Web 用戶端 SDK 開發人員指南

Amazon DCV

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

Amazon DCV: Web 用戶端 SDK 開發人員指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 Amazon DCV Web 用戶端 SDK？ ... 1
先決條件 ... 1
支援的功能 ... 2
瀏覽器支援 ... 2
版本控制慣例 ... 3
開始使用 ... 4
連線至 Amazon DCV 伺服器並取得第一個影格 ... 5
步驟 1：準備您的 HTML 頁面 ... 5
步驟 2：驗證、連線和取得第一個影格 ... 6
獎勵：自動建立 HTML 登入表單 ... 8
使用 Amazon DCV 功能 ... 10
了解 featuresUpdate 回呼函數 ... 10
處理功能更新 ... 10
使用 Amazon DCV Web UI SDK ... 11
先決條件 ... 11
步驟 1：準備您的 HTML 頁面 ... 12
步驟 2：驗證、連接和轉譯 `DCVViewer` React 元件。 ... 12
從 AWS-UI 更新至 Cloudscape 設計系統 ... 16
SDK 參考 ... 18
DCV 模組 ... 18
方法 ... 18
成員 ... 21
輸入和回呼定義 ... 25
連線類別 ... 60
方法 ... 18
身分驗證類別 ... 88
方法 ... 18
資源類別 ... 89
方法 ... 18
Amazon DCV Web UI SDK ... 90
元件 ... 90
版本備註和文件歷史記錄 ... 99
版本備註 ... 99
1.8.7 — 2024 年 10 月 31 日 ... 100

1.8.4 — 2024 年 10 月 1 日 100
1.5.10 — 2023 年 12 月 19 日 101
1.5.6 — 2023 年 11 月 9 日 101
1.4.4 — 2023 年 6 月 29 日 101
1.4.0 — 2023 年 3 月 28 日 102
1.3.1 — 2022 年 12 月 9 日 103
1.3.0 — 2022 年 11 月 11 日 104
1.2.1 — 2022 年 7 月 21 日 104
1.2.0 — 2022 年 6 月 29 日 105
1.1.3 — 2022 年 5 月 23 日 105
1.1.2 — 2022 年 5 月 19 日 105
1.1.1 — 2022 年 3 月 23 日 106
1.1.0 — 2022 年 2 月 23 日 106
1.0.4 — 2021 年 12 月 20 日 107
1.0.3 — 2021 年 9 月 1 日 107
1.0.2 — 2021 年 7 月 30 日 108
1.0.1 — 2021 年 5 月 31 日 108
1.0.0 — 2021 年 3 月 24 日 109
文件歷史記錄 109
..... cxi

什麼是 Amazon DCV Web 用戶端 SDK？

> **Note**
>
> Amazon DCV 先前稱為 NICE DCV。

Amazon DCV 是一種高效能遠端顯示通訊協定。它可讓您透過不同的網路條件，將遠端桌面和應用程式串流從任何雲端或資料中心安全地傳遞至任何裝置。透過搭配 Amazon EC2 使用 Amazon DCV，您可以在 Amazon EC2 執行個體上遠端執行圖形密集型應用程式。然後，您可以將結果串流至較普通的用戶端電腦，因此無需使用昂貴的專用工作站。

Amazon DCV Web 用戶端 SDK 是 JavaScript 程式庫，可用來開發自己的 Amazon DCV Web 瀏覽器用戶端應用程式。您的最終使用者可以使用這些應用程式來連線至執行中的 Amazon DCV 工作階段，並與之互動。

使用 Amazon DCV Web 用戶端 SDK 做為建置區塊，您可以建置自訂的 Web 應用程式，讓使用者從任何地方都能立即存取其桌面或應用程式，其回應能力高且流暢的效能幾乎與原生安裝的應用程式不同。

本指南說明如何使用 Amazon DCV Web Client SDK 建置自訂 Web 瀏覽器用戶端應用程式，以與工作流程中的 Amazon DCV 工作階段互動。

主題

- 先決條件
- 支援的功能
- 瀏覽器支援
- 版本控制慣例

先決條件

開始使用 Amazon DCV Web 用戶端 SDK 之前，請確定您已熟悉 Amazon DCV 和 Amazon DCV 工作階段。如需詳細資訊，請參閱 Amazon DCV 管理員指南。

Amazon DCV Web 用戶端 SDK 支援 Amazon DCV 伺服器 2020 版及更新版本。

支援的功能

您可以建置支援下列 Amazon DCV 功能的自訂 Web 瀏覽器用戶端應用程式：

- 連線至 Windows Amazon DCV 伺服器
- 連線至 Linux Amazon DCV 伺服器
- 管理串流模式
- 傳輸檔案
- 從工作階段列印
- 複製和貼上
- 立體聲 2.0 音訊播放
- 立體聲 2.0 音訊錄音 (在 Windows 伺服器上)
- 觸控螢幕
- Stylus（在 Linux、Windows 10 和 Windows Server 2019 伺服器上）
- 支援多監視器

如需這些功能的詳細資訊，請參閱《Amazon DCV 使用者指南》中的支援功能。

瀏覽器支援

Amazon DCV Web 用戶端 SDK 支援 JavaScript (ES6)，並且可以從 JavaScript 或 TypeScript 應用程式使用。

Amazon DCV Web Client SDK 支援下列 Web 瀏覽器：

瀏覽器	版本
Google Chrome	最新三個主要版本
Mozilla Firefox	最新三個主要版本
Microsoft Edge	最新三個主要版本
適用於 macOS 的 Apple Safari	最新三個主要版本

版本控制慣例

Amazon DCV Web 用戶端 SDK 版本以下列格式定義：*major*.*minor*.*patch*。版本控制慣例通常遵循語意版本控制模型。主要版本的變更，例如從 1.x.x到 2.x.x，表示已引入可能需要程式碼變更和計劃部署的中斷變更。次要版本的變更，例如從 1.1.x變更為 1.2.x，可回溯相容，但可能包含已棄用元素。

Amazon DCV Web 用戶端 SDK 入門

Amazon DCV Web 用戶端 SDK 包含主要`dcv.js`檔案和一些輔助元件。所有檔案都會分佈在壓縮封存中，可從 Amazon DCV 網站 下載。

開始使用 Amazon DCV Web 用戶端 SDK

1. Amazon DCV Web 用戶端 SDK 封存會使用安全的 GPG 簽章進行數位簽署。若要驗證封存的簽章，您必須匯入 NICE GPG 金鑰。若要這樣做，請開啟終端機視窗並匯入 NICE GPG 金鑰。

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

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

2. 從 Amazon DCV 網站 下載 Amazon DCV Web 用戶端 SDK 封存https://download.nice-dcv.com/webclientsdk.html和 Amazon DCV Web 用戶端 SDK 封存簽章。
3. 使用簽章驗證 Amazon DCV Web 用戶端 SDK 封存的簽章。

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

例如：

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

4. 如果簽章驗證成功，請擷取 Amazon DCV Web 用戶端 SDK 封存的內容，並將擷取的目錄放在 Web 伺服器上。例如：

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

```
    /
```

> **Important**
>
> - 在 Web 伺服器上部署 Amazon DCV Web 用戶端 SDK 時，您必須保留 資料夾結構。
> - 使用 Amazon DCV Web UI SDK 時，請注意 `DCVViewer` React 元件預期來自此套件的 EULA.txt third-party-licenses.txt 檔案會出現在內嵌 Web 伺服器的 URL 路徑中。third-party-licenses.txt 檔案也應該修改，以包含來自 Amazon DCV Web 用戶端 SDK 套件的對應檔案內容，以及可能來自耗用使用者應用程式所用程式庫的任何其他授權資訊。

連線至 Amazon DCV 伺服器並取得第一個影格

下列教學課程說明如何為自訂 Web 用戶端準備 HTML 頁面、如何驗證和連線至 Amazon DCV 伺服器，以及如何從 Amazon DCV 工作階段接收串流內容的第一個影格。

主題

- 步驟 1：準備您的 HTML 頁面
- 步驟 2：驗證、連線和取得第一個影格
- 獎勵：自動建立 HTML 登入表單

步驟 1：準備您的 HTML 頁面

在網頁中，您必須載入所需的 JavaScript 模組，而且必須新增 `<div>` HTML 元素，其中包含您希望 Amazon DCV Web 用戶端 SDK 從遠端 Amazon DCV 伺服器繪製內容串流的有效`id`位置。

例如：

```
<!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>
```

步驟 2：驗證、連線和取得第一個影格

本節說明如何完成使用者身分驗證程序、如何連接 Amazon DCV 伺服器，以及如何從 Amazon DCV 伺服器接收第一個內容影格。

首先，從 `index.js` 檔案匯入 Amazon DCV Web 用戶端 SDK。它可以匯入為通用模組定義 (UMD) 模組，如下所示：

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

否則，從版本 開始`1.1.0`，也可以從對應的套件匯入為 ECMAScript 模組 (ESM)，如下所示：

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

定義用來存放身分驗證物件、連線物件和 Amazon DCV 伺服器 URL 的變數。

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

在指令碼載入時，記錄 Amazon DCV Web 用戶端 SDK 版本，並在頁面載入時呼叫 `main`函數。

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

`main` 函數會設定日誌層級並啟動身分驗證程序。

```
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
    }
  );
}
```

、 `error` 和 `promptCredentials` `success`函數是必要的回呼函數，必須在身分驗證程序中定義。

如果 Amazon DCV 伺服器提示登入資料，回`promptCredentials`呼函數會從 Amazon DCV 伺服器收到請求的登入資料挑戰。如果 Amazon DCV 伺服器設定為使用系統身分驗證，則必須提供登入憑證。下列程式碼範例假設使用者名稱為 ，`my_dcv_user`密碼為 `my_password`。

如果身分驗證失敗，回`error`呼函數會從 Amazon DCV 伺服器接收錯誤物件。

如果身分驗證成功，回`success`呼函數會接收包含工作階段 ID () `sessionId` 和授權字符 (`authToken`) 的耦合陣列，以供`my_dcv_user`使用者在 Amazon DCV 伺服器上連線到每個工作階段。下列程式碼範例會呼叫連線函數，並連線到陣列中傳回的第一個工作階段。

> **Note**
>
> 在下列程式碼範例中，將 取代`MY_DCV_USER`為您自己的使用者名稱，並將 `MY_PASSWORD`取代為您自己的密碼。

```
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);
}
```

連線至 Amazon DCV 伺服器。從 Amazon DCV 伺服器接收第一個影格時，呼叫回`firstFrame`呼方法。

```
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);
  });
}
```

獎勵：自動建立 HTML 登入表單

呼叫回`promptCredentials`呼函數時，會傳回`challenge`物件。它包含名為 `requiredCredentials` 物件陣列的 屬性 - Amazon DCV 伺服器請求的每個登入資料一個物件。每個物件都包含請求的登入資料的名稱和類型。您可以使用 `challenge`和 `requiredCredentials` 物件自動建立 HTML 登入表單。

下列程式碼範例示範如何執行此操作。

```
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));
}
```

使用 Amazon DCV 功能

Amazon DCV 功能的可用性取決於為 Amazon DCV 工作階段設定的許可，以及用戶端 Web 瀏覽器的功能。

Amazon DCV 工作階段中可用的功能是由為工作階段指定的許可進行管理。這表示即使 Amazon DCV Web 用戶端 SDK 支援某項功能，根據工作階段管理員定義的許可，仍可能會阻止存取該功能。如需詳細資訊，請參閱《Amazon DCV 管理員指南》中的設定 Amazon DCV 授權。

了解 featuresUpdate 回呼函數

當 Amazon DCV 工作階段中的功能可用性變更時，Amazon DCV Web 用戶端 SDK 會使用您在建立連線時指定的回`featuresUpdate`呼函數來通知您。例如：

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

回呼函數只會通知您可用性已變更的功能。`list` 參數是字串的陣列，而且只包含已更新功能的名稱。例如，如果工作階段的音訊輸入功能的可用性變更， 參數只會包含 `["audio-in"]` 。如果稍後，剪貼簿複製和貼上功能的可用性會變更工作階段，則 參數只會包含 `["clipboard-copy", "clipboard-paste"]` 。

處理功能更新

回`featuresUpdate`呼函數只會通知您一或多個功能的可用性已變更。若要了解哪些功能已更新，您必須使用 `connection.queryFeature`方法查詢該功能。這可以在收到變更通知後隨時完成。此方法會傳回`Promise`解析為請求功能更新狀態的 。該`status`值一律相關聯，且具有名為 `enabled` 的布林值 (`true` | `false`) 屬性。某些功能在 `status`值中可能會有其他屬性。如果功能的可用性尚未更新，則會遭到拒絕。

下列範例程式碼示範如何執行此操作。

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

```
}
```

使用 Amazon DCV Web UI SDK

下列教學課程說明如何向 Amazon DCV 伺服器進行身分驗證、連接至伺服器，以及從 Amazon DCV Web UI SDK 轉譯 `DCVViewer` React 元件。

主題

- 先決條件
- 步驟 1：準備您的 HTML 頁面
- 步驟 2：驗證、連接和轉譯 DCVViewer React 元件。
- 從 AWS-UI 更新至 Cloudscape 設計系統

先決條件

您需要安裝 `React`、`ReactDOM`、`Cloudscape Design Components React` `Cloudscape Design Global Styles`和 `Cloudscape Design Design Tokens`。

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

您也需要下載 `Amazon DCV Web Client SDK`。請參閱 Amazon DCV Web 用戶端 SDK 入門 以閱讀如何執行此操作的step-by-step指南。

您必須建立用於匯入`dcv`模組的別名，因為它是 Amazon DCV Web UI SDK 的外部相依性。例如，如果您使用 Webpack 綁定 Web 應用程式，您可以使用 resolve.alias 選項，如下所示：

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

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

如果您使用彙總進行綁定，您可以安裝 @rollup/plugin-alias，並像這樣使用：

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

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

步驟 1：準備您的 HTML 頁面

在您的網頁中，您必須載入必要的 JavaScript 模組，而且應該有一個 `<div>` HTML 元素，其中包含一個有效的 `id` HTML 元素，其中會轉譯您應用程式的進入元件。

例如：

```
<!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>
```

步驟 2：驗證、連接和轉譯 `DCVViewer` React 元件。

本節說明如何完成使用者身分驗證程序、如何連接 Amazon DCV 伺服器，以及如何轉譯 `DCVViewer` React 元件。

首先，從 `index.js` 檔案匯入 `React` `ReactDOM`和您的頂層`App`元件。

```
import React from "react";
```

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

轉譯應用程式的頂層容器節點。

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

在 `App.js` 檔案中，匯入 Amazon DCV Web 用戶端 SDK 做為 ESM 模組、Amazon DCV Web UI SDK 的 `DCVViewer` React 元件，`React`以及 `Cloudscape Design Global Styles`套件。

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

以下範例示範如何在身分驗證成功的情況下，對 Amazon DCV 伺服器進行身分驗證，並從 Amazon DCV Web UI SDK 轉譯 `DCVViewer` React 元件。

```
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;
```

、 `error` 和 `promptCredentials` `success`函數是必要的回呼函數，必須在身分驗證程序中定義。

如果 Amazon DCV 伺服器提示登入資料，回`promptCredentials`呼函數會從 Amazon DCV 伺服器收到請求的登入資料挑戰。如果 Amazon DCV 伺服器設定為使用系統身分驗證，則必須以使用者名稱和密碼的形式提供憑證。

如果身分驗證失敗，回`error`呼函數會從 Amazon DCV 伺服器接收錯誤物件。

如果身分驗證成功，回`success`呼函數會接收包含工作階段 ID () `sessionId` 和授權字符 (`authToken`) 的耦合陣列，供使用者在 Amazon DCV 伺服器上連線到每個工作階段。上述程式碼範例會更新 React 狀態，以在身分驗證成功時轉譯`DCVViewer`元件。

若要進一步了解此元件接受的屬性，請參閱 Amazon DCV Web UI SDK 參考。

若要進一步了解自我簽署憑證，請參閱自我簽署憑證的重新導向說明。

從 AWS-UI 更新至 Cloudscape 設計系統

從 SDK 1.3.0 版開始，我們將`DCVViewer`元件從 AWS-UI 更新至其演變：Cloudscape Design。

Cloudscape 使用與 AWS-UI 不同的視覺化佈景主題，但基礎程式碼基礎保持不變。因此，根據 遷移應用程式`DCVViewer`應該很簡單。若要遷移，請將您安裝的 AWS-UI 相關 NPM 套件取代為相關聯的 Cloudscape 套件：

AWS-UI 套件名稱	Cloudscape 套件名稱
@awsui/components-react	@cloudscape-design/元件
@awsui/global 樣式	@cloudscape-design/全域樣式
@awsui/collection-hooks	@cloudscape-design/collection-hooks
@awsui/design-tokens	@cloudscape-design/design-tokens

如需遷移的詳細資訊，請參閱 AWS-UI GitHub 文件頁面。

SDK 參考

本節提供 Amazon DCV Web 用戶端 SDK 的說明、語法和使用範例。

主題

- DCV 模組
- 連線類別
- 身分驗證類別
- 資源類別
- Amazon DCV Web UI SDK

DCV 模組

實作 DCV 通訊協定用戶端的模組。

公開

- 方法
- 成員
- 輸入和回呼定義

方法

清單

- authenticate(authParams) → {Authentication}
- connect(config) → {Promise.<Connection>|Promise.<{code： ConnectionErrorCode， message： string}>}
- setLogHandler(handler) → {void}
- setLogLevel(level) → {void}

authenticate(authParams) → {Authentication}

啟動指定 Amazon DCV 伺服器端點的身分驗證程序。

參數：

名稱	Type	描述
`url`	string	執行中 Amazon DCV 伺服器的主機名稱和連接埠，格式如下：`https://dcv_host_address:port` 。例如：`https://my-dcv-server:8443` 。
`authenticationToken`	string	用於身分驗證的身分驗證字符。
`callbacks`	authenticationCallbacks	可在身分驗證程序期間呼叫的回呼。

傳回：

- 身分驗證物件。

Type

身分驗證

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

連線至指定的 Amazon DCV 伺服器端點。如果連線成功，它會傳回連線物件。如果連線失敗，它會傳回錯誤物件。

參數：

名稱	Type	描述
`config`	ConnectionConfig	ConnectionConfig 物件。

傳回：

- 連線物件或錯誤物件。

Type

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

setLogHandler(handler) → {void}

設定自訂日誌處理常式函數。覆寫預設日誌處理常式時，使用瀏覽器主控台進行偵錯時，原始日誌項目位置將會遺失。

參數：

名稱	Type	描述
`handler`	函數	自訂日誌處理常式函數。處理常式函數包含關卡 (number)、levelName (string)、網域 (string) 和訊息 (string)。

傳回：

Type

void

setLogLevel(level) → {void}

設定日誌層級。只有在使用預設日誌處理常式時，才需要這樣做。

參數：

名稱	Type	描述
`level`	LogLevel	欲使用的日誌層級。

傳回：

Type

void

成員

清單

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

（常數）AudioError ：AudioErrorCode

AudioError 代碼列舉。

類型：

- AudioErrorCode

（常數） AuthenticationError ：AuthenticationErrorCode

AuthenticationError 代碼列舉。

類型：

- AuthenticationErrorCode

（常數） ChannelError ：ChannelErrorCode

ChannelError 代碼列舉。

類型：

- ChannelErrorCode

（常數） ClosingReasonError ：ClosingReasonErrorCode

ClosingReasonError 代碼列舉。

類型：

- ClosingReasonErrorCode

（常數） ConnectionError ：ConnectionErrorCode

ConnectionError 代碼列舉。

類型：

- ConnectionErrorCode

（常數） CustomChannelError ：CustomChannelErrorCode

CustomChannelError 代碼列舉。

類型：

- CustomChannelErrorCode

（常數） DisplayConfigError ：DisplayConfigErrorCode

DisplayConfigError 代碼列舉。

類型：

- DisplayConfigErrorCode

（常數） FileStorageError ：FileStorageErrorCode

FileStorageError 代碼列舉。

類型：

- FileStorageErrorCode

（常數） LogLevel：LogLevel

可用的 SDK 日誌層級。

類型：

- LogLevel

（常數） MultiMonitorError ：MultiMonitorErrorCode

MultiMonitorError 程式碼列舉。

類型：

- MultiMonitorErrorCode

（常數） ResolutionError ：ResolutionErrorCode

ResolutionError 代碼列舉。

類型：

- ResolutionErrorCode

（常數） TimezoneRedirectionError ：TimezoneRedirectionErrorCode

TimezoneRedirectionError 代碼列舉。

類型：

- TimezoneRedirectionErrorCode

（常數） TimezoneRedirectionSetting ：TimezoneRedirectionSettingCode

TimezoneRedirectionSetting 代碼列舉。

類型：

- TimezoneRedirectionSettingCode

（常數） TimezoneRedirectionStatus ：TimezoneRedirectionStatusCode

TimezoneRedirectionStatus 代碼列舉。

類型：

- TimezoneRedirectionStatusCode

（常數） 版本

具有主要、次要、修補程式、修訂、延伸和 versionStr 的 Amazon DCV 版本。

屬性：

名稱	Type	描述
`major`	integer	主要版本編號。
`minor`	integer	次要版本編號。
`patch`	integer	修補程式版本編號。
`revision`	integer	修訂編號。

名稱	Type	描述
`extended`	string	延伸字串。
`versionStr`	string	主要、次要、修補程式和修訂編號的串連形式為 `major.minor.patch+build.revision` 。

（常數）ScreenshotError：ScreenshotErrorCode

ScreenshotError 代碼列舉。

類型：

- ScreenshotErrorCode

（常數）WebcamError ：WebcamErrorCode

WebcamError 代碼列舉。

類型：

- WebcamErrorCode

輸入和回呼定義

清單

- AudioErrorCode
- authenticationCallbacks
- AuthenticationErrorCode
- authErrorCallback（身分驗證、錯誤）
- authPromptCredentialsCallback （身分驗證、挑戰）
- authSuccessCallback（身分驗證、 authenticationData)
- 頻道
- ChannelErrorCode

- clipboardEventCallback（事件）
- ClosingReasonErrorCode
- 色彩空間
- connectionCallbacks
- ConnectionConfig
- ConnectionErrorCode
- createDirectory（路徑）
- CustomChannelErrorCode
- dataChannelCallback（資訊）
- deleteFile（路徑）
- deviceChangeEventCallback()
- disconnectCallback（原因）
- displayAvailabilityCallback(status， displayId)
- DisplayConfigErrorCode
- displayLayoutCallback(serverWidth、serverHeight、heads)
- 功能
- featuresUpdateCallback(featuresList)
- fileDownloadCallback(fileResource)
- filePrintedCallback(printResource)
- 檔案儲存
- filestorageEnabledCallback（已啟用）
- FileStorageErrorCode
- firstFrameCallback(resizeEnabled， relativeMouseModeEnabled， displayId)
- idleWarningNotificationCallback(disconnectionDateTime)
- collaboratorListCallback（協作者）
- licenseNotificationCallback（通知）
- list（路徑）
- LogLevel
- 監控

- MultiMonitorErrorCode
- qualityIndicatorStateCallback（狀態）
- renameDirectory(src， dest)
- renameFile(src， dest)
- ResolutionErrorCode
- retrieveFile（路徑）
- screenshotCallback（螢幕擷取畫面）
- ScreenshotErrorCode
- serverInfo
- 統計資料
- storeFile(file， dir)
- TimezoneRedirectionErrorCode
- TimezoneRedirectionSettingCode
- TimezoneRedirectionStatusCode
- WebcamErrorCode
- httpExtraSearchParamsCallback（方法、url、內文）
- httpExtraHeadersCallback（方法、url、內文）

AudioErrorCode

DCV 模組中可用的 AudioError 程式碼列舉

- `SETTING_AUDIO_FAILED`
- `CHANNEL_NOT_AVAILABLE`

類型：

- number

authenticationCallbacks

身分驗證回呼

類型：

- 物件

屬性：

名稱	Type	描述
promptCredentials	authPromptCredentialsCallback	當使用者受到憑證挑戰時要呼叫的回呼函數。
error	authErrorCallback	驗證失敗時要呼叫的回呼函數。
success	authSuccessCallback	身分驗證成功時要呼叫的回呼函數。
httpExtraSearchParamsCallback	httpExtraSearchParamsCallback	authenticate 方法中要呼叫的回呼函數，用於在啟動請求之前將自訂查詢參數插入身分驗證 URL。它也可以用於 connect方法，以自訂在建立與 DCV 伺服器的 WebSocket 連線時所使用的 URL。

AuthenticationErrorCode

DCV 模組中可用的 AuthenticationError 程式碼列舉

- 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

類型：

- number

authErrorCallback（身分驗證、錯誤）

驗證失敗時要呼叫的回呼函數。

參數：

名稱	Type	描述
authentication	身分驗證	身分驗證物件。
error	物件	身分驗證程序引發的錯誤物件。 名稱 / Type / 描述 code / AuthenticationErrorCode / 錯誤代碼。 message / string / 錯誤訊息。

authPromptCredentialsCallback（身分驗證、挑戰）

當使用者受到憑證挑戰時要呼叫的回呼函數。使用者必須提供請求的登入資料來回答挑戰。

參數：

<table>
<tr><th>名稱</th><th>Type</th><th>描述</th></tr>
<tr><td>authentication</td><td>身分驗證</td><td>身分驗證物件。</td></tr>
<tr><td>challenge</td><td>物件</td><td>挑戰。
<table>
<tr><th>名稱</th><th>Type</th><th>描述</th></tr>
<tr><td>require redentis</td><td>陣列。< 物件 ></td><td>請求的登入資料物件陣列。
<table>
<tr><th>名稱</th><th>Type</th><th>描述</th></tr>
<tr><td>n</td><td>s</td><td>請求的登入資料名稱。</td></tr>
<tr><td>t</td><td>s</td><td>請求的登入資料的</td></tr>
</table></td></tr>
</table>
</td></tr>
</table>

名稱	Type	描述
		名稱 Type 描述 類型。

authSuccessCallback (身分驗證、 authenticationData)

身分驗證成功時要呼叫的回呼函數。

參數：

名稱	Type	描述
`authentication`	身分驗證	身分驗證物件。
`authenticationData`	陣列。 < 物件 >	包含 Amazon DCV 工作階段 IDs 和身分驗證字符的物件陣列。 名稱 Type 描述 `sessior` string Amazon DCV 工作階段 ID。 `authTok` string Amazon DCV 工作階段的身分驗證字符。

頻道

可指定的可用頻道。

類型：

- "剪貼簿" | "display" | "input" | "audio" | "filestorage"

ChannelErrorCode

DCV 模組中可用的 ChannelError 程式碼列舉

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

類型：

- number

clipboardEventCallback（事件）

產生 時要呼叫`clipboardEvent`的回呼函數。

參數：

<table>
<tr><th>名稱</th><th>Type</th><th>描述</th></tr>
<tr><td>event</td><td>物件</td><td>剪貼簿事件的相關資訊。
<table><tr><th>名稱</th><th>Type</th><th>Attribus</th><th>描述</th></tr><tr><td>name</td><td>已建立 | 複製 | 貼上 |</td><td></td><td>永遠存在。事件</td></tr></table></td></tr>
</table>

名稱	Type	描述
		名稱 / Type / Attribus / 描述
		dataSilert | autoCone | newDailable | autoPaDone | remotor | pasteAlableD / 的名稱。
		`clipbData` / 物件 | 字串 / / 剪貼簿中的資料。
		`autoC` / boolea / < 選用 > / 指出是否已啟用從工作階段剪貼簿自動複製到本機用戶

名稱	Type	描述
		名稱 / Type / Attribus / 描述 端剪貼簿。 maxDaze / numbe / < 選用 > / 可放置在剪貼簿中的資料量上限。 error / string / < 選用 > / 適用時的錯誤資訊。

ClosingReasonErrorCode

DCV 模組中可用的 ClosingReasonError 程式碼列舉

- 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`

類型：

- number

色彩空間

可指定的可用色彩空間。

類型：

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

connectionCallbacks

發生連線錯誤時可呼叫的回呼。

類型：

- 物件

屬性：

名稱	Type	描述
`disconnect`	disconnectCallback	連線結束時要呼叫的回呼函數。
`displayLayout`	displayLayoutCallback	變更顯示配置或解析度時要呼叫的回呼函數。
`displayAvailability`	displayAvailabilityCallback	顯示可用性變更時要呼叫的回呼函數。

名稱	Type	描述
`firstFrame`	firstFrameCallback	從 Amazon DCV 伺服器接收第一個影格時要呼叫的回呼函數。
`filePrinted`	filePrintedCallback	在 Amazon DCV 伺服器上列印檔案時要呼叫的回呼函數。
`fileDownload`	fileDownloadCallback	準備好從 Amazon DCV 伺服器下載檔案時要呼叫的回呼函數。
`dataChannel`	dataChannelCallback	當 Amazon DCV 伺服器傳送有關資料通道可用性的通知時，要呼叫的回呼函數。
`licenseNotification`	licenseNotificationCallback	當 Amazon DCV 伺服器傳送授權狀態的通知時，要呼叫的回呼函數。
`idleWarningNotification`	idleWarningNotificationCallback	Amazon DCV 伺服器傳送閒置逾時警告時要呼叫的回呼函數。
`collaboratorList`	collaboratorListCallback	Amazon DCV 伺服器傳送協作者清單時要呼叫的回呼函數（自 Amazon DCV Web 用戶端 SDK 1.1.0 版起）。
`qualityIndicatorState`	qualityIndicatorStateCallback	當連線品質指標變更狀態時要呼叫的回呼函數。
`filestorageEnabled`	filestorageEnabledCallback	啟用或停用檔案儲存時要呼叫的回呼函數。
`featuresUpdate`	featuresUpdateCallback	當功能的狀態變更時要呼叫的回呼函數。

名稱	Type	描述
clipboardEvent	clipboardEventCallback	產生 時要呼叫clipboardEvent 的回呼函數。
deviceChangeEvent	deviceChangeEventCallback	觸發deviceChange 事件時要呼叫的回呼函數。
screenshot	screenshotCallback	當 screenshot 可用時要呼叫的回呼函數。
httpExtraSearchParamsCallback	httpExtraSearchParamsCallback	建立與 Amazon DCV 伺服器的 WebSocket 連線時，要呼叫以自訂 URL 的回呼函數。請注意，此回呼也可以與 authenticate 方法搭配使用，以在 SDK 傳送請求之前，將查詢參數動態附加至身分驗證 URL。
httpExtraHeadersCallback	httpExtraHeadersCallback	在建立連線期間，要呼叫以將自訂標頭新增至 HTTP 請求的回呼函數。

ConnectionConfig

Amazon DCV 連線的組態。

類型：

- 物件

屬性：

名稱	Type	描述
url	string	執行中 Amazon DCV 伺服器的主機名稱和連接埠，格式如下：https://d

名稱	Type	描述
		`cv_host_address:port` 。例如：`https://my-dcv-server:8443` 。
`sessionId`	string	Amazon DCV 工作階段 ID。
`authToken`	string	連線至伺服器時要使用的身分驗證字符。
`baseUrl`	string	要從中載入 SDK 檔案的絕對或相對 URL。
`resourceBaseUrl`	string	存取 DCV 資源的絕對或相對 URL。
`enabledChannels`	陣列。 < 頻道 >	指出可啟用的頻道清單。如果未指定或提供空白陣列，則預設為所有可用的頻道。
`losslessColorspace`	色彩空間	指出將使用的色彩空間。如果未指定，則預設為 "RGB"。
`divId`	string	HTML DOM 中的`div`物件 ID，其中 SDK 應使用遠端串流建立畫布。
`volumeLevel`	integer	偏好的磁碟區層級。有效範圍為 0 到 100。
`clipboardAutoSync`	boolean	指出是否針對相容的 Web 瀏覽器啟用從 Amazon DCV 工作階段剪貼簿自動複製到本機用戶端剪貼簿。
`dynamicAudioTuning`	boolean	指出是否在建立連線時，根據 Amazon DCV 伺服器音訊設定動態調整音訊。

名稱	Type	描述
`clientHiDpiScaling`	boolean	指出是否根據用戶端的 DPI 擴展畫布。
`highColorAccuracy`	boolean	指出如果可用，是否應使用高顏色準確度。如果未指定，則預設為 `false`。
`enableWebCodecs`	Boolean	指出是否應使用 WebCodecs。如果未指定，則預設為 `false`。
`observers`	connectionCallbacks	針對與連線相關的事件呼叫的回呼函數。
`callbacks`	connectionCallbacks	與 `observers` 屬性相同，但每個回呼都包含 Connection 物件做為第一個參數。

ConnectionErrorCode

DCV 模組中可用的 ConnectionError 程式碼列舉

- `ALREADY_OPEN`
- `INVALID_CONFIG`
- `INITIALIZATION_FAILED`
- `REJECTED`
- `MAIN_CHANNEL_ALREADY_OPEN`
- `GENERIC_ERROR`（自 DCV Server 2021.0 起）
- `INTERNAL_SERVER_ERROR`（自 DCV Server 2021.0 起）
- `AUTHENTICATION_FAILED`（自 DCV Server 2021.0 起）
- `PROTOCOL_ERROR`（自 DCV Server 2021.0 起）
- `INVALID_SESSION_ID`（自 DCV Server 2021.0 起）
- `INVALID_CONNECTION_ID`（自 DCV Server 2021.0 起）
- `CONNECTION_LIMIT_REACHED`（自 DCV Server 2021.0 起）
- `SERVER_UNREACHABLE`（自 DCV 伺服器 2022.1 起）

- `GATEWAY_BUSY`
- `UNSUPPORTED_CREDENTIAL`（自 DCV Server 2022.2 起）
- `TRANSPORT_ERROR`

類型：

- number

createDirectory（路徑）

參數：

名稱	Type	描述
`path`	string	伺服器上要建立目錄的絕對路徑。它還應該包含目標目錄的名稱。

CustomChannelErrorCode

DCV 模組中可用的 CustomChannelError 程式碼列舉

- `TRANSPORT_ERROR`

類型：

- number

dataChannelCallback（資訊）

當 Amazon DCV 伺服器傳送有關資料通道可用性的通知時，要呼叫的回呼函數。

參數：

名稱	Type	描述
`info`	物件	有關資料頻道的資訊。

名稱	Type	描述
		名稱 / Type / 描述 name / string / 資料通道的名稱。 token / string / 資料通道的身分驗證字符。

deleteFile（路徑）

參數：

名稱	Type	描述
`path`	string	伺服器上識別我們要刪除之檔案的絕對路徑。

deviceChangeEventCallback()

觸發`deviceChange`事件時要呼叫的回呼函數。

disconnectCallback（原因）

連線結束時要呼叫的回呼函數。

參數：

名稱	Type	描述
`reason`	物件	中斷連線的原因。

名稱	Type	描述
		名稱 / Type / 描述 code / number / 原因代碼。 message / string / 原因訊息。

displayAvailabilityCallback(status， displayId)

顯示可用性變更時要呼叫的回呼函數。

參數：

名稱	Type	描述
status	物件	顯示的狀態。 名稱 / Type / 描述 enablec / boolean / 指出是否已啟用顯示。 closed / boolean / 指出顯示是否關閉。
displayId	number	顯示的識別符。

DisplayConfigErrorCode

DCV 模組中可用的 DisplayConfigError 程式碼列舉

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

類型：

- number

displayLayoutCallback(serverWidth、serverHeight、heads)

變更顯示配置或解析度時要呼叫的回呼函數。

參數：

名稱	Type	描述
`serverWidth`	number	主要顯示器的寬度（以像素為單位）。
`serverHeight`	number	主要顯示器的高度（以像素為單位）。
`heads`	陣列。 < Monitor >	Amazon DCV 伺服器支援的顯示頭。

功能

功能值。

- `display` - 指出單一顯示器視訊串流的可用性。
- `display-multi` - 指出多顯示器視訊串流的可用性。
- `high-color-accuracy` - 指出高色彩準確性的可用性（自 Amazon DCV Web 用戶端 SDK 1.1.0 版起）。
- `mouse` - 指出滑鼠功能的可用性。
- `keyboard` - 指出鍵盤功能的可用性。
- `keyboard-sas` - 指出 SAS 序列（控制 + Alt + 刪除）功能的可用性。

- `relative-mouse` - 指出相對滑鼠模式的可用性。
- `clipboard-copy` - 指出從 Amazon DCV 伺服器到用戶端的剪貼簿複製功能可用性。
- `clipboard-paste` - 指出從用戶端到 Amazon DCV 伺服器的剪貼簿貼圖功能可用性。
- `audio-in` - 表示使用麥克風提供音訊輸入功能。
- `audio-out` - 指出音訊播放功能的可用性。
- `webcam` - 指出網路攝影機串流功能的可用性。
- `file-download` - 指出檔案下載功能從 Amazon DCV 伺服器到用戶端的可用性。
- `file-upload` - 指出從用戶端到 Amazon DCV 伺服器的檔案上傳功能的可用性。
- `timezone-redirection` - 指出時區重新導向功能的可用性（自 Amazon DCV Web 用戶端 SDK 1.3.0 版起）。

類型：

- string

featuresUpdateCallback(featuresList)

當功能的狀態變更時要呼叫的回呼函數。

參數：

名稱	Type	描述
`featuresList`	Array.<feature>	已變更的功能陣列。

fileDownloadCallback(fileResource)

準備好從 Amazon DCV 伺服器下載檔案時要呼叫的回呼函數。

參數：

名稱	Type	描述
`fileResource`	物件	已準備好下載之檔案的相關資訊。

名稱	Type	描述
		名稱 / Type / 描述 id / string / 檔案的識別符。 url / string / 用來下載檔案的 URL。 domain / string / 資源網域。 token / string / 用來下載檔案的身分驗證字符。權杖也包含在 URL 中。

filePrintedCallback(printResource)

在 Amazon DCV 伺服器上列印檔案時要呼叫的回呼函數。

參數：

名稱	Type	描述
`printResource`	物件	列印檔案的相關資訊。

名稱	Type	描述
		名稱 / Type / 描述 id / string / 列印檔案的識別符。 url / string / 用來下載列印檔案的 URL。 domain / string / 資源網域。在此情況下，為 printer。 token / string / 用來下載列印檔案的身分驗證字符。權杖也包含在 URL 中。

檔案儲存

允許在檔案系統上探索和執行動作的物件。

類型：

- 物件

屬性：

名稱	Type	描述
`list`	清單	允許 列出存在於伺服器上所提供路徑的項目（檔案和目錄）的函數。
`createDirectory`	createDirectory	允許 在伺服器上指定路徑建立目錄的 函數。
`retrieveFile`	retrieveFile	允許 在伺服器指定路徑本機下載檔案的 函數。
`deleteFile`	deleteFile	允許 在伺服器上指定路徑刪除檔案的 函數。
`renameFile`	renameFile	允許 將檔案從指定的來源路徑重新命名為指定目的地路徑的函數。
`renameDirectory`	renameDirectory	允許 將目錄從指定的來源路徑重新命名為絕對目的地路徑的函數。
`storeFile`	storeFile	允許 將本機檔案上傳至伺服器上所提供路徑的 函數。

filestorageEnabledCallback（已啟用）

啟用檔案儲存時要呼叫的回呼函數。僅限 Internet Explorer 11 上的延遲頻道。

參數：

名稱	Type	描述
`enabled`	boolean	指出是否已啟用檔案儲存。

FileStorageErrorCode

DCV 模組中可用的 FileStorageError 程式碼列舉

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

類型：

- number

firstFrameCallback(resizeEnabled， relativeMouseModeEnabled， displayId)

從 Amazon DCV 伺服器接收第一個影格時要呼叫的回呼函數。為每個顯示器發出。

參數：

名稱	Type	描述
`resizeEnabled`	boolean	指出伺服器是否支援調整用戶端顯示配置的大小。
`relativeMouseModeEnabled`	boolean	指出伺服器是否支援相對滑鼠模式。
`displayId`	number	顯示的識別符。

idleWarningNotificationCallback(disconnectionDateTime)

Amazon DCV 伺服器傳送閒置逾時警告時要呼叫的回呼函數。

參數：

名稱	Type	描述
disconnectionDateTime	日期	中斷連線的日期和時間。

collaboratorListCallback（協作者）

Amazon DCV 伺服器傳送協作器清單時要呼叫的回呼函數。

參數：

名稱	Type	描述
collaborators	陣列。< 物件 >	包含協作者相關資訊的物件清單。 名稱 / Type / 描述 usernam / string / 協作者的使用者名稱。 owner / boolean / 指出協作者是否為工作階段擁有者。 connectnId / number / 指出伺服器指派給連線的 ID。

licenseNotificationCallback（通知）

當 Amazon DCV 伺服器傳送授權狀態的通知時，要呼叫的回呼函數。

參數：

<table>
<tr><th>名稱</th><th>Type</th><th>描述</th></tr>
<tr><td>notification</td><td>物件</td><td>通知。
<table>
<tr><th>名稱</th><th>Type</th><th>描述</th></tr>
<tr><td>product</td><td>string</td><td>DCV 產品。</td></tr>
<tr><td>status</td><td>string</td><td>授權的狀態。</td></tr>
<tr><td>message</td><td>string</td><td>訊息。</td></tr>
<tr><td>leftDay</td><td>number</td><td>授權到期之前的天數。</td></tr>
<tr><td>isDemo</td><td>boolean</td><td>指出授權是否為示範授權。</td></tr>
<tr><td>numUnlinsed</td><td>number</td><td>未授權連線的數量。</td></tr>
<tr><td>licensiMode</td><td>string</td><td>授權模式。</td></tr>
<tr><td>documentionUrl</td><td>string</td><td>文件的 URL。</td></tr>
</table></td></tr>
</table>

list（路徑）

參數：

名稱	Type	描述
`path`	string	伺服器上要列出內容的絕對路徑。

LogLevel

可用的 SDK 日誌層級。

類型：

- 追蹤 | 除錯 | 資訊 | 警告 | 錯誤 | 靜音

監控

類型：

- 物件

屬性：

名稱	Type	描述
`name`	string	顯示頭的名稱。
`rect`	物件	顯示頭的相關資訊。 名稱 | Type | 描述 x | number | 顯示頭的初始x座標。

<table>
<tr><th>名稱</th><th>Type</th><th>描述</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>名稱</th><th>Type</th><th>描述</th></tr>
<tr><td>y</td><td>number</td><td>顯示頭的初始y座標。</td></tr>
<tr><td>width</td><td>number</td><td>顯示頭的寬度（以像素為單位）。</td></tr>
<tr><td>height</td><td>number</td><td>顯示頭的高度（以像素為單位）。</td></tr>
</table>
</td></tr>
<tr><td>primary</td><td>boolean</td><td>指出顯示頭是否為主要顯示頭。如果可用，這取決於遠端作業系統。</td></tr>
<tr><td>dpi</td><td>number</td><td>顯示頭的 DPI。</td></tr>
</table>

MultiMonitorErrorCode

DCV 模組中可用的 MultiMonitorError 程式碼列舉

- NO_DISPLAY_CHANNEL
- MAX_DISPLAY_NUMBER_REACHED
- INVALID_ARGUMENT
- DISPLAY_NOT_OPENED_BY_SERVER
- REQUEST_TIMEOUT
- GENERIC_ERROR

- NO_ERROR

類型：

- number

qualityIndicatorStateCallback（狀態）

當連線品質指標變更狀態時要呼叫的回呼函數。

參數：

名稱	Type	描述
state	陣列。< 物件 >	連線品質的相關資訊。 名稱 / Type / 描述 name / string / 指標的名稱。 status / NORMAL | 警告 | 嚴重 / 狀態的描述。 changed / boolean / 指出狀態是否變更。

renameDirectory(src， dest)

參數：

名稱	Type	描述
src	string	伺服器上識別要重新命名之目錄的絕對來源路徑。

名稱	Type	描述
`dest`	string	伺服器上指定目標路徑和目錄名稱的絕對目的地路徑。

renameFile(src， dest)

參數：

名稱	Type	描述
`src`	string	伺服器上識別要重新命名之檔案的絕對來源路徑。
`dest`	string	伺服器上指定目標路徑和檔案名稱的絕對目的地路徑。

ResolutionErrorCode

DCV 模組中可用的 ResolutionError 程式碼列舉

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

類型：

- number

retrieveFile（路徑）

參數：

名稱	Type	描述
`path`	string	伺服器上的絕對路徑，識別我們想要在本機下載的檔案。

screenshotCallback（螢幕擷取畫面）

螢幕擷取畫面可用時要呼叫的回呼函數。

參數：

名稱	Type	描述
`screenshot`	byte[]	PNG 格式的螢幕擷取畫面緩衝區，或螢幕擷取失敗`null`時。

ScreenshotErrorCode

DCV 模組中可用的 ScreenshotError 程式碼列舉

- `NO_CHANNEL`
- `GENERIC_ERROR`

類型：

- number

serverInfo

類型：

- 物件

屬性：

名稱	Type	描述
name	string	軟體的名稱。
version	物件	軟體版本號碼。 名稱 / Type / 描述 major / number / 主要版本編號。 minor / number / 次要版本編號。 revisic / number / 修訂版本編號。
os	string	作業系統。
arch	string	架構。
hostname	string	主機名稱。

統計資料

類型：

- 物件

屬性：

名稱	Type	描述
`fps`	number	目前每秒影格數。
`traffic`	number	位元/秒的目前流量。
`peakTraffic`	number	自建立連線以來，以位元/秒為單位的流量峰值。
`latency`	number	目前延遲，以毫秒為單位。
`currentChannels`	number	自建立連線以來已開啟的頻道數量。
`openedChannels`	number	目前開啟的頻道數量。
`channelErrors`	number	已報告錯誤的頻道數量。

storeFile(file， dir)

參數：

名稱	Type	描述
`file`	檔案	要上傳至伺服器的檔案物件（如需詳細資訊，請參閱 https：//https://developer.mozilla.org/en-US/docs/Web/API/File)。
`dir`	string	伺服器上要上傳檔案的絕對路徑。

TimezoneRedirectionErrorCode

DCV 模組中可用的 TimezoneRedirectionError 程式碼列舉

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

類型：

- number

TimezoneRedirectionSettingCode

DCV 模組中可用的 TimezoneRedirectionSetting 程式碼列舉

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

類型：

- number

TimezoneRedirectionStatusCode

DCV 模組中可用的 TimezoneRedirectionStatus 程式碼列舉

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

類型：

- number

WebcamErrorCode

DCV 模組中可用的 WebcamError 程式碼列舉

- `SETTING_WEBCAM_FAILED`

- `CHANNEL_NOT_AVAILABLE`

類型：

- number

httpExtraSearchParamsCallback（方法、url、內文）

呼叫回呼函數，以在身分驗證和建立連線期間將自訂查詢參數插入 URLs。這可啟用進階整合案例，包括能夠附加自訂查詢參數，以及新增 AWS 簽章第 4 版 (SigV4) 簽章值，以透過外部系統保護和授權連線。

此回呼也用於自訂建立與 Amazon DCV 伺服器之 WebSocket 連線時所使用的 URL。

參數：

名稱	Type	描述
`method`	string	用於請求的 HTTP 方法。
`url`	string	將用於請求的 URL。
`body`	string	請求本體的內容。

傳回：

`URLSearchParams` 物件包含要附加至 URL 的自訂查詢參數。

Type

URLSearchParams

httpExtraHeadersCallback（方法、url、內文）

在建立連線期間，要呼叫以將自訂標頭（例如 `Authorization`) 插入 HTTP 請求的回呼函數。

參數：

名稱	Type	描述
`method`	string	用於請求的 HTTP 方法。
`url`	string	將用於請求的 URL。
`body`	string	請求本體的內容。

傳回：

物件包含代表要新增至 HTTP 請求之自訂標頭的鍵/值對。

Type

物件

連線類別

呼叫 `dcv`模組的 `connect`方法所取得的連線類別。如需示範如何使用的範例，請參閱入門一節。

公開

- 方法

方法

清單

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

- enableTimezoneRedirection(enable) → {Promise|Promise.<{code： TimezoneRedirectionErrorCode， message： string}>}
- enterRelativeMouseMode() → {void}
- getConnectedDevices() → {Promise.<陣列。 < MediaDeviceInfo>>|Promise.<{message： string}>}
- getFileExplorer() → {Promise.<filestorage>|Promise.<{code： ChannelErrorCode， message： string}>}
- getServerInfo() → {serverInfo}
- getScreenshot() → {Promise|Promise.<{code： ScreenshotErrorCode， message： string}>}
- getStats() → {stats}
- latchModifierKey（金鑰、位置、isDown) → {布林值}
- openChannel(name， authToken， callbacks， namespace) → {Promise|Promise.<{code： ChannelErrorCode， message： string}>}
- queryFeature(featureName) → {Promise.<{enabled： boolean， remote？： string， autoCopy？： boolean， autoPaste？： boolean， serviceStatus？： string， available？： boolean}>| Promise.<{message： string}>}
- registerKeyboardShortcuts（快速鍵） → {void}
- requestDisplayConfig(highColorAccuracy) → {Promise|Promise.<{code： DisplayConfigErrorCode， message： string}>}
- requestDisplayLayout（配置） → {Promise|Promise.<{code： ResolutionErrorCode， message： string}>}
- requestResolution（寬度，高度） → {Promise|Promise.<{code： ResolutionErrorCode， message： string}>}
- sendKeyboardEvent(event) → {布林值}
- sendKeyboardShortcut(shortcut) → {void}
- setDisplayQuality(min， maxopt) → {void}
- setDisplayScale(scaleRatio， displayId) → {Promise|Promise.<{code： ResolutionErrorCode， message： string}>} (DEPRECATED)
- setKeyboardQuirks(quirks) → {void}
- setMaxDisplayResolution(maxWidth， maxHeight) → {void}
- setMicrophone（啟用） → {Promise|Promise.<{code： AudioErrorCode， message： string}>}
- setMinDisplayResolution(minWidth， minHeight) → {void}
- setUploadBandwidth（值） → {number}

- setVolume(volume) → {void}
- setMicrophone (啟用 ， deviceId) → {Promise|Promise.<{code： AudioErrorCode ， message： string}>}
- setWebcam (啟用 ， deviceId) → {Promise|Promise.<{code： WebcamErrorCode ， message： string}>}
- syncClipboards() → {布林值}

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

將特定顯示附加至視窗。您無法連接主要顯示器。如果成功，函數會傳回 `displayId`。

參數：

名稱	Type	描述
`win`	物件	必須連接顯示所在的視窗。
`displayConf`	物件	顯示器的組態。 名稱 | Type | Attributes | 描述 `displ` | numbe | < 選用 > | 顯示的 ID。 `displvName` | | | 顯示 div 的名稱。

傳回：

Promise。如果拒絕，承諾會傳回錯誤物件。

Type

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

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

開始或停止接聽複製貼上事件。如果是互動式剪貼簿（一律是貼上），我們需要開始聆聽複製/貼上事件。只有在需要時才開始和停止接聽可能很有用，例如，顯示模態時。

參數：

名稱	Type	Attributes	描述
`enabled`	boolean		若要開始接聽事件，請指定 `true`。若要停止接聽事件，請指定 `false`。
`win`	物件	< 選用 >	要接聽事件的視窗。如果省略，則會使用預設視窗。
`displayId`	number	< 選用 >	應該監聽事件的顯示器 ID。如果省略，則會使用視窗的預設顯示。

傳回：

Type

void

detachDisplay(displayId) → {void}

分離特定顯示。主要顯示器無法分離。

參數：

名稱	Type	描述
`displayId`	number	要分離的顯示器 ID。

傳回：

Type

void

disconnect() → {void}

中斷與 Amazon DCV 伺服器的連線，並關閉連線。

傳回：

Type

void

disconnectCollaborator(connectionId) → {void}

請求中斷與所提供連線 ID（自 Amazon DCV Web 用戶端 SDK 1.1.0 版起）連線的協作器連線。

參數：

名稱	Type	描述
`connectionId`	boolean	將中斷連線的連線 ID。

傳回：

Type

void

enableDisplayQualityUpdates(enable) → {void}

針對未接收更新的串流區域啟用或停用顯示品質更新。停用顯示品質更新可減少頻寬使用量，但也會降低顯示品質。

參數：

名稱	Type	描述
`enable`	boolean	若要啟用顯示品質更新，請指定 `true`。若要停用顯示品質更新，請指定 `false`。

傳回：

Type

void

enableHighPixelDensity（啟用）→ {void}

在用戶端上啟用或停用高像素密度。

參數：

名稱	Type	描述
`enable`	boolean	是否應該啟用高像素密度。

傳回：

Type

void

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

啟用或停用時區重新導向。啟用後，用戶端會請求伺服器使伺服器桌面時區與用戶端時區相符。

參數：

名稱	Type	描述
`enable`	boolean	若要啟用時區重新導向，請指定 `true`。若要停用時區重新導向，請指定 `false`。

傳回：

Promise。如果拒絕，承諾會傳回錯誤物件。

Type

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

enterRelativeMouseMode() → {void}

啟用相對滑鼠模式。

傳回：

Type

void

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

請求連接至用戶端電腦的媒體裝置清單。

傳回：

如果成功，它會傳回解析為 MediaDeviceInfo 物件陣列的 Promise。如需詳細資訊，請參閱 https：// https://developer.mozilla.org/en-US/docs/Web/API/MediaDeviceInfo。如果拒絕，承諾會傳回錯誤物件。

Type

Promise。 < 陣列。 < MediaDeviceInfo>> | Promise.<{message： string}>

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

取得物件來管理 Amazon DCV 伺服器的檔案儲存。

傳回：

Promise。如果已完成，則解析為檔案總管物件，如果已拒絕，則解析為錯誤物件。

Type

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

getServerInfo() → {serverInfo}

取得 Amazon DCV 伺服器的相關資訊。

傳回：

伺服器軟體的相關資訊。

Type

serverInfo

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

擷取 PNG 格式的遠端桌面螢幕擷取畫面。螢幕擷取畫面會在 screenshotCallback 觀察程式中傳回。`null`將在失敗時傳回。

傳回：

如果處理請求，會解析的承諾。如果拒絕，我們會收到錯誤物件。

Type

Promise | Promise。<{code： ScreenshotErrorCode， message： string}>

getStats() → {stats}

取得 Amazon DCV 伺服器的統計資料。

傳回：

串流統計資料的相關資訊。

Type

統計資料

latchModifierKey（金鑰、位置、isDown）→ {布林值}

傳送允許修改器的單一鍵盤`keydown`或`keyup`事件。

參數：

名稱	Type	描述
`key`	控制 | Alt | AltGraph | Meta | OS | Shift	要傳送的金鑰。
`location`	KeyboardEvent.location	金鑰的位置。如需詳細資訊，請參閱 https：//https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/location。
`isDown`	boolean	如果要注入的金鑰事件是 keydown (`true`) 或 keyup (`false`)。

傳回：

如果請求的組合有效，函數會傳回 `true`，否則會傳回 `false`。

Type

boolean

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

如果在 Amazon DCV 伺服器上建立，請在連線上開啟自訂資料通道。

參數：

名稱	Type	描述
`name`	string	頻道名稱。
`authToken`	string	用於連線至頻道的身分驗證字符。
`callbacks`	物件	要呼叫的 onMessage 和 onClose 回呼函數。
`namespace`	string	頻道的命名空間。自 Amazon DCV Web 用戶端 SDK 1.2.0 和 Amazon DCV Server 2022.1 起提供。

傳回：

Promise。如果拒絕，我們會收到錯誤物件。

Type

Promise | Promise。<{code： ChannelErrorCode， message： string}>

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

查詢特定 Amazon DCV 伺服器功能的狀態。

參數：

名稱	Type	描述
`featureName`	功能	要查詢的功能名稱。

傳回：

Promise。如果已解決，函數會傳回物件，該`status`物件一律包含 `enabled` 屬性，也可能包含其他屬性。如果拒絕，函數會傳回 `error` 物件。

Type

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

registerKeyboardShortcuts（快速鍵）→ {void}

註冊鍵盤快速鍵。

參數：

名稱	Type	描述
`shortcuts`	陣列。< 物件 >	要註冊的金鑰和映射陣列。 名稱 / Type / 描述 `sequenc` / 陣列。< 物件 > / 要註冊的鍵盤快速鍵。

名稱	Type	描述
		名稱 Type 描述
		名稱 Type 描述
		keyboardEvent.key Key 使用者按下的金鑰值。如需詳細資訊，請參閱 https://https://developer.mozilla.org/en-US/docs/

名稱	Type	描述
		名稱 Type 描述
		名稱 Type 描述
		Web/API/KeyboardEvent/key。
		location KeyboardEvent.location 要傳送的金鑰陣列。鍵盤上金鑰的位置。如需詳細資訊，請

名稱	Type	描述
		名稱 Type 描述
		名稱 Type 描述
		參閱 https：//https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/location。
		output 陣列。< 物件 > 要由捷徑執行的預期動作。

名稱	Type	描述
		名稱 Type 描述
		k K 使用者按下的金鑰值。如需詳細資訊，請參閱 https://developer.mozilla.org/en-US/docs/ boardEvent.key

名稱	Type	描述
		名稱 Type 描述
		名稱 Type 描述
		Web/API/KeyboardEvent/key。
		location 要傳送的金鑰陣列。鍵盤上金鑰的位置。如需詳細資訊，請

名稱	Type	描述
		名稱 Type 描述 參閱 https://https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/location。

傳回：

Type

void

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

從 Amazon DCV 伺服器請求更新的顯示組態。自 Amazon DCV Web 用戶端 SDK 1.1.0 和 Amazon DCV Server 2022.0 起提供。

參數：

名稱	Type	描述
`highColorAccuracy`	boolean	是否應請求高顏色準確度。

傳回：

Promise。如果拒絕，承諾會傳回錯誤物件。

Type

Promise | Promise。<{code： DisplayConfigErrorCode， message： string}>

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

請求更新連線的顯示配置。

參數：

名稱	Type	描述
`layout`	陣列。 < Monitor >	請求會顯示在配置中。

傳回：

Promise。如果拒絕，我們會收到錯誤物件。

Type

Promise | Promise。<{code： ResolutionErrorCode， message： string}>

requestResolution（寬度，高度）→ {Promise|Promise.<{code： ResolutionErrorCode， message： string}>}

從 Amazon DCV 伺服器請求更新的顯示器解析度。

參數：

名稱	Type	描述
`width`	number	請求的寬度，以像素為單位。允許的最低值為 `0`。
`height`	number	以像素為單位請求的高度。允許的最低值為 `0`。

傳回：

Promise。如果拒絕，承諾會傳回錯誤物件。

Type

Promise | Promise。<{code： ResolutionErrorCode， message： string}>

sendKeyboardEvent(event) → {布林值}

傳送鍵盤快速鍵事件。如需鍵盤事件的詳細資訊，請參閱 https：//https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent。有效的鍵盤事件包括：`keypress`、 `keydown`和 `keyup`。如需這些事件的詳細資訊，請參閱 https：//https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent#events。

參數：

名稱	Type	描述
`event`	KeyboardEvent	要傳送的鍵盤事件。

傳回：

如果事件無效，函數會傳回 `false`。如果事件有效，函數會傳回 `true`。

Type

boolean

sendKeyboardShortcut(shortcut) → {void}

傳送鍵盤快速鍵。使用此函數傳送完整 `keydown` 或 `keyup`序列。例如，傳送 Ctrl + Alt + Del 會傳送`keydown`事件之後所有金鑰`keyup`的事件。即使您想要傳送單一金鑰，也請使用此函數。

參數：

<table>
<tr><th>名稱</th><th>Type</th><th>描述</th></tr>
<tr><td>shortcut</td><td>陣列。 < 物件 ></td><td>要傳送的金鑰陣列。
<table>
<tr><th>名稱</th><th>Type</th><th>描述</th></tr>
<tr><td>key</td><td>Keyboard vent.key</td><td>使用者按下的金鑰值。如需詳細資訊，請參閱 https：// https://developer.mozilla.org/en-US/docs/Web/API/</td></tr>
</table></td></tr>
</table>

名稱	Type	描述
		名稱 / Type / 描述 KeyboardEvent/key。 `locatic` / Keyboarc vent.location / 要傳送的金鑰陣列。鍵盤上金鑰的位置。如需詳細資訊，請參閱 https：//https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/location。

傳回：

Type

void

setDisplayQuality(min， maxopt) → {void}

設定用於連線的影像品質。有效範圍為 0到 100，影像品質1最低，影像品質100最高。指定 0以保留目前的值。

參數：

名稱	Type	Attributes	描述
min	number		最低影像品質。
max	number	< 選用 >	最高影像品質。

傳回：

Type

void

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

自 1.3.0 版以來已棄用。不再需要設定顯示縮放。滑鼠座標將在內部自動管理。

通知 Amazon DCV 在用戶端擴展顯示。使用此選項來通知伺服器，其需要擴展滑鼠事件以符合用戶端的顯示比例。

參數：

名稱	Type	描述
scaleRatio	float	要使用的擴展比例。必須是嚴格正數。

名稱	Type	描述
`displayId`	number	要擴展的顯示器 ID。

傳回：

Promise。如果拒絕，承諾會傳回錯誤物件。

Type

Promise | Promise。<{code： ResolutionErrorCode， message： string}>

setKeyboardQuirks(quirks) → {void}

設定用戶端電腦的鍵盤查詢。

參數：

名稱	Type	描述
`quirks`	物件	要啟用或停用的鍵盤查詢。 名稱 / Type / 描述 `macOptiToAlt` / boolean / 若要將選項金鑰映射至 macOS 的 Alt，請指定 `true`。否則請指定 `false`。

名稱	Type	描述
		名稱 / Type / 描述 macCommdToContl / boolean / 若要將 Command 金鑰映射至 macOS 的 Ctrl，請指定 true。否則請指定 false。

傳回：

Type

void

setMaxDisplayResolution(maxWidth， maxHeight) → {void}

設定用於連線的最大顯示解析度。

參數：

名稱	Type	描述
maxWidth	number	以像素為單位的最大顯示寬度。允許的最低值為 0。
maxHeight	number	以像素為單位的最大顯示高度。允許的最低值為 0。

傳回：

Type

void

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

啟用或停用麥克風。

參數：

名稱	Type	描述
`enable`	boolean	若要啟用麥克風，請指定 `true`。若要停用麥克風，請指定 `false`。

傳回：

Promise。如果拒絕，承諾會傳回錯誤物件。

Type

Promise | Promise。<{code： AudioErrorCode， message： string}>

setMinDisplayResolution(minWidth， minHeight) → {void}

設定用於連線的最低顯示解析度。有些應用程式可能需要最低的顯示解析度。如果所需的最低解析度大於用戶端支援的最大解析度，則會使用調整大小策略。請仔細使用此函數。調整大小策略可能會導致較不精確的滑鼠和觸控輸入系統。

參數：

名稱	Type	描述
`minWidth`	number	以像素為單位的最小顯示寬度。允許的最低值為 `0`。
`minHeight`	number	以像素為單位的最小顯示高度。允許的最低值為 `0`。

傳回：

Type

void

setUploadBandwidth（值）→ {number}

設定用於將檔案上傳至 Amazon DCV 伺服器的最大頻寬。

參數：

名稱	Type	描述
`value`	number	最大頻寬限制，以 kbps 為單位。有效範圍為 1024 kbps 到 102400 kbps。

傳回：

- 設定的頻寬限制。`null`如果伺服器上已停用檔案儲存功能。

Type

number

setVolume(volume) → {void}

設定用於音訊的磁碟區層級。有效範圍為 0 到 100，0 為最低磁碟區，100 為最高磁碟區。

參數：

名稱	Type	描述
`volume`	number	要使用的磁碟區層級。

傳回：

Type

void

setMicrophone（啟用， deviceId) → {Promise|Promise.<{code： AudioErrorCode， message： string}>}

【實驗性 - 未來可能會變更】 啟用或停用麥克風。

參數：

名稱	Type	描述
`enable`	boolean	若要啟用麥克風，請指定 `true`。若要停用麥克風，請指定 `false`。
`deviceId`	string	麥克風的裝置 ID。如果未提供 deviceId，則會使用 `default` deviceId。

傳回：

Promise。如果拒絕，承諾會傳回錯誤物件。

Type

Promise | Promise。<{code： AudioErrorCode， message： string}>

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

啟用或停用網路攝影機。

參數：

名稱	Type	描述
`enable`	boolean	若要啟用網路攝影機，請指定 `true`。若要停用網路攝影機，請指定 `false`。
`deviceId`	string	網路攝影機的裝置 ID。

傳回：

承諾如果成功， 會解析為已連接/已移除的網路攝影機 deviceId。如果拒絕，承諾會傳回錯誤物件。

Type

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

syncClipboards() → {布林值}

同步本機用戶端剪貼簿與遠端 Amazon DCV 伺服器剪貼簿。瀏覽器必須支援自動複製。

傳回：

如果剪貼簿已同步，則函數會傳回 `true`。如果剪貼簿尚未同步，或瀏覽器不支援自動複製，則函數會傳回 `false`。

Type

boolean

身分驗證類別

必須使用身分驗證類別，透過呼叫 dcv模組的 authenticate方法來取得身分驗證字符。如需示範如何使用的範例，請參閱入門一節。

公開

- 方法

方法

清單

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

retry() → {void}

重試身分驗證程序。

傳回：

Type

void

sendCredentials（憑證）→ {void}

將用戶端提供的身分驗證憑證傳送至 Amazon DCV 伺服器。

參數：

名稱	Type	描述
credentials	物件	包含所提供登入資料的物件。登入資料必須具有相同的名稱，且與挑戰中指定的類型相同。

傳回：

Type

void

資源類別

資源類別可以擷取或捨棄剛列印或下載的對應檔案。執行這些動作時，對應的觀察者會分別使用資源物件做為其唯一引數`fileDownload`來叫用 `filePrinted`和 。您可以接受或拒絕這類資源，以擷取或捨棄其參考的檔案。

公開

- 方法

方法

清單

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

accept(urlParameters) → {void}

本機下載 資源。

參數：

名稱	Type	描述
`urlParameters`	物件	選用物件，其中包含傳遞給請求以擷取資源的 URL 搜尋參數金鑰/值對。

傳回：

Type

void

decline() → {void}

捨棄資源。

傳回：

Type

void

Amazon DCV Web UI SDK

JavaScript React 元件程式庫，目前正在匯出名為 的單一 React 元件`DCVViewer`，該元件會連線至 Amazon DCV 伺服器，並轉譯工具列以與遠端串流互動。

公開

- 元件

元件

清單

- DCVViewer

DCVViewer

React 元件會轉譯工具列及其所有功能，以便與遠端串流互動。

屬性：

清單

- dcv
- uiConfig

dcv

名稱	Type	必要	描述
dcv	物件	是	定義與 Amazon DCV Server 建立連線所需的屬性、設定日誌層級以及從何處載入 Amazon DCV Web 用戶端 SDK 資產和存取 DCV 資源的 URL 的物件。 名稱 / Type / 必要 / 描述 ses / 字串 / 是 / Amazon DCV 工作階段 ID。 aut / 字串 / 是 / 連線至伺服器時要使用的身分

名稱	Type	必要	描述
			名稱 / Type / 必要 / 描述 驗證字符。 ser / 字串 / 是 / 執行中 Amazon DCV 伺服器的主機名稱和連接埠，格式如下：https：//h ttps://dc v_host_ad dress:por

名稱	Type	必要	描述
			名稱 / Type / 必要 / 描述 t。例如：https://https://my-dcv-server:8443。 bas / 字串 / 是 / 要從中載入 SDK 檔案的絕對或相對 URL。 res ase / 字串 / 否（預設： / 存取 DCV 資

名稱	Type	必要	描述
			名稱 | Type | 必要 | 描述 源的絕對或相對 URL。 onDect | 函數 | 否（預設：=> {}) | 中斷與 Amazon DCV 伺服器的連線時叫用回呼函數，且連線已

名稱	Type	必要	描述
			名稱 Type 必要 描述 關閉。 log LogLevel. 否（預設：INF 要在檢視器中使用的日誌層級。

uiConfig

名稱	Type	必要	描述
uiConfig	物件	否（預設：{})	定義屬性的物件，以設定工具列是否可見，以及是否要在工具列上顯示全螢幕和多監視器按鈕。

名稱	Type	必要	描述

名稱	Type	必要	描述
too	物件	否（預設：	物件定義工具列的組態選項。

名稱	Type	必要	描述
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]：true 否顯示或隱藏工具列的選項。

名稱	Type	必要	描述
			名稱 / Type / 必要 / 描述
			必要 / 描述
			否 / 顯示或隱藏工具列上全螢幕按鈕的選項。
			否 / 在工具

名稱	Type	必要	描述
			名稱 Type 必要 描述
			必要 Type 描述
			列上顯示或隱藏多顯示器按鈕的選項。

Amazon DCV Web 用戶端 SDK 的版本備註和文件歷史記錄

此頁面提供 Amazon DCV Web 用戶端 SDK 的版本備註和文件歷史記錄。

主題

- Amazon DCV Web 用戶端 SDK 版本備註
- 文件歷史記錄

Amazon DCV Web 用戶端 SDK 版本備註

本節依發行日期提供 Amazon DCV Web 用戶端 SDK 的發行備註。

主題

- 1.8.7 — 2024 年 10 月 31 日
- 1.8.4 — 2024 年 10 月 1 日
- 1.5.10 — 2023 年 12 月 19 日
- 1.5.6 — 2023 年 11 月 9 日
- 1.4.4 — 2023 年 6 月 29 日
- 1.4.0 — 2023 年 3 月 28 日
- 1.3.1 — 2022 年 12 月 9 日
- 1.3.0 — 2022 年 11 月 11 日
- 1.2.1 — 2022 年 7 月 21 日
- 1.2.0 — 2022 年 6 月 29 日
- 1.1.3 — 2022 年 5 月 23 日
- 1.1.2 — 2022 年 5 月 19 日
- 1.1.1 — 2022 年 3 月 23 日
- 1.1.0 — 2022 年 2 月 23 日
- 1.0.4 — 2021 年 12 月 20 日
- 1.0.3 — 2021 年 9 月 1 日
- 1.0.2 — 2021 年 7 月 30 日

- 1.0.1 — 2021 年 5 月 31 日
- 1.0.0 — 2021 年 3 月 24 日

1.8.7 — 2024 年 10 月 31 日

建置編號	變更與錯誤修正	
• 語意版本：1.8.7 • 組建：858	• 修正 Firefox 130 和更新版本的轉譯	

1.8.4 — 2024 年 10 月 1 日

建置編號	新功能	變更與錯誤修正
• 語意版本：1.8.4 • 組建：840	已新增下列功能： • 已重新命名為「Amazon DCV Web 用戶端 SDK」 • 新增適用於高 dpi 顯示器的新 API enableHighPixelDensity • 新增實驗性 API setMicrophone，以選取相容瀏覽器中的麥克風 • 新增連線錯誤 GATEWAY_BUSY、UNSUPPORTED_CREDENTIAL 和 TRANSPORT_ERROR • 新增關閉原因 EXTERNAL_PROTOCOL_CONNECTION_EVICTED 和 DISCONNECTION_REQUESTED	• 改善網路攝影機處理 • 改善音訊播放處理 • 改善 WebCodecs 處理 • 改善了麥克風和網路攝影機的插頭和插頭 • 改善多監視器時的遠端視窗拖曳 • 檔案儲存上傳和下載許可現在已正確傳播 • 轉譯的次要修正

1.5.10 — 2023 年 12 月 19 日

版本	版本備註
• 語意版本：1.5.10 • 組建：684	變更與錯誤修正 • 修正串流解碼錯誤

1.5.6 — 2023 年 11 月 9 日

版本	版本備註
• 語意版本：1.5.6 • 組建：659	變更與錯誤修正 • 串流解碼和轉譯的效能改善 • 已移除對 Internet Explorer 11 的支援

1.4.4 — 2023 年 6 月 29 日

版本	版本備註
• 語意版本：1.4.4 • 組建：573	變更與錯誤修正 • 檢視器使用者介面元件現在會在支援 API 的瀏覽器上使用 `navigator.keyboard.lock` API，以全螢幕處理特殊金鑰。 • 修正使用 Chrome 114 或更新版本時，可能會導致顏色錯誤的問題。 •

版本	版本備註
	改善 WebCodecs 偵測。 • 修正進入視窗時滑鼠按鈕狀態的問題。 • 已修正可能導致修改器金鑰在 macOS 上保持按下狀態的問題。 • 改善音訊穩定性以降低網路條件。 • 修正記憶體洩漏。 • 改善日誌以包含時間和層級。

1.4.0 — 2023 年 3 月 28 日

版本	版本備註
• 語意版本：1.4.0 • 組建：476	新功能 • 已將新`uploadFiles` 方法新增至`FileStorage` 物件，以上傳多個檔案。 • 檢視器 UI 元件現在支援拖放以啟動檔案上傳。 • WebCodecs 瀏覽器 API 現在也用於音訊和網路攝影機。 變更與錯誤修正 • 修正與來自相同頁面的重複連線相關的記憶體流失。

版本	版本備註
	• `setUploadBandwidth` 現在允許高達 1 Gbps 的值。 • 最佳化 UI 元件的轉譯。 • 修正對 Windows 上動畫游標的支援。 • 已修正相同操作同時存在文字和影像資料時，剪貼簿支援的問題。 • 改善了 Webcam API 的穩健性：請求正在進行時，無法變更設定，`webcam.setEnabled` 現在會追蹤請求的裝置 ID，並傳回 Promise。檢視器使用者介面元件會在發生錯誤時顯示通知。

1.3.1 — 2022 年 12 月 9 日

版本	版本備註
• 語意版本：1.3.1 • 組建：413	變更與錯誤修正 • 修正可能導致時區重新導向 UI 與伺服器不同步的問題。 • 已修正多次重新連線後的記憶體洩漏。 • 已修正造成不連貫上空白頁面的問題。 • 已修正導致主控台警告音訊解碼器關閉的錯誤。

1.3.0 — 2022 年 11 月 11 日

版本	版本備註
• 語意版本：1.3.0 • 組建：407	新功能 • UI Viewer 元件的採用 Cloudscape (https://cloudscape.design)。 • 新增時區重新導向的支援。 變更與錯誤修正 • 修正 DCV 檢視器聚焦時，非同步剪貼簿上遺失的更新。 • 擴展用戶端的顯示時，不再需要 `setDisplayScale` 函數。 • `DCVViewer` 元件現在會在卸載`disconnect()` 時自動呼叫。

1.2.1 — 2022 年 7 月 21 日

版本	版本備註
• 語意版本：1.2.1 • 組建：358	變更與錯誤修正 • 修正導致無法連線至 Amazon DCV 伺服器 2019.1 及更舊版本的問題。

1.2.0 — 2022 年 6 月 29 日

版本	版本備註
• 語意版本：1.2.0 • 組建：352	變更與錯誤修正 • 修正當接收的影格大於支援的最大解析度 (4096x2160) 時的當機錯誤。 • 資源物件（以引數傳遞給 `fileDownload` 和 `filePrinted` 觀察者）現在具有 `accept`和 `decline`方法，可以在物件上呼叫，分別下載和捨棄資源。 • 中斷連線時自動剪貼簿同步的小錯誤修正。

1.1.3 — 2022 年 5 月 23 日

版本	版本備註
• 語意版本：1.1.3 • 組建：329	變更與錯誤修正 • 修正指定 web-url-path 選項時無法成功連線的問題。

1.1.2 — 2022 年 5 月 19 日

版本	版本備註
• 語意版本：1.1.2 •	變更與錯誤修正 •

版本	版本備註
組建：322	修正連線後可能導致輸入無法正常運作的問題。 • 已修正縮放比例大於 1 時的滑鼠座標。

1.1.1 — 2022 年 3 月 23 日

版本	版本備註
• 語意版本：1.1.1 • 組建：309	變更與錯誤修正 • 與伺服器的通訊逾時`Transport Error`時報告。 • 修正串流大型解析度時的重複解碼錯誤。

1.1.0 — 2022 年 2 月 23 日

版本	版本備註
• 語意版本：1.1.0 • 組建：295	新功能 • 使用 `DCVViewer` React 元件發行 Amazon DCV Web UI SDK 程式庫。 • 匯出 Amazon DCV Web 用戶端 SDK 做為 UMD 和 ES 模組。 • 新增了高色彩準確度支援。 •

版本	版本備註
	新增列出並與連線至工作階段的用戶端互動的功能。新增連線和中斷連線的通知。 變更與錯誤修正 • 改善 Webcodec 解碼支援。 • 各種鍵盤改進。 • 修正剪貼簿停用時，阻止開啟第二個畫面的錯誤。

1.0.4 — 2021 年 12 月 20 日

版本	版本備註
• 語意版本：1.0.4 • 組建：249	新功能 • 支援從相同頁面開啟多個連線。 • 支援從 CDN 載入 SDK。

1.0.3 — 2021 年 9 月 1 日

版本	版本備註
• 語意版本：1.0.3 • 組建：202	新功能 •

版本	版本備註
	WebCodecs 的實驗性支援。依預設會停用此功能，而且必須使用新的 屬性透過 `ConnectionConfig` 物件啟用`enableWebCodecs` 。 • 剪貼簿：新增對 Chromium 瀏覽器上`image/png` 資料類型的支援。 • 新增觀察者/回呼，以取得伺服器的螢幕擷取畫面做為 PNG 映像（需要 Amazon DCV 伺服器 2021.2)。 變更與錯誤修正 • 改善鍵盤修改器的處理。

1.0.2 — 2021 年 7 月 30 日

版本	版本備註
• 語意版本：1.0.2 • 組建：167	• 修正手寫筆事件的壓力偵測。 • 已改善對 Chrome 上的韓文鍵盤配置的支援。

1.0.1 — 2021 年 5 月 31 日

版本	版本備註
• 語意版本：1.0.1	• 修正連線錯誤傳播和關閉原因

版本	版本備註
• 組建：141	• 已修正檔案儲存區塊進度更新 • 改善網路攝影機處理 • 改善音訊輸入處理

1.0.0 — 2021 年 3 月 24 日

版本	版本備註
• 語意版本：1.0.0 • 組建：81	Amazon DCV Web 用戶端 SDK 的初始版本。

文件歷史記錄

下表說明此 Amazon DCV Web 用戶端 SDK 版本的文件。

變更	描述	日期
Amazon DCV Web 用戶端 SDK 1.8.4 版	Amazon DCV Web 用戶端 SDK 1.8.4 現已推出。如需詳細資訊，請參閱 SDK v.1.8.4。	2024 年 10 月 1 日
Amazon DCV Web 用戶端 SDK 1.5.6 版	Amazon DCV Web 用戶端 SDK 1.5.6 現已推出。如需詳細資訊，請參閱 SDK v.1.5.6。	2023 年 11 月 9 日
Amazon DCV Web 用戶端 SDK 1.4.4 版	Amazon DCV Web 用戶端 SDK 1.4.4 現已推出。如需詳細資訊，請參閱 SDK v.1.4.4。	2023 年 6 月 29 日

變更	描述	日期
Amazon DCV Web 用戶端 SDK 1.4.0 版	Amazon DCV Web 用戶端 SDK 1.4.0 現已推出。如需詳細資訊，請參閱 SDK v.1.4.0。	2023 年 3 月 28 日
Amazon DCV Web 用戶端 SDK 1.3.1 版	Amazon DCV Web 用戶端 SDK 1.3.1 現已推出。如需詳細資訊，請參閱 SDK v.1.3.1。	2022 年 12 月 9 日
Amazon DCV Web 用戶端 SDK 1.3.0 版	Amazon DCV Web 用戶端 SDK 1.3.0 現已推出。如需詳細資訊，請參閱 SDK v.1.3.0。	2022 年 11 月 11 日
Amazon DCV Web 用戶端 SDK 1.2.0 版	Amazon DCV Web 用戶端 SDK 1.2.0 現已推出。如需詳細資訊，請參閱 SDK v.1.2.0。	2022 年 6 月 29 日
Amazon DCV Web 用戶端 SDK 1.1.0 版	Amazon DCV Web 用戶端 SDK 1.1.0 現已推出。如需詳細資訊，請參閱 SDK v.1.1.0。	2022 年 2 月 23 日
Amazon DCV Web 用戶端 SDK 1.0.1 版	已修正一些錯字。修正次要錯誤，請參閱 SDK v.1.0.1。	2021 年 5 月 31 日
初始版本	此內容的首次發佈。	2021 年 3 月 24 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。