Promise|Promise. < {código: AudioErrorCode, mensaje: cadena} >}

Habilita o deshabilita el micrófono.

Parámetros:

Nombre	Tipo	Descripción
`enable`	booleano	Para habilitar el micrófono , especifique `true`. Para

Nombre	Tipo	Descripción
		deshabilitar el micrófono, especifique `false`.

Devuelve:

Promesa. Si se rechaza, la promesa devuelve un objeto de error.

Tipo

Promesa | Promesa. **<** {código: AudioErrorCode, mensaje: cadena} **>**

setMinDisplayResolución (minWidth, minHeight) → {void}

Establece la resolución mínima de pantalla que se utilizará en la conexión. Es posible que algunas aplicaciones requieran una resolución mínima de pantalla. Si la resolución mínima requerida es mayor que la resolución máxima admitida por el cliente, se utiliza una estrategia de cambio de tamaño. Utilice esta función con cuidado. La estrategia de cambio de tamaño podría provocar que el ratón y el sistema de entrada táctil fueran menos precisos.

Parámetros:

Nombre	Tipo	Descripción
`minWidth`	número	Ancho mínimo de pantalla en píxeles. El valor mínimo permitido es `0`.
`minHeight`	número	Alto mínimo de pantalla en píxeles. El valor mínimo permitido es `0`.

Devuelve:

Tipo

void

setUploadBandwidth(valor) → {número}

Establece el ancho de banda máximo que se utilizará para cargar archivos en el servidor Amazon DCV.

Parámetros:

Nombre	Tipo	Descripción
`value`	número	Ancho de banda de subida máximo en Kbps. El intervalo válido es de 1024 kbps a 102400 kbps.

Devuelve:

- Límite de ancho de banda establecido. `null` si la característica de almacenamiento de archivos está deshabilitada en el servidor.

Tipo

número

setVolume(volume) → {void}

Establece el nivel de volumen que se utilizará en el audio. El intervalo válido es de 0 a 100, donde 0 es el volumen más bajo y 100 el más alto.

Parámetros:

Nombre	Tipo	Descripción
`volume`	número	Nivel de volumen que utilizará.

Devuelve:

Tipo

void

setMicrophone (habilitar, DeviceID) → {Promise|Promise. < {código:, mensaje: AudioErrorCodecadena} >}

[Experimental: podría cambiar en el futuro] Habilita o deshabilita el micrófono.

Parámetros:

Nombre	Tipo	Descripción
`enable`	booleano	Para habilitar el micrófono , especifique `true`. Para deshabilitar el micrófono, especifique `false`.
`deviceId`	cadena	El ID de dispositivo del micrófono. Si no se proporcio na ningún deviceId, se utiliza el deviceId `default`.

Devuelve:

Promesa. Si se rechaza, la promesa devuelve un objeto de error.

Tipo

Promesa | Promesa. < {código: AudioErrorCode, mensaje: cadena} >

setWebcam (habilitar, DeviceID) → {Promise|Promise. < {código:, mensaje: WebcamErrorCodecadena} >}

Habilita o deshabilita la cámara web.

Parámetros:

Nombre	Tipo	Descripción
`enable`	booleano	Para habilitar la cámara web, especifique `true`. Para

Nombre	Tipo	Descripción
		deshabilitar la cámara web, especifique `false`.
`deviceId`	cadena	ID de dispositivo de la cámara web.

Devuelve:

Promesa que, si es correcta, se resuelve en el deviceId de la cámara web asociada/desasociada. Si se rechaza, la promesa devuelve un objeto de error.

Tipo

Promesa. <string>| Promesa. < {código: WebcamErrorCode, mensaje: cadena} >

syncClipboards() → {boolean}

Sincroniza el portapapeles del cliente local con el portapapeles del servidor Amazon DCV remoto. El navegador debe admitir la función de copia automática.

Devuelve:

Si los portapapeles se han sincronizado, la función devuelve `true`. Si los portapapeles no se han sincronizado o si el navegador no admite copia automática, la función devuelve `false`.

Tipo

booleano

Clase de autenticación

La clase de autenticación debe utilizarse para obtener un token de autenticación llamando al método `authenticate` del módulo `dcv`. Para ver un ejemplo de cómo utilizarlo, consulte la sección Introducción.

Expone

- Métodos

Métodos

Enumeración

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

retry() → {void}

Vuelve a intentar el proceso de autenticación.

Devuelve:

Tipo

void

sendCredentials(credentials) → {void}

Envía las credenciales de autenticación proporcionadas por el cliente al servidor Amazon DCV.

Parámetros:

Nombre	Tipo	Descripción
`credentials`	Objeto	Objeto que contiene las credenciales suministradas. Las credenciales deben tener el mismo nombre y ser del mismo tipo que las especific ados en el desafío.

Devuelve:

Tipo

void

Clase de recurso

La clase de recurso puede recuperar o descartar el archivo correspondiente que se acaba de imprimir o descargar. Al realizar estas acciones, las funciones de observador correspondiente `filePrinted` y `fileDownload` se invocarían, respectivamente, con el objeto de recurso como único argumento. Este recurso se puede aceptar o rechazar para recuperar o descartar el archivo al que hace referencia.

Expone

- Métodos

Métodos

Enumeración

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

accept(urlParameters) → {void}

Descarga el recurso de forma local.

Parámetros:

Nombre	Tipo	Descripción
`urlParameters`	Objeto	Objeto opcional que contiene los pares de clave/valor de los parámetros de búsqueda de URL transferidos a la solicitud para recuperar el recurso.

Devuelve:

Tipo

void

decline() → {void}

Descarta el recurso.

Devuelve:

Tipo

void

Amazon DCV Web UI SDK

Una biblioteca de componentes de JavaScript React, que actualmente exporta un único componente de React llamado `DCVViewer` que se conecta al servidor Amazon DCV y renderiza la barra de herramientas para interactuar con la transmisión remota.

Expone

- Componentes

Componentes

Enumeración

- DCVViewer

DCVViewer

Componente React que hace que la barra de herramientas con todas sus funcionalidades sea útil para interactuar con el flujo remoto.

Propiedades:

Enumeración

- dcv
- uiConfig

dcv

Nombre	Tipo	Obligatorio	Descripción
`dcv`	Objeto	Sí	Objeto que define las propiedades necesarias para establecer la conexión con el servidor Amazon DCV, estableciendo el nivel de registro y la URL desde donde se cargan los activos del Amazon DCV Web Client SDK y se accede a los recursos de DCV.

Nom	Tipo	Obli io	Descripción
`ses`	Cad	Sí	ID de la sesión de Amazon DCV.
`aut`	Cad	Sí	Token de autenticación que se utilizará

Nombre	Tipo	Obligatorio	Descripción
			Nom Tipo Obli io Descripci ón al conectars e al servidor. ser Cad Sí Nombre y puerto de host del servidor Amazon DCV en ejecución con el siguiente formato: https:// d cv_host_a ddress:po rt. Por ejemplo: https: //:8443 my- dcv-

Nombre	Tipo	Obligatorio	Descripción
			Nom Tipo Obli io Descripci ón se rver bas Cad Sí URL absoluta o relativa desde la que se cargarán los archivos del SDK. res ase Cad No (valc pred inad "") URL absoluta o relativa desde la que se accederá a los recursos de DCV.

Nombre	Tipo	Obligatorio	Descripción

Nom	Tipo	Obligatorio	Descripción
`onDisconnect`	func	No (valor predeterminado `() => {}`)	Función de devolución de llamada que se invoca cuando se desconecta del servidor Amazon DCV y se cierra la conexión.
`log`	LogL	No (valor predeterminado `Log.INF`	Nivel de registro que utilizará en el visor.

uiConfig

Nombre	Tipo	Obligatorio	Descripción
`uiConfig`	Objeto	No (valor predeterminado: `{}`)	Objeto que define las propiedades para configurar si la barra de herramientas es visible y si se muestran en ella los botones de pantalla completa y de varios monitores. Nom / Tipo / Obligatorio / Descripción `too` / Obje / No (valor predeterminado: `{}`) / Objeto que define las opciones de configuración de la barra de herramientas.

Nombre	Tipo	Obligatorio	Descripción
			Nom Tipo Obligatorio Descripción [illegible] ión [illegible] (valor predeterminado: true) se mostrará u ocultará la barra de herramientas. [illegible] (valor predeterminado: true) se mostrará u ocultará el botón de pantalla completa en la

Nombre	Tipo	Obligatorio	Descripción
			Nom Tipo Obli io Descripci ón
			barra de herramie tas.
			mostrará u ocultará el botón de varios monitore en la barra de herramie tas.

Notas de la versión e historial de revisión del Amazon DCV Web Client SDK

Esta página recoge las notas de la versión y el historial de revisión del Amazon DCV Web Client SDK.

Temas

- Notas de la versión del Amazon DCV Web Client SDK
- Historial de documentos

Notas de la versión del Amazon DCV Web Client SDK

En esta sección se proporcionan las notas de la versión del Amazon DCV Web Client SDK por fecha de publicación.

Temas

- 1.8.7 — 31 de octubre de 2024
- 1.8.4: 1 de octubre de 2024
- 1.5.10: 19 de diciembre de 2023
- 1.5.6: 9 de noviembre de 2023
- 1.4.4: 29 de junio de 2023
- 1.4.0: 28 de marzo de 2023
- 1.3.1: 9 de diciembre de 2022
- 1.3.0: 11 de noviembre de 2022
- 1.2.1 : 21 de julio de 2022
- 1.2.0: 29 de junio de 2022
- 1.1.3: 23 de mayo de 2022
- 1.1.2: 19 de mayo de 2022
- 1.1.1: 23 de marzo de 2022
- 1.1.0: 23 de febrero de 2022
- 1.0.4: 20 de diciembre de 2021

- 1.0.3: 1 de septiembre de 2021
- 1.0.2: 30 de julio de 2021
- 1.0.1: 31 de mayo de 2021
- 1.0.0: 24 de marzo de 2021

1.8.7 — 31 de octubre de 2024

Números de compilación	Cambios y correcciones de errores	
• Versión semántica: 1.8.7 • Versión: 858	• Se corrigió el renderizado en Firefox 130 y versiones posteriores	

1.8.4: 1 de octubre de 2024

Números de compilación	Nuevas características	Cambios y correcciones de errores
• Versión semántic: 1.8.4 • Compilación: 840	Se han añadido las siguientes características: • Se ha cambiado el nombre a “Amazon DCV Web Client SDK” • Se agregó una nueva enableHighPixel densidad de API para pantallas de alta resolución	• Se ha mejorado el control de cámaras web • Se ha mejorado el control de la reproducción de audio • WebCodecs Manejo mejorado •

Números de compilación	Nuevas características	Cambios y correcciones de errores
	• Se ha agregado una API experimental setMicrophone para seleccionar el micrófono en los navegadores compatibles • Se han agregado nuevos errores de conexión GATEWAY_BUSY, UNSUPPORTED_CREDENTIAL y TRANSPORT_ERROR • Se han agregado nuevos motivos de cierre: EXTERNAL_PROTOCOL_CONNECTION_EVICTED y DISCONNECTION_REQUESTED	Se ha mejorado la función de conectar y desconectar el micrófono y la cámara web • Se ha mejorado el arrastre remoto de ventanas cuando se utilizan varios monitores • Los permisos de carga y descarga en el almacenamiento de archivos ahora se propagan correctamente • Pequeñas correcciones en el renderizado

1.5.10: 19 de diciembre de 2023

Versión	Notas de la versión
• Versión semántica: 1.5.10 • Compilación: 684	Cambios y correcciones de errores • Corrección de los errores de decodificación de transmisiones

1.5.6: 9 de noviembre de 2023

Versión	Notas de la versión

Versión	Notas de la versión
• Versión semántica: 1.5.6 • Compilación: 659	Cambios y correcciones de errores • Mejoras en el rendimiento de descodificación y representación de secuencias • Se ha eliminado la compatibilidad con Internet Explorer 11

1.4.4: 29 de junio de 2023

Versión	Notas de la versión
• Versión semántica: 1.4.4 • Compilación: 573	Cambios y correcciones de errores • El componente de interfaz de usuario del visor utiliza ahora la API `navigator.keyboard.lock` en los navegador es que lo admiten para gestionar claves especiales en pantalla completa. • Se ha corregido un problema que podía provocar colores incorrectos al utilizar Chrome 114 o versiones posteriores. • WebCodecs Detección mejorada. • Se ha corregido un problema con el estado del botón del ratón al entrar en la ventana. • Se ha corregido un problema que podía provocar que las teclas modificadoras permanecieran pulsadas en macOS. •

Versión	Notas de la versión
	• Se ha mejorado la robustez de audio en condiciones de red degradadas. • Se han corregido fugas de memoria. • Se han mejorado los registros para incluir hora y nivel.

1.4.0: 28 de marzo de 2023

Versión	Notas de la versión
• Versión semántica: 1.4.0 • Compilación: 476	Nuevas características • Se ha agregado un nuevo método `uploadFiles` al objeto `FileStorage` para cargar varios archivos. • El componente de interfaz de usuario del visor admite ahora arrastrar y soltar para iniciar la carga de archivos. • La API WebCodecs del navegador ahora también se usa para audio y cámara web. Cambios y correcciones de errores • Se han corregido fugas de memoria relacionadas con conexiones repetidas desde la misma página. •

Versión	Notas de la versión
	`setUploadBandwidth` permite ahora valores de hasta 1 Gbps. • Se ha optimizado la representación de los componentes de la interfaz de usuario. • Se ha corregido la compatibilidad con cursores animados en Windows. • Se ha corregido un problema de compatibi lidad con el portapapeles cuando había datos de texto e imagen para la misma operación. • Se ha mejorado la solidez de la API de cámara web: no se puede cambiar la configuración mientras haya una solicitud en curso. Ahora `webcam.setEnabled` realiza un seguimiento del identificador del dispositivo cuando la solicitud está en curso y devuelve una promesa. El componente de interfaz de usuario del visor muestra una notificación en caso de error.

1.3.1: 9 de diciembre de 2022

Versión	Notas de la versión
• Versión semántica: 1.3.1 • Compilación: 413	Cambios y correcciones de errores • Se ha corregido un problema que podía provocar que la interfaz de usuario de redireccionamiento de zona horaria perdiera la sincronización con el servidor.

Versión	Notas de la versión
	• Se ha corregido una fuga de memoria tras varias reconexiones. • Se ha corregido un problema que generaba una página en blanco al desconectarse. • Se ha corregido un error que provocaba que se cerraran las advertencias de la consola en el decodificador de audio.

1.3.0: 11 de noviembre de 2022

Versión	Notas de la versión
• Versión semántica: 1.3.0 • Compilación: 407	Nuevas características • Adoptó Cloudscape (ponente)https://cloudscape.design) for the UI Viewer com. • Se ha agregado compatibilidad con el redireccionamiento de zona horaria. Cambios y correcciones de errores • Se ha corregido la falta de actualización en el portapapeles asíncrono cuando el visor DCV está enfocado. • La función `setDisplayScale` ya no es necesaria para escalar la pantalla del lado del cliente.

Versión	Notas de la versión
	• El componente `DCVViewer` llama ahora automáticamente a `disconnect()` cuando está desmontado.

1.2.1 : 21 de julio de 2022

Versión	Notas de la versión
• Versión semántica: 1.2.1 • Compilación: 358	Cambios y correcciones de errores • Se ha corregido un problema que provocaba un error en la conexión con el servidor Amazon DCV 2019.1 y versiones anteriores.

1.2.0: 29 de junio de 2022

Versión	Notas de la versión
• Versión semántica: 1.2.0 • Compilación: 352	Cambios y correcciones de errores • Se ha corregido un error de bloqueo que se producía cuando los fotogramas recibidos superaban la resolución máxima admitida (4096 x 2160). • Los objetos de recurso (pasados como argumentos observadores `fileDownload` y `filePrinted`) tienen ahora los métodos `accept` y `decline` a los que se

Versión	Notas de la versión
	puede llamar en el objeto para descargar y descartar el recurso respectivamente. • Se ha corregido un error menor en la sincronización automática del portapapeles al desconectarse.

1.1.3: 23 de mayo de 2022

Versión	Notas de la versión
• Versión semántica: 1.1.3 • Compilación: 329	Cambios y correcciones de errores • Se ha corregido un problema que impedía que la conexión se realizara correctamente al especificar la opción. web-url-path

1.1.2: 19 de mayo de 2022

Versión	Notas de la versión
• Versión semántica: 1.1.2 • Compilación: 322	Cambios y correcciones de errores • Se ha corregido un problema que podía provocar que la entrada no funcionara correctamente después de la conexión. • Se han corregido las coordenadas del ratón cuando la relación de escala es superior a 1.

1.1.1: 23 de marzo de 2022

Versión	Notas de la versión
• Versión semántica: 1.1.1 • Compilación: 309	Cambios y correcciones de errores • Se comunica `Transport Error` cuando se agota el tiempo de espera de la comunicación con el servidor. • Se ha corregido un error de descodificación recurrente al transmitir en streaming en alta resolución.

1.1.0: 23 de febrero de 2022

Versión	Notas de la versión
• Versión semántica: 1.1.0 • Compilación: 295	Nuevas características • Publicación de la biblioteca Amazon DCV Web UI SDK con el componente `DCVViewer` de React. • Exportación del Amazon DCV Web Client SDK como módulos UMD y ES. • Se ha agregado compatibilidad de alta precisión de color. • Se ha agregado la capacidad de enumerar los clientes conectados a una sesión y de interactuar con ellos. Se han agregado notificaciones de conexión y desconexión.

Versión	Notas de la versión
	Cambios y correcciones de errores • Se ha mejorado la compatibilidad de descodificación de webcodecs. • Diversas mejoras en el teclado. • Se ha corregido un error que impedía abrir una segunda pantalla cuando el portapapeles estaba desactivado.

1.0.4: 20 de diciembre de 2021

Versión	Notas de la versión
• Versión semántica: 1.0.4 • Compilación: 249	Nuevas características • Compatibilidad para abrir varias conexiones desde la misma página. • Compatibilidad para cargar el SDK desde una CDN.

1.0.3: 1 de septiembre de 2021

Versión	Notas de la versión
• Versión semántica: 1.0.3 • Compilación: 202	Nuevas características •

Versión	Notas de la versión
	Soporte experimental para WebCodecs . Está deshabilitada de forma predeterminada y debe habilitarse a través del objeto `ConnectionConfig` que utiliza la nueva propiedad `enableWebCodecs` . • Portapapeles: se ha agregado compatibilidad con el tipo de datos `image/png` en navegadores basados en Chromium. • Se ha agregado un observador/devolución de llamada para obtener la captura de pantalla del servidor como una imagen PNG (requiere servidor Amazon DCV 2021.2). Cambios y correcciones de errores • Se ha mejorado la gestión de los modificadores del teclado.

1.0.2: 30 de julio de 2021

Versión	Notas de la versión
• Versión semántica: 1.0.2 • Compilación: 167	• Detección de presión fija para eventos del lápiz óptico. • Se ha mejorado la compatibilidad con la distribución del teclado coreano en Chrome.

1.0.1: 31 de mayo de 2021

Versión	Notas de la versión
• Versión semántica: 1.0.1 • Compilación:141	• Se ha corregido la propagación de errores de conexión y motivos de cierre • Se ha corregido la actualización del progreso de fragmentos de almacenamiento de archivos • Se ha mejorado el control de cámaras web • Se ha mejorado el procesamiento de entrada de audio

1.0.0: 24 de marzo de 2021

Versión	Notas de la versión
• Versión semántica: 1.0.0 • Compilación: 81	Versión inicial del Amazon DCV Web Client SDK.

Historial de documentos

En la siguiente tabla se describe la documentación de esta versión del Amazon DCV Web Client SDK.

Cambio	Descripción	Fecha
Amazon DCV Web Client SDK versión 1.8.4	Ya está disponible Amazon DCV Web Client SDK 1.8.4. Para obtener más información, consulte SDK v.1.8.4.	1 de octubre de 2024
Amazon DCV Web Client SDK versión 1.5.6	Ya está disponible Amazon DCV Web Client SDK 1.5.6. Para obtener más información, consulte SDK v.1.5.6.	9 de noviembre de 2023
Amazon DCV Web Client SDK versión 1.4.4	Ya está disponible Amazon DCV Web Client SDK 1.4.4. Para obtener más información, consulte SDK v.1.4.4.	29 de junio de 2023
Amazon DCV Web Client SDK versión 1.4.0	Ya está disponible Amazon DCV Web Client SDK 1.4.0. Para obtener más información, consulte SDK v.1.4.0.	28 de marzo de 2023
Amazon DCV Web Client SDK versión 1.3.1	Ya está disponible Amazon DCV Web Client SDK 1.3.1. Para obtener más información, consulte SDK v.1.3.1.	9 de diciembre de 2022
Amazon DCV Web Client SDK versión 1.3.0	Ya está disponible Amazon DCV Web Client SDK 1.3.0. Para obtener más información, consulte SDK v.1.3.0.	11 de noviembre de 2022
Amazon DCV Web Client SDK versión 1.2.0	Ya está disponible Amazon DCV Web Client SDK 1.2.0. Para obtener más información, consulte SDK v.1.2.0.	29 de junio de 2022

Cambio	Descripción	Fecha
Amazon DCV Web Client SDK versión 1.1.0	Ya está disponible Amazon DCV Web Client SDK 1.1.0. Para obtener más información, consulte SDK v.1.1.0.	23 de febrero de 2022
Amazon DCV Web Client SDK versión 1.0.1	Se han corregido algunos errores tipográficos. Se han corregido errores menores, consulte SDK v1.0.1.	31 de mayo de 2021
Versión inicial	Primera publicación de este contenido.	24 de marzo de 2021

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.