



웹 클라이언트 SDK 개발자 안내서

Amazon DCV



Amazon DCV: 웹 클라이언트 SDK 개발자 안내서

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

Amazon DCV 웹 클라이언트 SDK란 무엇인가요?	1
사전 조건	1
지원 기능	2
브라우저 지원	2
버전 관리 규칙	3
시작하기	4
Amazon DCV 서버에 연결하고 첫 번째 프레임 가져오기	5
1단계: HTML 페이지 준비	5
2단계: 인증, 연결 및 첫 번째 프레임 가져오기	6
보너스: HTML 로그인 양식 자동 생성	9
Amazon DCV 기능으로 작업하기	10
featuresUpdate 콜백 함수 이해	10
기능 업데이트 처리	10
Amazon DCV 웹 UI SDK 사용	11
사전 조건	11
1단계: HTML 페이지 준비	12
2단계: DCVViewer React 구성 요소 인증, 연결 및 렌더링	13
AWS-UI에서 Cloudscape Design System으로 업데이트	17
SDK 참조	18
DCV 모듈	18
메서드	18
Members	21
유형 및 콜백 정의	25
연결 클래스	65
메서드	18
인증 클래스	94
메서드	18
리소스 클래스	96
메서드	18
Amazon DCV 웹 UI SDK	97
구성 요소	97
릴리스 정보 및 문서 기록	112
릴리스 정보	112
1.13.2 — 2026년 2월 19일	113

1.10.1 - 2025년 10월 22일	113
1.9.100 - 2025년 7월 2일	114
1.8.7 - 2024년 10월 31일	114
1.8.4 - 2024년 10월 1일	115
1.5.10 - 2023년 12월 19일	115
1.5.6 — 2023년 11월 9일	116
1.4.4 — 2023년 6월 29일	116
1.4.0 — 2023년 3월 28일	117
1.3.1 — 2022년 12월 9일	118
1.3.0 — 2022년 11월 11일	119
1.2.1 — 2022년 7월 21일	119
1.2.0 — 2022년 6월 29일	120
1.1.3 — 2022년 5월 23일	120
1.1.2 — 2022년 5월 19일	121
1.1.1 — 2022년 3월 23일	121
1.1.0 — 2022년 2월 23일	121
1.0.4 — 2021년 12월 20일	122
1.0.3 — 2021년 9월 1일	123
1.0.2 — 2021년 7월 30일	123
1.0.1 — 2021년 5월 31일	124
1.0.0 — 2021년 3월 24일	124
문서 기록	125
.....	cxxvii

Amazon DCV 웹 클라이언트 SDK란 무엇인가요?

Note

Amazon DCV는 이전에 NICE DCV로 알려졌습니다.

Amazon DCV는 고성능 원격 디스플레이 프로토콜입니다. 이를 통해 다양한 네트워크 조건에서 클라우드 또는 데이터 센터에서 모든 디바이스로 원격 데스크톱 및 애플리케이션 스트리밍을 안전하게 제공할 수 있습니다. Amazon EC2와 함께 Amazon DCV를 사용하면 그래픽 집약적 애플리케이션을 Amazon EC2 인스턴스에서 원격으로 실행할 수 있습니다. 그런 다음 결과를 적절한 클라이언트 머신으로 스트리밍할 수 있으므로 비용이 많이 드는 전용 워크스테이션이 필요하지 않습니다.

Amazon DCV 웹 클라이언트 SDK는 자체 Amazon DCV 웹 브라우저 클라이언트 애플리케이션을 개발하는 데 사용 가능한 JavaScript 라이브러리입니다. 최종 사용자는 이러한 애플리케이션을 사용하여 실행 중인 Amazon DCV 세션에 연결하고 상호 작용할 수 있습니다.

Amazon DCV 웹 클라이언트 SDK를 빌딩 블록으로 사용하면 기본적으로 설치된 애플리케이션과 거의 차이가 없는 반응성과 유연한 성능으로 어디서나 데스크톱 또는 애플리케이션에 즉시 액세스할 수 있는 맞춤형 웹 애플리케이션을 구축할 수 있습니다.

이 안내서에서는 Amazon DCV 웹 클라이언트 SDK를 사용하여 워크플로 내에서 Amazon DCV 세션과 상호 작용하는 사용자 지정 웹 브라우저 클라이언트 애플리케이션을 구축하는 방법을 설명합니다.

주제

- [사전 조건](#)
- [지원 기능](#)
- [브라우저 지원](#)
- [버전 관리 규칙](#)

사전 조건

Amazon DCV 웹 클라이언트 SDK로 작업을 시작하려면 Amazon DCV 및 Amazon DCV 세션에 대해 잘 알고 있어야 합니다. 자세한 내용은 [Amazon DCV 관리자 안내서](#)를 참조하세요.

Amazon DCV 웹 클라이언트 SDK는 Amazon DCV 서버 버전 2020 이상을 지원합니다.

지원 기능

다음과 같은 Amazon DCV 기능을 지원하는 사용자 지정 웹 브라우저 클라이언트 애플리케이션을 구축할 수 있습니다.

- Windows Amazon DCV 서버에 연결
- Linux Amazon DCV 서버에 연결
- 스트리밍 모드 관리
- 파일 전송
- 세션에서 인쇄
- 복사 및 붙여넣기
- Stereo 2.0 오디오 재생
- Stereo 2.0 오디오 레코딩(Windows 서버)
- 터치스크린
- 스타일러스(Linux, Windows 10, Windows Server 2019 서버)
- 다중 모니터 지원

이러한 기능에 대한 자세한 내용은 Amazon DCV 사용 설명서의 [지원 기능](#)을 참조하세요.

브라우저 지원

Amazon DCV 웹 클라이언트 SDK는 JavaScript(ES6)를 지원하며 JavaScript 또는 TypeScript 애플리케이션에서 사용할 수 있습니다.

Amazon DCV 웹 클라이언트 SDK는 다음 웹 브라우저를 지원합니다.

브라우저	버전
Google Chrome	최신 3개 주요 버전
Mozilla Firefox	최신 3개 주요 버전
Microsoft Edge	최신 3개 주요 버전
macOS용 Apple Safari	최신 3개 주요 버전

버전 관리 규칙

Amazon DCV 웹 클라이언트 SDK 버전은 *major.minor.patch* 형식으로 정의됩니다. 버전 관리 규칙은 일반적으로 [시맨틱 버전 관리 모델](#)을 따릅니다. 메이저 버전이 변경된 경우(예: 1.x.x에서 2.x.x로) 코드 변경 및 계획된 배포가 필요할 수 있는 주요 변경 사항이 도입되었음을 나타냅니다. 마이너 버전이 변경된 경우(예: 1.1.x에서 1.2.x로) 이전 버전과 호환되지만 더 이상 사용되지 않는 요소가 포함될 수 있습니다.

Amazon DCV 웹 클라이언트 SDK 시작하기

Amazon DCV 웹 클라이언트 SDK는 기본 `dcv.js` 파일과 일부 보조 구성 요소로 구성되어 있습니다. 모든 파일은 [Amazon DCV 웹사이트](#)에서 다운로드할 수 있는 압축 아카이브에 배포됩니다.

Amazon DCV 웹 클라이언트 SDK를 시작하려면 다음과 같이 하세요.

1. Amazon DCV 웹 클라이언트 SDK 아카이브는 안전한 GPG 서명으로 디지털 서명됩니다. 아카이브의 서명을 확인하려면 NICE GPG 키를 가져와야 합니다. 이렇게 하려면 터미널 창을 열고 NICE GPG 키를 가져옵니다.

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

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

2. [Amazon DCV 웹사이트](#)에서 Amazon DCV 웹 클라이언트 SDK 아카이브 및 Amazon DCV 웹 클라이언트 SDK 아카이브 서명을 다운로드합니다.
3. 서명을 사용하여 Amazon DCV 웹 클라이언트 SDK 아카이브의 서명을 확인합니다.

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

예제:

```
$ gpg --verify nice-dcv-web-client-sdk-1.10.1-1011.zip.sign nice-dcv-web-client-  
sdk-1.10.1-1011.zip
```

4. 서명이 성공적으로 확인되면 Amazon DCV 웹 클라이언트 SDK 아카이브의 내용을 추출하고, 추출된 디렉터리를 웹 서버에 배치합니다. 예제:

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

```
server_directory
/
```

⚠ Important

- 웹 서버에 Amazon DCV 웹 클라이언트 SDK를 배포할 때는 폴더 구조를 유지해야 합니다.
- Amazon DCV 웹 UI SDK를 사용할 때, DCVViewer React 구성 요소는 이 패키지의 EULA.txt 및 third-party-licenses.txt 파일이 임베디드 웹 서버의 URL 경로에 있을 것으로 예상한다는 점에 유의하세요. third-party-licenses.txt 파일은 Amazon DCV 웹 클라이언트 SDK 패키지의 해당 파일 내용과 사용 중인 사용자 애플리케이션에서 이용하는 라이브러리의 기타 라이선스 정보도 포함하도록 수정해야 합니다.

Amazon DCV 서버에 연결하고 첫 번째 프레임 가져오기

다음 튜토리얼에서는 사용자 지정 웹 클라이언트용 HTML 페이지를 준비하는 방법, Amazon DCV 서버를 인증하고 연결하는 방법, Amazon DCV 세션에서 스트리밍된 콘텐츠의 첫 번째 프레임을 수신하는 방법을 보여줍니다.

주제

- [1단계: HTML 페이지 준비](#)
- [2단계: 인증, 연결 및 첫 번째 프레임 가져오기](#)
- [보너스: HTML 로그인 양식 자동 생성](#)

1단계: HTML 페이지 준비

웹 페이지에서 필요한 JavaScript 모듈을 로드해야 하며 Amazon DCV 웹 클라이언트 SDK가 원격 Amazon DCV 서버에서 콘텐츠 스트림을 가져오도록 하려는 위치에 유효한 id가 있는 <div> HTML 요소를 추가해야 합니다.

예제:

```
<!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>
```

2단계: 인증, 연결 및 첫 번째 프레임 가져오기

이 섹션에서는 사용자 인증 프로세스를 완료하는 방법, Amazon DCV 서버를 연결하는 방법, Amazon DCV 서버에서 콘텐츠의 첫 번째 프레임을 수신하는 방법을 보여줍니다.

먼저 `index.js` 파일에서 Amazon DCV 웹 클라이언트 SDK를 가져옵니다. 다음과 같이 UMD(범용 모듈 정의) 모듈로 가져올 수 있습니다.

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

아니면 1.1.0 버전부터는 다음과 같이 해당 패키지에서 ECMAScript 모듈(ESM)로 가져올 수도 있습니다.

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

인증 객체, 연결 객체 및 Amazon DCV 서버 URL을 저장하는 데 사용할 변수를 정의합니다.

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

스크립트 로드 시 Amazon DCV 웹 클라이언트 SDK 버전을 로깅하고 페이지 로드 시 `main` 함수를 직접적으로 호출합니다.

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

`main` 함수는 로그 수준을 설정하고 인증 프로세스를 시작합니다.

```
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
    }
  );
}
```

`promptCredentials`, `error`, `success` 함수는 인증 프로세스에서 정의해야 하는 필수 콜백 함수입니다.

Amazon DCV 서버에서 자격 증명을 요청하는 메시지가 표시되면 `promptCredentials` 콜백 함수는 Amazon DCV 서버로부터 요청된 자격 증명 챌린지를 수신합니다. Amazon DCV 서버가 시스템 인증을 사용하도록 구성된 경우 로그인 자격 증명을 제공해야 합니다. 다음 코드 샘플에서는 사용자 이름이 `my_dcv_user`이고 암호가 `my_password`인 것으로 가정합니다.

인증이 실패하면 `error` 콜백 함수는 Amazon DCV 서버로부터 오류 객체를 수신합니다.

인증에 성공하면 `success` 콜백 함수는 Amazon DCV 서버에서 `my_dcv_user` 사용자가 연결할 수 있는 각 세션의 세션 ID(`sessionId`) 및 권한 부여 토큰(`authToken`)이 함께 포함된 배열을 수신합니다. 다음 코드 예제는 `connect` 함수를 직접적으로 호출하고 배열에 반환된 첫 번째 세션에 연결합니다.

Note

다음 코드 예제에서는 `MY_DCV_USER`를 사용자 이름으로, `MY_PASSWORD`를 암호로 바꿉니다.

```
function onPromptCredentials(auth, challenge) {
  // Let's check if in challenge 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);
}

```

Amazon DCV 서버에 연결합니다. Amazon DCV 서버에서 첫 프레임을 수신하면 `firstFrame` 콜백 메서드가 직접적으로 호출됩니다.

```

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);
  });
}

```

보너스: HTML 로그인 양식 자동 생성

`promptCredentials` 콜백 함수가 직접적으로 호출되면 `challenge` 객체가 반환됩니다. 여기에는 Amazon DCV 서버에서 요청하는 자격 증명당 하나의 객체, 즉 객체 배열에 해당하는 `requiredCredentials` 속성이 포함됩니다. 각 객체에는 요청된 자격 증명의 이름과 유형이 포함됩니다. `challenge` 및 `requiredCredentials` 객체를 사용하여 HTML 로그인 양식을 자동으로 만들 수 있습니다.

다음 코드 예제에서는 이를 수행하는 방법을 보여줍니다.

```
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));
  }

```

Amazon DCV 기능으로 작업하기

Amazon DCV 기능의 사용 가능 여부는 Amazon DCV 세션에 대해 구성된 권한 및 클라이언트 웹 브라우저의 기능에 따라 달라집니다.

Amazon DCV 세션에서 사용할 수 있는 기능은 세션에 지정된 권한으로 관리됩니다. 즉, Amazon DCV 웹 클라이언트 SDK에서 기능을 지원하는 경우에도 세션 관리자가 정의한 권한에 따라 해당 기능에 대한 액세스가 차단될 수 있습니다. 자세한 내용은 Amazon DCV 관리자 안내서의 [Amazon DCV 권한 부여 구성](#)을 참조하세요.

featuresUpdate 콜백 함수 이해

Amazon DCV 세션의 기능 사용 가능 여부가 변경되면 Amazon DCV 웹 클라이언트 SDK는 연결을 설정할 때 지정한 featuresUpdate 콜백 함수를 사용하여 사용자에게 알립니다. 예제:

```

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

```

콜백 함수는 가용성이 변경된 기능만 사용자에게 알립니다. list 파라미터는 문자열 배열이며 업데이트된 기능의 이름만 포함합니다. 예를 들어 세션에서 오디오 입력 기능의 사용 가능 여부가 변경될 경우, 파라미터에는 ["audio-in"]만 포함됩니다. 나중에 세션에 대한 클립보드 복사 및 붙여넣기 기능의 사용 가능 여부가 변경될 경우 파라미터에는 ["clipboard-copy", "clipboard-paste"]만 포함됩니다.

기능 업데이트 처리

featuresUpdate 콜백 함수는 하나 이상의 기능에 대한 가용성이 변경되었다는 정보만 알려줍니다. 어떤 기능이 변경되었는지 확인하려면 connection.queryFeature 메서드를 사용하여 기능을 쿼리해야 합니다. 변경 알림을 받은 후 언제든지 이 작업을 수행할 수 있습니다. 이 메서드는 요청된 기능

의 업데이트 상태를 확인하는 Promise를 반환합니다. status 값은 항상 연결되어 있으며 enabled 값이 직접 호출되는 부울(true | false) 속성이 있습니다. 일부 기능의 status 값에 추가 속성이 있을 수 있습니다. 기능의 가용성이 변경되지 않은 경우 해당 기능은 거부됩니다.

다음 코드 예제에서는 이를 수행하는 방법을 보여줍니다.

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

Amazon DCV 웹 UI SDK 사용

다음 튜토리얼은 Amazon DCV 서버에 대해 인증하고, 연결하고, Amazon DCV 웹 UI SDK에서 DCVViewer React 구성 요소를 렌더링하는 방법을 보여줍니다.

주제

- [사전 조건](#)
- [1단계: HTML 페이지 준비](#)
- [2단계: DCVViewer React 구성 요소 인증, 연결 및 렌더링](#)
- [AWS-UI에서 Cloudscape Design System으로 업데이트](#)

사전 조건

React , ReactDOM , Cloudscape Design Components React , Cloudscape Design Global Styles, Cloudscape Design Design Tokens를 설치해야 합니다.

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

Amazon DCV Web Client SDK도 다운로드해야 합니다. 이 작업에 대한 단계별 안내는 [Amazon DCV 웹 클라이언트 SDK 시작하기](#) 섹션을 참조하세요.

dcv 모듈은 Amazon DCV 웹 UI SDK의 외부 종속성이므로, 이 모듈을 가져오기 위한 별칭을 만들어야 합니다. 예를 들어 webpack을 사용하여 웹 앱을 번들로 제공하는 경우, 다음과 같이 [resolve.alias](#) 옵션을 사용할 수 있습니다.

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

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

번들링에 롤업을 사용하는 경우 [@rollup/plugin-alias](#)를 설치하고 다음과 같이 사용할 수 있습니다.

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

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

1단계: HTML 페이지 준비

웹 페이지에서 필수 JavaScript 모듈을 로드해야 하며 앱의 입력 구성 요소가 렌더링될 위치에 유효한 id가 있는 <div> HTML 요소가 있어야 합니다.

예제:

```
<!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>

```

2단계: DCVViewer React 구성 요소 인증, 연결 및 렌더링

이 섹션에서는 사용자 인증 프로세스를 완료하는 방법, Amazon DCV 서버를 연결하는 방법, DCVViewer React 구성 요소를 렌더링하는 방법을 보여줍니다.

먼저 `index.js` 파일에서 React, ReactDOM 및 최상위 App 구성 요소를 가져옵니다.

```

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

```

앱의 최상위 컨테이너 노드를 렌더링합니다.

```

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

```

`App.js` 파일에서 Amazon DCV 웹 클라이언트 SDK를 ESM 모듈로 가져오고, Amazon DCV 웹 UI SDK에서 DCVViewer React 구성 요소와 React 및 Cloudscape Design Global Styles 패키지를 가져옵니다.

```

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

```

다음은 Amazon DCV 서버에 대해 인증을 수행하고, 인증이 성공적으로 이루어진 경우 Amazon DCV 웹 UI SDK에서 DCVViewer React 구성 요소를 렌더링하는 방법을 보여주는 예제입니다.

```

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;

```

promptCredentials, error, success 함수는 인증 프로세스에서 정의해야 하는 필수 콜백 함수입니다.

Amazon DCV 서버에서 자격 증명을 요청하는 메시지가 표시되면 `promptCredentials` 콜백 함수는 Amazon DCV 서버로부터 요청된 자격 증명 챌린지를 수신합니다. Amazon DCV 서버가 시스템 인증을 사용하도록 구성된 경우 자격 증명을 사용자 이름 및 암호 형식으로 제공해야 합니다.

인증이 실패하면 `error` 콜백 함수는 Amazon DCV 서버로부터 오류 객체를 수신합니다.

인증에 성공하면 `success` 콜백 함수는 Amazon DCV 서버에서 사용자가 연결할 수 있는 각 세션의 세션 ID(`sessionId`) 및 권한 부여 토큰(`authToken`)이 함께 포함된 배열을 수신합니다. 위의 코드 샘플은 인증 성공 시 `DCVViewer` 구성 요소를 렌더링하도록 React 상태를 업데이트합니다.

이 구성 요소가 허용하는 속성에 대한 자세한 내용은 [Amazon DCV 웹 UI SDK 참조](#)를 참조하세요.

자체 서명된 인증서에 대한 자세한 내용은 [자체 서명된 인증서를 사용한 리디렉션 설명](#)을 참조하세요.

AWS-UI에서 Cloudscape Design System으로 업데이트

SDK 버전 1.3.0부터 `DCVViewer` 구성 요소가 AWS-UI에서 발전한 [Cloudscape Design](#)으로 업데이트되었습니다.

Cloudscape는 AWS-UI와 다른 시각적 테마를 사용하지만 기본 코드 기반은 동일하게 유지됩니다. 따라서 `DCVViewer`를 기반으로 애플리케이션을 쉽게 마이그레이션할 수 있습니다. 마이그레이션하려면 설치한 AWS-UI 관련 NPM 패키지를 관련 Cloudscape 패키지로 교체하세요.

AWS-UI 패키지 이름	Cloudscape 패키지 이름
<code>@awsui/components-react</code>	<code>@cloudscape-design/components</code>
<code>@awsui/global-styles</code>	<code>@cloudscape-design/global-styles</code>
<code>@awsui/collection-hooks</code>	<code>@cloudscape-design/collection-hooks</code>
<code>@awsui/design-tokens</code>	<code>@cloudscape-design/design-tokens</code>

마이그레이션에 대한 자세한 내용은 [AWS-UI GitHub 설명서 페이지](#)를 참조하세요.

SDK 참조

이 섹션에서는 Amazon DCV 웹 클라이언트 SDK에 대한 설명, 구문 및 사용 예제를 제공합니다.

주제

- [DCV 모듈](#)
- [연결 클래스](#)
- [인증 클래스](#)
- [리소스 클래스](#)
- [Amazon DCV 웹 UI SDK](#)

DCV 모듈

DCV 프로토콜의 클라이언트 측을 구현하는 모듈입니다.

노출

- [메서드](#)
- [Members](#)
- [유형 및 콜백 정의](#)

메서드

List

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

[인증\(authParams\) → {Authentication}](#)

지정된 Amazon DCV 서버 엔드포인트에 대한 인증 프로세스를 시작합니다.

파라미터:

이름	Type	설명
url	문자열	실행 중인 Amazon DCV 서버의 호스트 이름 및 포트 형식은 다음과 같습니다. <code>https://dcv_host_address:port</code> 예를 들어 <code>https://my-dcv-server:8443</code> 입니다.
authenticationToken	문자열	인증에 사용할 인증 토큰.
callbacks	authenticationCallbacks	인증 프로세스 중에 직접적으로 호출할 수 있는 콜백입니다.

반환 형식:

- Authentication 객체입니다.

Type

[Authentication](#)

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

지정된 Amazon DCV 서버 엔드포인트에 연결합니다. 연결이 성공하면 `Connection` 객체를 반환합니다. 연결에 실패하면 `error` 객체가 반환됩니다.

파라미터:

이름	Type	설명
config	ConnectionConfig	<code>ConnectionConfig</code> 객체입니다.

반환 형식:

- Connection 객체 또는 error 객체

Type

Promise.<[Connection](#)> | Promise.<{code: [ConnectionErrorCode](#), message: string}>

setLogHandler(handler) → {void}

사용자 지정 로그 핸들러 함수를 설정합니다. 기본 로그 핸들러를 재정의하면 브라우저 콘솔로 디버깅할 때 기존 로그 항목 위치가 손실됩니다.

파라미터:

이름	Type	설명
handler	함수	사용자 지정 로그 핸들러 함수입니다. 핸들러 함수에는 level(숫자), levelName(문자열), domain(문자열) 및 message(문자열)가 포함됩니다.

반환 형식:**Type**

void

setLogLevel(level) → {void}

로그 수준을 설정합니다. 이는 기본 로그 핸들러를 사용하는 경우에만 필요합니다.

파라미터:

이름	Type	설명
level	LogLevel	사용할 로그 수준입니다.

반환 형식:

Type

void

Members

List

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

(상수) AudioError :[AudioErrorCode](#)

AudioError 코드는 열거형입니다.

유형:

- [AudioErrorCode](#)

(상수) AuthenticationError : [AuthenticationErrorCode](#)

AuthenticationError 코드는 열거형입니다.

유형:

- [AuthenticationErrorCode](#)

(상수) ChannelError : [ChannelErrorCode](#)

ChannelError 코드는 열거형입니다.

유형:

- [ChannelErrorCode](#)

(상수) ClosingReasonError : [ClosingReasonErrorCode](#)

ClosingReasonError 코드는 열거형입니다.

유형:

- [ClosingReasonErrorCode](#)

(상수) ConnectionError : [ConnectionErrorCode](#)

ConnectionError 코드는 열거형입니다.

유형:

- [ConnectionErrorCode](#)

(상수) CustomChannelError : [CustomChannelErrorCode](#)

CustomChannelError 코드는 열거형입니다.

유형:

- [CustomChannelErrorCode](#)

(상수) DisplayConfigError : [DisplayConfigErrorCode](#)

DisplayConfigError 코드는 열거형입니다.

유형:

- [DisplayConfigErrorCode](#)

(상수) FileStorageError : [FileStorageErrorCode](#)

FileStorageError 코드는 열거형입니다.

유형:

- [FileStorageErrorCode](#)

(상수) LogLevel : [LogLevel](#)

사용 가능한 SDK 로그 수준입니다.

유형:

- [LogLevel](#)

(상수) MultiMonitorError : [MultiMonitorErrorCode](#)

MultiMonitorError 코드는 열거형입니다.

유형:

- [MultiMonitorErrorCode](#)

(상수) ResolutionError : [ResolutionErrorCode](#)

ResolutionError 코드는 열거형입니다.

유형:

- [ResolutionErrorCode](#)

(상수) TimezoneRedirectionError : [TimezoneRedirectionErrorCode](#)

TimezoneRedirectionError 코드는 열거형입니다.

유형:

- [TimezoneRedirectionErrorCode](#)

(상수) TimezoneRedirectionSetting : [TimezoneRedirectionSettingCode](#)

TimezoneRedirectionSetting 코드는 열거형입니다.

유형:

- [TimezoneRedirectionSettingCode](#)

(상수) TimezoneRedirectionStatus : [TimezoneRedirectionStatusCode](#)

TimezoneRedirectionStatus 코드는 열거형입니다.

유형:

- [TimezoneRedirectionStatusCode](#)

(상수) version

major, minor, patch, revision, extended, versionStr이 포함된 Amazon DCV 버전입니다.

속성:

이름	Type	설명
major	정수	메이저 버전 번호
minor	정수	마이너 버전 번호
patch	정수	패치 버전 번호
revision	정수	개정 번호

이름	Type	설명
extended	문자열	확장된 문자열
versionStr	문자열	메이저, 마이너, 패치 및 개정 번호를 major.minor.patch+build.revision 양식으로 연결

(상수) ScreenshotError : [ScreenshotErrorCode](#)

ScreenshotError 코드는 열거형입니다.

유형:

- [ScreenshotErrorCode](#)

(상수) WebcamError : [WebcamErrorCode](#)

WebcamError 코드는 열거형입니다.

유형:

- [WebcamErrorCode](#)

유형 및 콜백 정의

List

- [AudioErrorCode](#)
- [authenticationCallbacks](#)
- [AuthenticationErrorCode](#)
- [authErrorCallback\(authentication, error\)](#)
- [authPromptCredentialsCallback\(authentication, challenge\)](#)
- [authSuccessCallback\(authentication, authenticationData\)](#)
- [채널](#)
- [ChannelErrorCode](#)

- [clipboardEventCallback\(event\)](#)
- [ClosingReasonErrorCode](#)
- [Colorspace](#)
- [connectionCallbacks](#)
- [ConnectionConfig](#)
- [ConnectionErrorCode](#)
- [createDirectory\(path\)](#)
- [CustomChannelErrorCode](#)
- [dataChannelCallback\(info\)](#)
- [deleteFile\(path\)](#)
- [deviceChangeEventCallback\(\)](#)
- [disconnectCallback\(reason\)](#)
- [displayAvailabilityCallback\(status, displayId\)](#)
- [DisplayConfigErrorCode](#)
- [displayLayoutCallback\(serverWidth, serverHeight, heads\)](#)
- [feature](#)
- [featuresUpdateCallback\(featuresList\)](#)
- [fileDownloadCallback\(fileResource\)](#)
- [filePrintedCallback\(printResource\)](#)
- [filestorage](#)
- [filestorageEnabledCallback\(enabled\)](#)
- [FileStorageErrorCode](#)
- [firstFrameCallback\(resizeEnabled, relativeMouseModeEnabled, displayId\)](#)
- [idleWarningNotificationCallback\(disconnectionDateTime\)](#)
- [collaboratorListCallback\(collaborators\)](#)
- [licenseNotificationCallback\(notification\)](#)
- [list\(path\)](#)
- [LogLevel](#)
- [모니터링](#)
- [MultiMonitorErrorCode](#)

- [qualityIndicatorStateCallback\(state\)](#)
- [renameDirectory\(src, dest\)](#)
- [renameFile\(src, dest\)](#)
- [ResolutionErrorCode](#)
- [retrieveFile\(path\)](#)
- [screenshotCallback\(screenshot\)](#)
- [ScreenshotErrorCode](#)
- [serverInfo](#)
- [통계](#)
- [storeFile\(file, dir\)](#)
- [TimezoneRedirectionErrorCode](#)
- [TimezoneRedirectionSettingCode](#)
- [TimezoneRedirectionStatusCode](#)
- [WebcamErrorCode](#)
- [httpExtraSearchParamsCallback\(메서드, URL, 본문\)](#)
- [httpExtraHeadersCallback\(메서드, URL, 본문\)](#)

AudioErrorCode

DCV 모듈에서 사용할 수 있는 AudioError 코드 열거형

- SETTING_AUDIO_FAILED
- CHANNEL_NOT_AVAILABLE

유형:

- number

authenticationCallbacks

인증 콜백

유형:

- 객체

속성:

이름	Type	설명
promptCredentials	authPromptCredentialsCallback	사용자에게 자격 증명이 요청될 때 직접적으로 호출되는 콜백 함수입니다.
error	authErrorCallback	인증 실패 시 직접적으로 호출되는 콜백 함수입니다.
success	authSuccessCallback	인증 성공 시 직접적으로 호출되는 콜백 함수입니다.
httpExtraSearchParamsCallback	httpExtraSearchParamsCallback	요청을 시작하기 전에 인증 URL에 사용자 지정 쿼리 파라미터를 주입하기 위해 <code>authenticate</code> 메서드에서 직접적으로 호출할 콜백 함수입니다. <code>connect</code> 메서드에서 DCV 서버에 대한 WebSocket 연결을 설정할 때 사용되는 URL을 사용자 지정하는 데 사용할 수도 있습니다.

AuthenticationErrorCode

DCV 모듈에서 사용할 수 있는 AuthenticationError 코드 열거형

- 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

유형:

- number

`authErrorCallback(authentication, error)`

인증 실패 시 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명									
<code>authentication</code>	Authentication	Authentication 객체입니다.									
<code>error</code>	객체	인증 프로세스에서 발생한 error 객체입니다. <table border="1" data-bbox="1068 1310 1507 1743"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td><code>code</code></td> <td>AuthenticationError rCode</td> <td>오류 코드입니다.</td> </tr> <tr> <td><code>message</code></td> <td>문자열</td> <td>오류 메시지입니다.</td> </tr> </tbody> </table>	이름	Type	설명	<code>code</code>	AuthenticationError rCode	오류 코드입니다.	<code>message</code>	문자열	오류 메시지입니다.
이름	Type	설명									
<code>code</code>	AuthenticationError rCode	오류 코드입니다.									
<code>message</code>	문자열	오류 메시지입니다.									

authPromptCredentialsCallback(authentication, challenge)

사용자에게 자격 증명이 요청될 때 직접적으로 호출되는 콜백 함수입니다. 사용자는 요청된 자격 증명을 제공하여 챌린지에 답해야 합니다.

파라미터:

이름	Type	설명						
authentication	Authentication	Authentication 객체입니다.						
challenge	객체	<p>챌린지입니다.</p> <table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>requiredCredentials</td> <td>Array.<Object></td> <td>요청된 자격 증명 객체의 배열입니다.</td> </tr> </tbody> </table>	이름	Type	설명	requiredCredentials	Array.<Object>	요청된 자격 증명 객체의 배열입니다.
이름	Type	설명						
requiredCredentials	Array.<Object>	요청된 자격 증명 객체의 배열입니다.						

이름	Type	설명									
		<table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>sessionID</td> <td>문자열</td> <td>Amazon DCV 세션 ID입니다.</td> </tr> <tr> <td>authToken</td> <td>문자열</td> <td>Amazon DCV 세션의 인증 토큰입니다.</td> </tr> </tbody> </table>	이름	Type	설명	sessionID	문자열	Amazon DCV 세션 ID입니다.	authToken	문자열	Amazon DCV 세션의 인증 토큰입니다.
이름	Type	설명									
sessionID	문자열	Amazon DCV 세션 ID입니다.									
authToken	문자열	Amazon DCV 세션의 인증 토큰입니다.									

채널

지정할 수 있는 사용 가능한 채널입니다.

유형:

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

ChannelErrorCode

DCV 모듈에서 사용할 수 있는 ChannelError 코드 열거형

- ALREADY_OPEN
- INITIALIZATION_FAILED
- REJECTED

유형:

- number

clipboardEventCallback(event)

clipboardEvent가 생성될 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명												
event	객체	클립보드 이벤트에 대한 정보입니다.												
		<table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>속성</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>established copy paste dataSiltert autoCopy newDataAvailable autoPaste Done remote paste/labelD</td> <td></td> <td>항상 존재합니다. 이벤트의 이름입니다.</td> </tr> <tr> <td>clipboardData</td> <td>Object string</td> <td></td> <td>클립보드의</td> </tr> </tbody> </table>	이름	Type	속성	설명	name	established copy paste dataSiltert autoCopy newDataAvailable autoPaste Done remote paste/labelD		항상 존재합니다. 이벤트의 이름입니다.	clipboardData	Object string		클립보드의
이름	Type	속성	설명											
name	established copy paste dataSiltert autoCopy newDataAvailable autoPaste Done remote paste/labelD		항상 존재합니다. 이벤트의 이름입니다.											
clipboardData	Object string		클립보드의											

이름	Type	설명			
		이름	Type	속성	설명
					데 이터 입니 다.
		autoC	부울	<optio >	세션 클립 보드 에서 로컬 클라 이언 트 클립 보드 로의 자동 복사 가 활성 화되 는지 여부 를 나타 냅니 다.
		maxDa ze	numbe	<optio >	클립 보드 에 넣을 수 있는

이름	Type	설명			
		이름	Type	속성	설명
					최대 데이터 양입니다.
		error	문자열	<optional>	오류 정보입니다 (해당하는 경우).

ClosingReasonErrorCode

DCV 모듈에서 사용할 수 있는 ClosingReasonError 열거형

- 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

유형:

- number

Colorspace

지정할 수 있는 사용 가능한 색상 공간입니다.

유형:

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

connectionCallbacks

연결 오류 발생 시 직접적으로 호출할 수 있는 콜백입니다.

유형:

- 객체

속성:

이름	Type	설명
disconnect	disconnectCallback	연결이 종료될 때 직접적으로 호출되는 콜백 함수입니다.
displayLayout	displayLayoutCallback	디스플레이 레이아웃 또는 해상도가 변경될 때 직접적으로 호출되는 콜백 함수입니다.
displayAvailability	displayAvailabilityCallback	디스플레이 가용성이 변경될 때 직접적으로 호출되는 콜백 함수입니다.

이름	Type	설명
firstFrame	firstFrameCallback	Amazon DCV 서버에서 첫 프레임을 수신하면 콜백 함수가 직접적으로 호출됩니다.
filePrinted	filePrintedCallback	Amazon DCV 서버에 파일이 인쇄될 때 직접적으로 호출되는 콜백 함수입니다.
fileDownload	fileDownloadCallback	Amazon DCV 서버에서 파일을 다운로드할 준비가 완료되면 직접적으로 호출되는 콜백 함수입니다.
dataChannel	dataChannelCallback	Amazon DCV 서버가 데이터 채널의 가용성에 대한 알림을 보낼 때 직접적으로 호출되는 콜백 함수입니다.
licenseNotification	licenseNotificationCallback	Amazon DCV 서버가 라이선스 상태에 대한 알림을 보낼 때 직접적으로 호출되는 콜백 함수입니다.
idleWarningNotification	idleWarningNotificationCallback	Amazon DCV 서버가 유휴 타임아웃 경고를 보낼 때 직접적으로 호출되는 콜백 함수입니다.
collaboratorList	collaboratorListCallback	Amazon DCV 서버가 공동 작업자 목록을 전송할 때 직접적으로 호출되는 콜백 함수입니다(Amazon DCV 웹 클라이언트 SDK 버전 1.1.0부터).

이름	Type	설명
qualityIndicatorState	qualityIndicatorStateCallback	연결 품질 표시기의 상태가 변경될 때 직접적으로 호출되는 콜백 함수입니다.
filestorageEnabled	filestorageEnabledCallback	파일 스토리지가 활성화되거나 비활성화될 때 직접적으로 호출되는 콜백 함수입니다.
featuresUpdate	featuresUpdateCallback	기능 상태가 변경될 때 직접적으로 호출되는 콜백 함수입니다.
clipboardEvent	clipboardEventCallback	clipboardEvent 가 생성될 때 직접적으로 호출되는 콜백 함수입니다.
deviceChangeEvent	deviceChangeEventCallback	deviceChange 이벤트가 트리거될 때 직접적으로 호출되는 콜백 함수입니다.
screenshot	screenshotCallback	screenshot 을 사용할 수 있을 때 직접적으로 호출되는 콜백 함수입니다.
httpExtraSearchParamsCallback	httpExtraSearchParamsCallback	Amazon DCV 서버에 대한 WebSocket 연결을 설정할 때 URL을 사용자 지정하기 위해 직접적으로 호출할 콜백 함수입니다. SDK가 요청을 보내기 전에 이 콜백을 authenticate 메서드와 함께 사용하여 인증 URL에 쿼리 파라미터를 동적으로 추가할 수도 있습니다.

이름	Type	설명
httpExtraHeadersCallback	httpExtraHeadersCallback	연결 설정 중에 HTTP 요청에 사용자 지정 헤더를 추가하기 위해 직접적으로 호출할 콜백 함수입니다.

ConnectionConfig

Amazon DCV 연결을 위한 구성입니다.

유형:

- 객체

속성:

이름	Type	설명
url	문자열	실행 중인 Amazon DCV 서버의 호스트 이름 및 포트 형식은 다음과 같습니다. <code>https://dcv_host_address:port</code> 예를 들어 <code>https://my-dcv-server:8443</code> 입니다.
sessionId	문자열	Amazon DCV 세션 ID입니다.
authToken	문자열	서버에 연결할 때 사용할 인증 토큰입니다.
baseUrl	문자열	SDK 파일을 로드할 절대 또는 상대 URL입니다.
resourceBaseUrl	문자열	DCV 리소스에 액세스할 수 있는 절대 또는 상대 URL입니다.

이름	Type	설명
enabledChannels	Array.< Channel >	활성화할 수 있는 채널 목록을 나타냅니다. 지정되지 않거나 빈 배열이 제공되면 사용 가능한 모든 채널이 기본값으로 설정됩니다.
losslessColorspace	Colorspace	사용할 색상 공간을 나타냅니다. 지정하지 않으면 기본적으로 "RGB"로 설정됩니다.
divId	문자열	SDK가 원격 스트림으로 캔버스를 생성해야 하는 HTML DOM의 div 객체 ID입니다.
volumeLevel	정수	선호하는 볼륨 수준입니다. 값의 범위는 0~100입니다.
clipboardAutoSync	부울	호환되는 웹 브라우저에 대해 Amazon DCV 세션 클립보드에서 로컬 클라이언트 클립보드의 자동 복사를 활성화할지 여부를 나타냅니다.
dynamicAudioTuning	부울	연결이 설정되었을 때 Amazon DCV 서버 오디오 설정을 기반으로 오디오를 동적으로 조정할지 여부를 나타냅니다.
clientHiDpiScaling	부울	클라이언트의 DPI에 따라 캔버스 크기를 조정할지 여부를 나타냅니다.

이름	Type	설명
highColorAccuracy	부울	가능한 경우 높은 색상 정확도를 사용해야 하는지 여부를 나타냅니다. 지정하지 않으면 기본적으로 false로 설정됩니다.
enableWebCodecs	부울	가능한 경우 WebCodecs를 사용해야 하는지 여부를 나타냅니다. 지정하지 않으면 기본적으로 false로 설정됩니다.
observers	connectionCallbacks	연결과 관련된 이벤트를 직접적으로 호출하는 콜백 함수입니다.
callbacks	connectionCallbacks	observers 속성과 동일하지만 각 콜백에는 Connection 객체가 첫 번째 파라미터로 포함됩니다.

ConnectionErrorCode

DCV 모듈에서 사용할 수 있는 ConnectionError 코드 열거형

- ALREADY_OPEN
- INVALID_CONFIG
- INITIALIZATION_FAILED
- REJECTED
- MAIN_CHANNEL_ALREADY_OPEN
- GENERIC_ERROR (DCV 서버 2021.0부터)
- INTERNAL_SERVER_ERROR (DCV 서버 2021.0부터)
- AUTHENTICATION_FAILED (DCV 서버 2021.0부터)
- PROTOCOL_ERROR (DCV 서버 2021.0부터)
- INVALID_SESSION_ID (DCV 서버 2021.0부터)
- INVALID_CONNECTION_ID (DCV 서버 2021.0부터)

- CONNECTION_LIMIT_REACHED (DCV 서버 2021.0부터)
- SERVER_UNREACHABLE (DCV 서버 2022.1부터)
- GATEWAY_BUSY
- UNSUPPORTED_CREDENTIAL(DCV 서버 2022.2부터)
- TRANSPORT_ERROR

유형:

- number

createDirectory(path)

파라미터:

이름	Type	설명
path	문자열	디렉토리를 만들려는 서버의 절대 경로입니다. 대상 디렉터리의 이름도 포함해야 합니다.

CustomChannelErrorCode

DCV 모듈에서 사용할 수 있는 CustomChannelError 코드 열거형

- TRANSPORT_ERROR

유형:

- number

dataChannelCallback(info)

Amazon DCV 서버가 데이터 채널의 가용성에 대한 알림을 보낼 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명									
info	객체	데이터 채널에 대한 정보입니다.									
		<table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>문자열</td> <td>데이터 채널의 이름입니다.</td> </tr> <tr> <td>token</td> <td>문자열</td> <td>데이터 채널의 인증 토큰입니다.</td> </tr> </tbody> </table>	이름	Type	설명	name	문자열	데이터 채널의 이름입니다.	token	문자열	데이터 채널의 인증 토큰입니다.
이름	Type	설명									
name	문자열	데이터 채널의 이름입니다.									
token	문자열	데이터 채널의 인증 토큰입니다.									

deleteFile(path)

파라미터:

이름	Type	설명
path	문자열	삭제하려는 파일을 식별하는 서버의 절대 경로입니다.

deviceChangeEventCallback()

deviceChange 이벤트가 트리거될 때 직접적으로 호출되는 콜백 함수입니다.

disconnectCallback(reason)

연결이 종료될 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명									
reason	객체	연결이 해제된 이유입니다.									
		<table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>code</td> <td>number</td> <td>이유 코드입니다.</td> </tr> <tr> <td>message</td> <td>문자열</td> <td>이유 메시지입니다.</td> </tr> </tbody> </table>	이름	Type	설명	code	number	이유 코드입니다.	message	문자열	이유 메시지입니다.
이름	Type	설명									
code	number	이유 코드입니다.									
message	문자열	이유 메시지입니다.									

displayAvailabilityCallback(status, displayId)

디스플레이 가용성이 변경될 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명						
status	객체	디스플레이의 상태입니다.						
		<table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>enabled</td> <td>부울</td> <td>디스플레이가 활성화되어 있는지 여부를 나타냅니다.</td> </tr> </tbody> </table>	이름	Type	설명	enabled	부울	디스플레이가 활성화되어 있는지 여부를 나타냅니다.
이름	Type	설명						
enabled	부울	디스플레이가 활성화되어 있는지 여부를 나타냅니다.						

이름	Type	설명						
		<table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>closed</td> <td>부울</td> <td>디스플레이가 종료되었는지 여부를 나타냅니다.</td> </tr> </tbody> </table>	이름	Type	설명	closed	부울	디스플레이가 종료되었는지 여부를 나타냅니다.
이름	Type	설명						
closed	부울	디스플레이가 종료되었는지 여부를 나타냅니다.						
displayId	number	디스플레이에 대한 식별자입니다.						

DisplayConfigErrorCode

DCV 모듈에서 사용할 수 있는 DisplayConfigError 코드 열거형

- INVALID_ARGUMENT
- UNSUPPORTED_OPERATION
- NO_CHANNEL

유형:

- number

displayLayoutCallback(serverWidth, serverHeight, heads)

디스플레이 레이아웃 또는 해상도가 변경될 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명
serverWidth	number	기본 디스플레이의 너비(픽셀 단위)입니다.

이름	Type	설명
serverHeight	number	기본 디스플레이의 높이(픽셀 단위)입니다.
heads	Array.< Monitor >	Amazon DCV 서버에서 지원하는 디스플레이 헤드입니다.

feature

기능 값입니다.

- display - 단일 디스플레이 비디오 스트림의 사용 가능 여부를 나타냅니다.
- display-multi - 다중 디스플레이 비디오 스트림의 사용 가능 여부를 나타냅니다.
- high-color-accuracy - 높은 색상 정확도의 사용 가능 여부를 나타냅니다(Amazon DCV 웹 클라이언트 SDK 버전 1.1.0부터).
- mouse - 마우스 기능의 사용 가능 여부를 나타냅니다.
- keyboard - 키보드 기능의 사용 가능 여부를 나타냅니다.
- keyboard-sas - SAS 시퀀스(Control + Alt + Delete) 기능의 사용 가능 여부를 나타냅니다.
- relative-mouse - 상대 마우스 모드의 사용 가능 여부를 나타냅니다.
- clipboard-copy - Amazon DCV 서버에서 클라이언트로의 클립보드 복사 기능 사용 가능 여부를 나타냅니다.
- clipboard-paste - 클라이언트에서 Amazon DCV 서버로의 클립보드 복사 기능 사용 가능 여부를 나타냅니다.
- audio-in - 마이크를 사용한 오디오 입력 기능의 사용 가능 여부를 나타냅니다.
- audio-out - 오디오 재생 기능의 사용 가능 여부를 나타냅니다.
- webcam - 웹캠 스트리밍 기능의 사용 가능 여부를 나타냅니다.
- file-download - Amazon DCV 서버에서 클라이언트로의 파일 다운로드 기능 사용 가능 여부를 나타냅니다.
- file-upload - 클라이언트에서 Amazon DCV 서버로의 파일 업로드 기능 사용 가능 여부를 나타냅니다.
- timezone-redirect - 시간대 리디렉션 기능의 사용 가능 여부를 나타냅니다(Amazon DCV 웹 클라이언트 SDK 버전 1.3.0부터).

유형:

- 문자열

featuresUpdateCallback(featuresList)

기능 상태가 변경될 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명
featuresList	Array.< feature >	변경된 기능의 목록입니다.

fileDownloadCallback(fileResource)

Amazon DCV 서버에서 파일을 다운로드할 준비가 완료되면 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명												
fileResource	객체	다운로드할 준비가 된 파일에 대한 정보입니다. <table border="1" data-bbox="1068 1136 1507 1879"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>문자열</td> <td>파일의 식별자입니다.</td> </tr> <tr> <td>url</td> <td>문자열</td> <td>파일을 다운로드할 때 사용할 URL입니다.</td> </tr> <tr> <td>domain</td> <td>문자열</td> <td>리소스 도메인입니다.</td> </tr> </tbody> </table>	이름	Type	설명	id	문자열	파일의 식별자입니다.	url	문자열	파일을 다운로드할 때 사용할 URL입니다.	domain	문자열	리소스 도메인입니다.
이름	Type	설명												
id	문자열	파일의 식별자입니다.												
url	문자열	파일을 다운로드할 때 사용할 URL입니다.												
domain	문자열	리소스 도메인입니다.												

이름	Type	설명		
		이름	Type	설명
		token	문자열	파일을 다운로드하는 데 사용할 인증 토큰입니다. 토큰은 URL에도 포함되어 있습니다.

filePrintedCallback(printResource)

Amazon DCV 서버에 파일이 인쇄될 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명		
printResource	객체	인쇄된 파일에 대한 정보입니다.		
		이름	Type	설명
		id	문자열	인쇄된 파일의 식별자입니다.
		url	문자열	인쇄된 파일을

이름	Type	설명		
		이름	Type	설명
				다운로드할 때 사용할 URL입니다.
		domain	문자열	리소스 도메인입니다. 이 경우 printer입니다.
		token	문자열	인쇄된 파일을 다운로드하는 데 사용할 인증 토큰입니다. 토큰은 URL에도 포함되어 있습니다.

filestorage

파일 시스템에서 작업을 탐색하고 수행할 수 있는 객체입니다.

유형:

- 객체

속성:

이름	Type	설명
list	list	서버의 제공된 경로에 있는 항목(파일 및 디렉터리)을 나열할 수 있는 함수입니다.
createDirectory	createDirectory	서버의 지정된 경로에 디렉터리를 생성할 수 있는 함수입니다.
retrieveFile	retrieveFile	서버의 지정된 경로에 있는 파일을 로컬로 다운로드할 수 있는 함수입니다.
deleteFile	deleteFile	서버의 지정된 경로에 있는 파일을 삭제할 수 있는 함수입니다.
renameFile	renameFile	지정된 소스 경로에서 지정된 대상 경로로 파일 이름을 바꿀 수 있는 함수입니다.
renameDirectory	renameDirectory	지정된 소스 경로에서 절대 대상 경로로 디렉터리 이름을 바꿀 수 있는 함수입니다.
storeFile	storeFile	서버의 제공된 경로에 로컬 파일을 업로드할 수 있는 함수입니다.

filestorageEnabledCallback(enabled)

파일 스토리지가 활성화될 때 직접적으로 호출되는 콜백 함수입니다. Internet Explorer 11의 지연 채널에만 해당합니다.

파라미터:

이름	Type	설명
enabled	부울	파일 스토리지를 활성화할지 여부를 나타냅니다.

FileStorageErrorCode

DCV 모듈에서 사용할 수 있는 FileStorageError 코드 열거형

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

유형:

- number

firstFrameCallback(resizeEnabled, relativeMouseModeEnabled, displayId)

Amazon DCV 서버에서 첫 프레임을 수신하면 콜백 함수가 직접적으로 호출됩니다. 각 디스플레이마다 발생합니다.

파라미터:

이름	Type	설명
resizeEnabled	부울	서버가 클라이언트 디스플레이 레이아웃 크기 조정을 지원하는지 여부를 나타냅니다.

이름	Type	설명
relativeMouseModeEnabled	부울	서버가 상대 마우스 모드를 지원하는지 여부를 나타냅니다.
displayId	number	디스플레이에 대한 식별자입니다.

idleWarningNotificationCallback(disconnectionDateTime)

Amazon DCV 서버가 유휴 타임아웃 경고를 보낼 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명
disconnectionDateTime	Date	연결 해제 날짜 및 시간입니다.

collaboratorListCallback(collaborators)

Amazon DCV 서버가 공동 작업자 목록을 전송할 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명
collaborators	Array.<Object>	공동 작업자에 대한 정보가 포함된 객체 목록입니다.

이름	Type	설명
username	문자열	공동 작업자의 사용자 이름입니다.

이름	Type	설명									
		<table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>owner</td> <td>부울</td> <td>공동 작업자가 세션 소유자인지 여부를 나타냅니다.</td> </tr> <tr> <td>connect number nId</td> <td></td> <td>서버에서 연결에 할당된 ID를 나타냅니다.</td> </tr> </tbody> </table>	이름	Type	설명	owner	부울	공동 작업자가 세션 소유자인지 여부를 나타냅니다.	connect number nId		서버에서 연결에 할당된 ID를 나타냅니다.
이름	Type	설명									
owner	부울	공동 작업자가 세션 소유자인지 여부를 나타냅니다.									
connect number nId		서버에서 연결에 할당된 ID를 나타냅니다.									

licenseNotificationCallback(notification)

Amazon DCV 서버가 라이선스 상태에 대한 알림을 보낼 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명									
notification	객체	알림입니다. <table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>product</td> <td>문자열</td> <td>DCV 제품입니다.</td> </tr> <tr> <td>status</td> <td>문자열</td> <td>라이선스의 상</td> </tr> </tbody> </table>	이름	Type	설명	product	문자열	DCV 제품입니다.	status	문자열	라이선스의 상
이름	Type	설명									
product	문자열	DCV 제품입니다.									
status	문자열	라이선스의 상									

이름	Type	설명
		이름
		Type
		설명
		태입니 다.
	message	문자열 메시지 입니다.
	leftDay	number 라이선 스가 만 료되기 까지 남 은 일 수.
	isDemo	부울 라이선 스가 데 모 라이 선스인 지 여부 를 나타 냅니다.
	numUnli nsed	number 라이선 스가 없 는 연결 수입니 다.
	licensi Mode	문자열 라이선 스 모드 입니다.
	documer tionUr]	문자열 설명서 의 URL 입니다.

list(path)

파라미터:

이름	Type	설명
path	문자열	콘텐츠를 나열하려는 서버의 절대 경로입니다.

LogLevel

사용 가능한 SDK 로그 수준입니다.

유형:

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

모니터링

유형:

- 객체

속성:

이름	Type	설명						
name	문자열	디스플레이 헤드의 이름입니다.						
rect	객체	디스플레이 헤드에 대한 정보입니다. <table border="1" data-bbox="1068 1696 1507 1881"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>x</td> <td>number</td> <td>디스플레이 헤</td> </tr> </tbody> </table>	이름	Type	설명	x	number	디스플레이 헤
이름	Type	설명						
x	number	디스플레이 헤						

이름	Type	설명		
		이름	Type	설명
				드의 초기 x 좌표입니다.
		y	number	디스플레이 헤드의 초기 y 좌표입니다.
		width	number	디스플레이 헤드의 너비(픽셀 단위)입니다.
		height	number	디스플레이 헤드의 높이(픽셀 단위)입니다.
primary	부울	디스플레이 헤드가 기본 디스플레이 헤드인지 여부를 나타냅니다. 가능한 경우 원격 운영 체제에서 결정됩니다.		
dpi	number	디스플레이 헤드의 DPI입니다.		

MultiMonitorErrorCode

DCV 모듈에서 사용할 수 있는 MultiMonitorError 코드 열거형

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

유형:

- number

qualityIndicatorStateCallback(state)

연결 품질 표시기의 상태가 변경될 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명									
state	Array.<Object>	연결 품질에 대한 정보입니다.									
		<table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>문자열</td> <td>지표의 이름입니다.</td> </tr> <tr> <td>status</td> <td>NORMAL WARNIN CRITICA</td> <td>상태에 대한 설명입니다.</td> </tr> </tbody> </table>	이름	Type	설명	name	문자열	지표의 이름입니다.	status	NORMAL WARNIN CRITICA	상태에 대한 설명입니다.
이름	Type	설명									
name	문자열	지표의 이름입니다.									
status	NORMAL WARNIN CRITICA	상태에 대한 설명입니다.									

이름	Type	설명		
		이름	Type	설명
		changece	부울	상태가 변경되었는지 여부를 나타냅니다.

renameDirectory(src, dest)

파라미터:

이름	Type	설명
src	문자열	이름을 변경하려는 디렉터리를 식별하는 서버의 절대 소스 경로입니다.
dest	문자열	대상 경로와 디렉터리 이름을 지정하는 서버의 절대 대상 경로입니다.

renameFile(src, dest)

파라미터:

이름	Type	설명
src	문자열	이름을 변경하려는 파일을 식별하는 서버의 절대 소스 경로입니다.

이름	Type	설명
dest	문자열	대상 경로와 파일 이름을 지정하는 서버의 절대 대상 경로입니다.

ResolutionErrorCode

DCV 모듈에서 사용할 수 있는 ResolutionError 코드 열거형

- INVALID_ARGUMENT
- NO_CHANNEL
- NOT_IMPLEMENTED

유형:

- number

retrieveFile(path)

파라미터:

이름	Type	설명
path	문자열	로컬로 다운로드하려는 파일을 식별하는 서버의 절대 경로입니다.

screenshotCallback(screenshot)

스크린샷을 사용할 수 있을 때 직접적으로 호출되는 콜백 함수입니다.

파라미터:

이름	Type	설명
screenshot	byte[]	PNG 형식의 스크린샷 버퍼이거나 스크린샷 검색이 실패한 경우 null입니다.

ScreenshotErrorCode

DCV 모듈에서 사용 가능한 ScreenshotError 코드 열거형

- NO_CHANNEL
- GENERIC_ERROR

유형:

- number

serverInfo

유형:

- 객체

속성:

이름	Type	설명
name	문자열	소프트웨어의 이름입니다.
version	객체	소프트웨어의 버전 번호입니다.

이름	Type	설명												
		<table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>major</td> <td>number</td> <td>메이저 버전 번호입니다.</td> </tr> <tr> <td>minor</td> <td>number</td> <td>마이너 버전 번호입니다.</td> </tr> <tr> <td>revisio</td> <td>number</td> <td>개정 버전 번호입니다.</td> </tr> </tbody> </table>	이름	Type	설명	major	number	메이저 버전 번호입니다.	minor	number	마이너 버전 번호입니다.	revisio	number	개정 버전 번호입니다.
이름	Type	설명												
major	number	메이저 버전 번호입니다.												
minor	number	마이너 버전 번호입니다.												
revisio	number	개정 버전 번호입니다.												
os	문자열	운영 체제입니다.												
arch	문자열	아키텍처입니다.												
hostname	문자열	호스트 이름입니다.												

통계

유형:

- 객체

속성:

이름	Type	설명
fps	number	현재 초당 프레임입니다.
traffic	number	현재 트래픽(비트/초)입니다.

이름	Type	설명
peakTraffic	number	연결이 설정된 이후 최대 트래픽(비트/초)입니다.
latency	number	현재 지연 시간(밀리초)입니다.
currentChannels	number	연결이 설정된 이후 열려 있는 채널 수입니다.
openedChannels	number	현재 열려 있는 채널 수입니다.
channelErrors	number	오류가 보고된 채널 수입니다.

storeFile(file, dir)

파라미터:

이름	Type	설명
file	파일	서버에 업로드하려는 파일 객체입니다(자세한 내용은 https://developer.mozilla.org/en-US/docs/Web/API/File 참조).
dir	문자열	파일을 업로드하려는 서버의 절대 경로입니다.

TimezoneRedirectionErrorCode

DCV 모듈에서 사용할 수 있는 TimezoneRedirectionError 코드 열거형

- INVALID_ARGUMENT
- NO_CHANNEL
- USER_CANNOT_CHANGE

유형:

- number

TimezoneRedirectionSettingCode

DCV 모듈에서 사용할 수 있는 TimezoneRedirectionSetting 코드 열거형

- ALWAYS_OFF
- ALWAYS_ON
- CLIENT_DECIDES

유형:

- number

TimezoneRedirectionStatusCode

DCV 모듈에서 사용할 수 있는 TimezoneRedirectionStatus 코드 열거형

- SUCCESS
- PERMISSION_ERROR
- GENERIC_ERROR

유형:

- number

WebcamErrorCode

DCV 모듈에서 사용할 수 있는 WebcamError 코드 열거형

- SETTING_WEBCAM_FAILED
- CHANNEL_NOT_AVAILABLE

유형:

- number

httpExtraSearchParamsCallback(메서드, URL, 본문)

인증 및 연결 설정 중에 URL에 사용자 지정 쿼리 파라미터를 주입하기 위해 직접적으로 호출되는 콜백 함수입니다. 이를 통해 사용자 지정 쿼리 파라미터를 추가하고 AWS 서명 버전 4(SigV4) 서명 값을 추가하여 외부 시스템을 통한 연결을 보호하고 승인하는 기능을 포함한 고급 통합 시나리오를 사용할 수 있습니다.

이 콜백은 Amazon DCV 서버에 대한 WebSocket 연결을 설정할 때 사용되는 URL을 사용자 지정하는 데에도 사용됩니다.

파라미터:

이름	Type	설명
method	문자열	요청에 사용되는 HTTP 메서드입니다.
url	문자열	요청에 사용할 URL입니다.
body	문자열	요청 본문의 내용

반환 형식:

URL에 추가할 사용자 지정 쿼리 파라미터가 포함된 `URLSearchParams` 객체입니다.

Type

`URLSearchParams`

httpExtraHeadersCallback(메서드, URL, 본문)

연결 설정 중에 HTTP 요청에 사용자 지정 헤더(예: Authorization)를 삽입하기 위해 직접적으로 호출할 콜백 함수입니다.

파라미터:

이름	Type	설명
method	문자열	요청에 사용되는 HTTP 메서드입니다.
url	문자열	요청에 사용할 URL입니다.
body	문자열	요청 본문의 내용

반환 형식:

HTTP 요청에 추가할 사용자 지정 헤더를 나타내는 키-값 페어가 포함된 객체입니다.

Type

객체

연결 클래스

dcv 모듈의 [connect 메서드](#)를 직접적으로 호출하여 얻은 연결 클래스입니다. 사용 방법을 보여주는 예제는 [시작하기](#) 섹션을 참조하세요.

노출

- [메서드](#)

메서드

List

- [attachDisplay\(win, displayConf\) → {Promise.<number>|Promise.<{code: MultiMonitorErrorCode, message: string}>}](#)
- [captureClipboardEvents\(enabled, win, displayId\) → {void}](#)
- [detachDisplay\(displayId\) → {void}](#)
- [disconnect\(\) → {void}](#)
- [disconnectCollaborator\(connectionId\) → {void}](#)

- [enableDisplayQualityUpdates\(enable\) → {void}](#)
- [enableHighPixelDensity\(enable\) → {void}](#)
- [enableTimezoneRedirection\(enable\) → {Promise|Promise.<{code: TimezoneRedirectionErrorCode, message: string}>}](#)
- [enterRelativeMouseMode\(\) → {void}](#)
- [getConnectedDevices\(\) → {Promise.<Array.<MediaDeviceInfo>>|Promise.<{message: string}>}](#)
- [getFileExplorer\(\) → {Promise.<filestorage>|Promise.<{code: ChannelErrorCode, message: string}>}](#)
- [getServerInfo\(\) → {serverInfo}](#)
- [getScreenshot\(\) → {Promise|Promise.<{code: ScreenshotErrorCode, message: string}>}](#)
- [getStats\(\) → {stats}](#)
- [latchModifierKey\(key, location, isDown\) → {boolean}](#)
- [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\(shortcuts\) → {void}](#)
- [requestDisplayConfig\(highColorAccuracy\) → {Promise|Promise.<{code: DisplayConfigErrorCode, message: string}>}](#)
- [requestDisplayLayout\(layout\) → {Promise|Promise.<{code: ResolutionErrorCode, message: string}>}](#)
- [requestResolution\(width, height\) → {Promise|Promise.<{code: ResolutionErrorCode, message: string}>}](#)
- [sendKeyboardEvent\(event\) → {boolean}](#)
- [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\(enable\) → {Promise|Promise.<{code: AudioErrorCode, message: string}>}](#)
- [setMinDisplayResolution\(minWidth, minHeight\) → {void}](#)

- [setUpUploadBandwidth\(value\) → {number}](#)
- [setVolume\(volume\) → {void}](#)
- [setMicrophone\(enable, deviceId\) → {Promise|Promise.<{code: AudioErrorCode, message: string}>}](#)
- [setWebcam\(enable, deviceId\) → {Promise|Promise.<{code: WebcamErrorCode, message: string}>}](#)
- [syncClipboards\(\) → {boolean}](#)

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

특정 디스플레이를 창에 연결합니다. 메인 디스플레이는 연결할 수 없습니다. 성공할 경우 함수는 `displayId`를 반환합니다.

파라미터:

이름	Type	설명												
win	객체	디스플레이가 연결되어야 하는 창입니다.												
displayConf	객체	디스플레이의 구성입니다. <table border="1" data-bbox="1068 1186 1507 1881"> <thead> <tr> <th>이름</th> <th>Type</th> <th>속성</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>displayId</td> <td>number</td> <td><optional></td> <td>디스플레이의 ID입니다.</td> </tr> <tr> <td>displayName</td> <td></td> <td></td> <td>디스플레이의 이름</td> </tr> </tbody> </table>	이름	Type	속성	설명	displayId	number	<optional>	디스플레이의 ID입니다.	displayName			디스플레이의 이름
이름	Type	속성	설명											
displayId	number	<optional>	디스플레이의 ID입니다.											
displayName			디스플레이의 이름											

이름	Type	설명			
		이름	Type	속성	설명
					입니 다.

반환 형식:

Promise를 반환합니다. 거부되면 promise는 오류 객체를 반환합니다.

Type

Promise.<number> | Promise.<{code: [MultiMonitorErrorCode](#), message: string}>

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

copy-paste 이벤트 수신을 시작하거나 중지합니다. 대화형 클립보드의 경우(항상 붙여넣기인 경우) copy/paste 이벤트 수신을 시작해야 합니다. 필요한 경우에만 수신을 시작하고 중지하는 것이 유용할 수 있습니다(예: 모달이 표시될 때).

파라미터:

이름	Type	속성	설명
enabled	부울		이벤트 수신을 시작하려면 true를 지정합니다. 이벤트 수신을 중지하려면 false를 지정합니다.
win	객체	<optional>	이벤트를 수신할 창입니다. 이 정보가 생략될 경우에는 기본 창이 사용됩니다.
displayId	number	<optional>	이벤트를 수신해야 하는 디스플레이의 ID입

이름	Type	속성	설명
			니다. 이 정보가 생략될 경우에는 기본 디스플레이 창이 사용됩니다.

반환 형식:

Type

void

`detachDisplay(displayId) → {void}`

특정 디스플레이를 분리합니다. 메인 디스플레이는 분리할 수 없습니다.

파라미터:

이름	Type	설명
<code>displayId</code>	number	분리할 디스플레이의 ID입니다.

반환 형식:

Type

void

`disconnect() → {void}`

Amazon DCV 서버와의 연결을 해제하고 연결을 종료합니다.

반환 형식:

Type

void

`disconnectCollaborator(connectionId) → {void}`

제공된 연결 ID로 연결된 공동 작업자의 연결 해제를 요청합니다(Amazon DCV 웹 클라이언트 SDK 버전 1.1.0부터).

파라미터:

이름	Type	설명
<code>connectionId</code>	부울	해제될 연결의 ID입니다.

반환 형식:

Type

`void`

`enableDisplayQualityUpdates(enable) → {void}`

업데이트를 수신하지 않는 스트리밍 영역의 디스플레이 품질 업데이트를 활성화하거나 비활성화합니다. 디스플레이 품질 업데이트를 비활성화하면 대역폭 사용량이 줄어들지만 디스플레이 품질도 떨어집니다.

파라미터:

이름	Type	설명
<code>enable</code>	부울	디스플레이 품질 업데이트를 활성화하려면 <code>true</code> 를 지정합니다. 디스플레이 품질 업데이트를 비활성화하려면 <code>false</code> 를 지정합니다.

반환 형식:

Type

`void`

`enableHighPixelDensity(enable) → {void}`

클라이언트에서 높은 픽셀 밀도를 활성화 또는 비활성화합니다.

파라미터:

이름	Type	설명
enable	부울	높은 픽셀 밀도를 활성화해야 하는지 여부.

반환 형식:

Type

void

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

시간대 리디렉션을 사용하거나 사용하지 않도록 설정합니다. 활성화되면 클라이언트는 서버 데스크톱 시간대가 클라이언트 시간대와 일치하도록 서버에 요청합니다.

파라미터:

이름	Type	설명
enable	부울	시간대 리디렉션을 활성화하려면 true를 지정합니다. 시간대 리디렉션을 비활성화하려면 false를 지정합니다.

반환 형식:

Promise를 반환합니다. 거부되면 promise는 오류 객체를 반환합니다.

Type

Promise.<number> | Promise.<{code: [TimezoneRedirectionErrorCode](#), message: string}>

`enterRelativeMouseMode()` → `{void}`

상대 마우스 모드를 활성화합니다.

반환 형식:

Type

`void`

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

클라이언트 컴퓨터에 연결된 미디어 디바이스 목록을 요청합니다.

반환 형식:

성공하면 Promise를 반환합니다. Promise는 MediaDeviceInfo 객체 배열로 확인됩니다. 자세한 내용은 <https://developer.mozilla.org/en-US/docs/Web/API/MediaDeviceInfo> 페이지를 참조하세요. 거부되면 promise는 오류 객체를 반환합니다.

Type

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

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

Amazon DCV 서버의 파일 스토리지를 관리할 객체를 가져옵니다.

반환 형식:

Promise를 반환합니다. 요청이 이행되면 파일 탐색기 객체로, 거부되면 오류 객체로 확인됩니다.

Type

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

`getServerInfo()` → `{serverInfo}`

Amazon DCV 서버에 대한 정보를 가져옵니다.

반환 형식:

서버 소프트웨어에 대한 정보입니다.

Type

[serverInfo](#)

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

원격 데스크톱의 스크린샷을 PNG 형식으로 검색합니다. [screenshotCallback](#) 관찰자에 스크린샷이 반환됩니다. 실패 시에는 그 대신 null이 반환됩니다.

반환 형식:

요청이 처리되면 확인되는 promise입니다. 거부되면 오류 객체가 수신됩니다.

Type

Promise | Promise.<{code: [ScreenshotErrorCode](#), message: string}>

getStats() → {[stats](#)}

Amazon DCV 서버에 대한 통계를 가져옵니다.

반환 형식:

스트리밍 통계에 대한 정보입니다.

Type

[stats](#)

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

허용된 보조 키에 대해 단일 키보드 keydown 또는 keyup 이벤트를 전송합니다.

파라미터:

이름	Type	설명
key	Control Alt AltGraph Meta OS Shift	전송할 키입니다.
location	KeyboardEvent.location	키의 위치입니다. 자세한 내용은 https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/location 페이지를 참조하세요.
isDown	부울	삽입할 키 이벤트가 keydown(true) 또는 keyup(false)인 경우입니다.

반환 형식:

요청된 조합이 유효한 경우 함수는 true를 반환하고 그렇지 않으면 false를 반환합니다.

Type

부울

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

Amazon DCV 서버에서 사용자 지정 데이터 채널을 만든 경우 연결에서 해당 채널을 엽니다.

파라미터:

이름	Type	설명
name	문자열	채널의 이름입니다.
authToken	문자열	채널 연결에 사용할 인증 토큰입니다.

이름	Type	설명
callbacks	객체	직접적으로 호출할 onMessage 및 onClose 콜백 함수입니다.
namespace	문자열	채널의 네임스페이스입니다. Amazon DCV 웹 클라이언트 SDK 1.2.0 및 Amazon DCV 서버 2022.1부터 사용할 수 있습니다.

반환 형식:

Promise를 반환합니다. 거부되면 오류 객체가 수신됩니다.

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}>}

특정 Amazon DCV 서버 기능의 상태를 쿼리합니다.

파라미터:

이름	Type	설명
featureName	feature	쿼리할 기능의 이름입니다.

반환 형식:

Promise를 반환합니다. 값이 확인되면 함수는 항상 enabled 속성을 포함하는 status 객체를 반환하며, 다른 속성도 포함할 수 있습니다. 거부되면 함수는 error 객체를 반환합니다.

Type

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

registerKeyboardShortcuts(shortcuts) → {void}

키보드 단축키를 등록합니다.

파라미터:

이름	Type	설명												
shortcuts	Array.<Object>	<p>등록할 키와 매핑의 배열입니다.</p> <table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>sequence</td> <td>Array.<Object></td> <td> <p>등록할 키보드 단축키입니다.</p> <table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>keyCode</td> <td>KeyboardEvent.keyCode</td> <td>키보드의 키 값입니다. </td></tr></tbody> </table> </td> </tr> </tbody> </table>	이름	Type	설명	sequence	Array.<Object>	<p>등록할 키보드 단축키입니다.</p> <table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>keyCode</td> <td>KeyboardEvent.keyCode</td> <td>키보드의 키 값입니다. </td></tr></tbody> </table>	이름	Type	설명	keyCode	KeyboardEvent.keyCode	키보드의 키 값입니다.
이름	Type	설명												
sequence	Array.<Object>	<p>등록할 키보드 단축키입니다.</p> <table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>keyCode</td> <td>KeyboardEvent.keyCode</td> <td>키보드의 키 값입니다. </td></tr></tbody> </table>	이름	Type	설명	keyCode	KeyboardEvent.keyCode	키보드의 키 값입니다.						
이름	Type	설명												
keyCode	KeyboardEvent.keyCode	키보드의 키 값입니다.												

이름	Type	설명		
		이름	Type	설명
				C T 설명 니 다. 자 세 한 내 용 은 see https:// d eveloper. mozilla.o rg/ en- US/ docs/ Web/ API/ Keybo ardEvent/ key 페 이 지 를 참 조 하

이름	Type	설명		
		이름	Type	설명
				설명 세 요.
				1 k 전 boardE v 송. loca ti 할 키 의 배 열 입 니 다. 키 보 드 의 키 위 치 입 니 다. 자 세 한 내 용 은 https:// d

이름	Type	설명		
		이름	Type	설명
				C T 설명 eveloper. mozilla.o rg/ en- US/ docs/ Web/ API/ Keybo ardEvent/ location 페 이 지 를 참 조 하 세 요.
		output	Array.<Object>	단축키를 사용하여 수행할 의도된 동작입니다.

이름	Type	설명		
		이름	Type	설명
				C T 설명 k K boardE v 용.key 자 가 누 른 키 의 값 입 니 다. 자 세 한 내 용 은 see https:// d eveloper. mozilla.o rg/ en- US/ docs/ Web/ API/ Keybo

이름	Type	설명		
		이름	Type	설명
				CT 설명 ardEvent/ key 페 이 지 를 참 조 하 세 요. 1 k 전 boardE v 송 loca ti 할 키 의 배 열 입 니 다. 키 보 드 의 키 위 치 입 니

이름	Type	설명		
		이름	Type	설명
				C T 설명 다. 자 세 한 내 용 은 https:// d eveloper. mozilla.o rg/ en- US/ docs/ Web/ API/ Keybo ardEvent/ location 페 이 지 를 참 조 하 세 요.

반환 형식:

Type

void

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

Amazon DCV 서버에서 업데이트된 디스플레이 구성을 요청합니다. Amazon DCV 웹 클라이언트 SDK 1.1.0 및 Amazon DCV 서버 2022.0부터 사용할 수 있습니다.

파라미터:

이름	Type	설명
highColorAccuracy	부울	높은 색상 정확도를 요청해야 하는지 여부입니다.

반환 형식:

Promise를 반환합니다. 거부되면 promise는 오류 객체를 반환합니다.

Type

Promise | Promise.<{code: [DisplayConfigErrorCode](#), message: string}>

requestDisplayLayout(layout) → {Promise|Promise.<{code: [ResolutionErrorCode](#), message: string}>}

연결을 위한 업데이트된 디스플레이 레이아웃을 요청합니다.

파라미터:

이름	Type	설명
layout	Array.< Monitor >	레이아웃에 요청된 디스플레이입니다.

반환 형식:

Promise를 반환합니다. 거부되면 오류 객체가 수신됩니다.

Type

Promise | Promise.<{code: [ResolutionErrorCode](#), message: string}>

requestResolution(width, height) → {Promise|Promise.<{code: [ResolutionErrorCode](#), message: string}>}

Amazon DCV 서버에서 업데이트된 디스플레이 해상도를 요청합니다.

파라미터:

이름	Type	설명
width	number	요청할 너비(픽셀 단위)입니다. 허용되는 최솟값은 0입니다.
height	number	요청할 높이(픽셀 단위)입니다. 허용되는 최솟값은 0입니다.

반환 형식:

Promise를 반환합니다. 거부되면 promise는 오류 객체를 반환합니다.

Type

Promise | Promise.<{code: [ResolutionErrorCode](#), message: string}>

sendKeyboardEvent(event) → {boolean}

키보드 단축키 이벤트를 전송합니다. 키보드 이벤트에 대한 자세한 내용은 <https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent> 페이지를 참조하세요. 유효한 키보드 이벤트에는 keydown, keypress, keyup 등이 포함됩니다. 키보드 이벤트에 대한 자세한 내용은 <https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent#events> 페이지를 참조하세요.

파라미터:

이름	Type	설명
event	KeyboardEvent	전송할 키보드 이벤트입니다.

반환 형식:

이벤트가 유효하지 않으면 함수는 `false`를 반환합니다. 이벤트가 유효하면 함수는 `true`를 반환합니다.

Type

부울

`sendKeyboardShortcut(shortcut) → {void}`

키보드 단축키를 전송합니다. 전체 `keydown` 또는 `keyup` 시퀀스를 전송하려면 이 함수를 사용하세요. 예를 들어 `Ctrl + Alt + Del`을 전송하면 모든 키에 대한 `keydown` 이벤트와 `keyup` 이벤트가 차례로 전송됩니다. 단일 키를 전송하려는 경우에도 이 함수를 사용하세요.

파라미터:

이름	Type	설명
shortcut	Array.<Object>	전송할 키의 배열입니다.

이름	Type	설명
key	KeyboardEvent.key	사용자가 누른 키의 값입니다. 자세 한 내용은 see https://

이름	Type	설명		
		이름	Type	설명
				d eveloper. mozilla.o rg/en- US/ docs/ Web/ API/ Keybo ardEvent/ key 페 이지를 참조하 세요.
		localitic	Keyboard vent.loca tion	전송할 키의 배 열입니 다. 키 보드의 키 위치 입니다. 자세한 내용은 https://d eveloper. mozilla.o rg/en- US/ docs/ Web/ API/

이름	Type	설명		
		이름	Type	설명
				KeyboardEvent/location 페이지를 참조하세요.

반환 형식:

Type

void

setDisplayQuality(min, maxopt) → {void}

연결에 사용할 이미지 품질을 설정합니다. 유효 범위는 0~100이며, 1이 최저 이미지 품질이고 100는 최고 이미지 품질입니다. 현재 값을 유지하려면 0을 지정합니다.

파라미터:

이름	Type	속성	설명
min	number		최소 이미지 품질입니다.
max	number	<optional>	최대 이미지 품질입니다.

반환 형식:

Type

void

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

버전 1.3.0부터는 사용되지 않습니다. 더 이상 디스플레이 배율을 설정할 필요가 없습니다. 마우스 좌표는 내부적으로 자동으로 관리됩니다.

클라이언트 측에서 디스플레이 배율이 조정되었음을 Amazon DCV에 알립니다. 이를 사용하여 클라이언트의 디스플레이 비율에 맞게 마우스 이벤트의 크기를 조정해야 함을 서버에 알릴 수 있습니다.

파라미터:

이름	Type	설명
<code>scaleRatio</code>	실수	사용할 조정 비율입니다. 반드시 양수여야 합니다.
<code>displayId</code>	number	조정할 디스플레이의 ID입니다.

반환 형식:

Promise를 반환합니다. 거부되면 promise는 오류 객체를 반환합니다.

Type

Promise | Promise.<{code: [ResolutionErrorCode](#), message: string}>

`setKeyboardQuirks(quirks) → {void}`

클라이언트 컴퓨터에 키보드 쿼크를 설정합니다.

파라미터:

이름	Type	설명
<code>quirks</code>	객체	사용하거나 사용하지 않을 키보드 쿼크입니다.

이름	Type	설명		
		이름	Type	설명
		macOpti ToAlt	부울	macOS 에서 Option 키를 Alt로 매핑하 려면 true로 지정합 니다. 아닌 경 우에는 false로 지정합 니다.
		macComn dToCont 1	부울	macOS 에서 Command 키를 Ctrl로 매핑하 려면 true로 지정합 니다. 아닌 경 우에는 false로 지정합 니다.

반환 형식:

Type

void

`setMaxDisplayResolution(maxWidth, maxHeight) → {void}`

연결에 사용할 최대 디스플레이 해상도를 설정합니다.

파라미터:

이름	Type	설명
maxWidth	number	최대 디스플레이 너비(픽셀 단위)입니다. 허용되는 최솟값은 0입니다.
maxHeight	number	최대 디스플레이 높이(픽셀 단위)입니다. 허용되는 최솟값은 0입니다.

반환 형식:

Type

void

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

마이크를 사용하거나 사용하지 않도록 설정합니다.

파라미터:

이름	Type	설명
enable	부울	마이크를 활성화하려면 true로 지정합니다. 마이크를

이름	Type	설명
		비활성화하려면 <code>false</code> 로 지정합니다.

반환 형식:

Promise를 반환합니다. 거부되면 promise는 오류 객체를 반환합니다.

Type

Promise | Promise.<{code: [AudioErrorCode](#), message: string}>

`setMinDisplayResolution(minWidth, minHeight) → {void}`

연결에 사용할 최소 디스플레이 해상도를 설정합니다. 일부 애플리케이션에는 최소 디스플레이 해상도가 필요할 수 있습니다. 필요한 최소 해상도가 클라이언트에서 지원하는 최대 해상도보다 큰 경우 크기 조정 전략이 사용됩니다. 이 함수는 주의해서 사용하세요. 크기 조정 전략으로 인해 마우스 및 터치 입력 시스템의 정확도가 떨어질 수 있습니다.

파라미터:

이름	Type	설명
<code>minWidth</code>	number	최소 디스플레이 너비(픽셀 단위)입니다. 허용되는 최솟값은 0입니다.
<code>minHeight</code>	number	최소 디스플레이 높이(픽셀 단위)입니다. 허용되는 최솟값은 0입니다.

반환 형식:

Type

void

setUploadBandwidth(value) → {number}

Amazon DCV 서버에 파일을 업로드하는 데 사용할 최대 대역폭을 설정합니다.

파라미터:

이름	Type	설명
value	number	최대 대역폭 제한(Kbps)입니다. 유효 범위는 1,024kbp~102,400kbps입니다.

반환 형식:

- 설정된 대역폭 제한입니다. 서버에서 파일 저장 기능이 비활성화된 경우 null입니다.

Type

number

setVolume(volume) → {void}

오디오에 사용할 볼륨 레벨을 설정합니다. 유효 범위는 0~100이며, 0은 가장 낮은 음량이고 100은 가장 높은 음량입니다.

파라미터:

이름	Type	설명
volume	number	사용할 볼륨 수준입니다.

반환 형식:

Type

void

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

[실험적 - 향후 변경될 수 있음] 마이크를 활성화 또는 비활성화합니다.

파라미터:

이름	Type	설명
enable	부울	마이크를 활성화하려면 true로 지정합니다. 마이크를 비활성화하려면 false로 지정합니다.
deviceId	문자열	마이크의 디바이스 ID입니다. deviceId가 제공되지 않으면 default deviceId가 사용됩니다.

반환 형식:

Promise를 반환합니다. 거부되면 promise는 오류 객체를 반환합니다.

Type

Promise | Promise.<{code: [AudioErrorCode](#), message: string}>

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

웹캠을 활성화 또는 비활성화합니다.

파라미터:

이름	Type	설명
enable	부울	웹캠을 활성화하려면 true를 지정합니다. 웹캠을 비활성화하려면 false를 지정합니다.
deviceId	문자열	웹캠의 디바이스 ID입니다.

반환 형식:

연결/분리된 웹캠의 deviceId가 확인되면 Promise를 반환합니다. 거부되면 promise는 오류 객체를 반환합니다.

Type

Promise.<string> | Promise.<{code: [WebcamErrorCode](#), message: string}>

syncClipboards() → {boolean}

로컬 클라이언트 클립보드를 원격 Amazon DCV 서버 클립보드와 동기화합니다. 브라우저에서 자동 복사를 지원해야 합니다.

반환 형식:

클립보드가 동기화되어 있는 경우 함수는 true를 반환합니다. 클립보드가 동기화되지 않았거나 브라우저가 자동 복사를 지원하지 않는 경우 함수는 false를 반환합니다.

Type

부울

인증 클래스

dcv 모듈의 [authenticate 메서드](#)를 직접적으로 호출하여 인증 토큰을 얻으려면 인증 클래스를 사용해야 합니다. 사용 방법을 보여주는 예제는 [시작하기](#) 섹션을 참조하세요.

노출

- [메서드](#)

메서드

List

- [retry\(\) → {void}](#)
- [sendCredentials\(credentials\) → {void}](#)

retry() → {void}

인증 프로세스를 다시 시도합니다.

반환 형식:

Type

void

sendCredentials(credentials) → {void}

클라이언트가 제공한 인증 자격 증명을 Amazon DCV 서버로 전송합니다.

파라미터:

이름	Type	설명
credentials	객체	제공된 자격 증명에 포함된 객체입니다. 자격 증명은 챌린지에 지정된 것과 이름 및 유형이 동일해야 합니다.

반환 형식:

Type

void

리소스 클래스

리소스 클래스는 방금 인쇄 또는 다운로드한 해당 파일을 가져오거나 삭제할 수 있습니다. 이러한 작업을 수행할 때 리소스 객체를 유일한 인수로 사용하여 해당 관찰자 함수 [filePrinted](#) 및 [fileDownload](#)가 각각 간접적으로 호출됩니다. 이러한 리소스를 수락하거나 거부하여 리소스가 참조하는 파일을 가져오거나 삭제할 수 있습니다.

노출

- [메서드](#)

메서드

List

- [accept\(urlParameters\) → {void}](#)
- [decline\(\) → {void}](#)

[accept\(urlParameters\) → {void}](#)

리소스를 로컬로 다운로드합니다.

파라미터:

이름	Type	설명
<code>urlParameters</code>	객체	리소스 가져오기 요청에 전달된 URL 검색 파라미터의 키/값 쌍을 포함하는 선택적 객체입니다.

반환 형식:

Type

`void`

`decline()` → {void}

리소스를 취소합니다.

반환 형식:

Type

void

Amazon DCV 웹 UI SDK

JavaScript React 구성 요소 라이브러리로, Amazon DCV 서버를 연결하고 도구 모음을 렌더링하여 원격 스트림과 상호 작용하는 DCVViewer라는 단일 React 구성 요소를 내보냅니다.

노출

- [구성 요소](#)

구성 요소

List

- [DCVViewer](#)

DCVViewer

React 구성 요소는 원격 스트림과 상호 작용하는 데 유용한 모든 기능을 갖춘 도구 모음을 렌더링합니다.

속성:

List

- [dcv](#)
- [uiConfig](#)

dcv

이름	Type	필수	설명												
dcv	객체	예	<p>Amazon DCV 서버에 연결하는 데 필요한 속성을 정의하는 객체로, Amazon DCV 웹 클라이언트 SDK 자산을 로드하고 DCV 리소스에 액세스하는 데 필요한 로그 수준 및 URL을 설정합니다.</p> <table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>필수</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>ses</td> <td>문자열</td> <td>예</td> <td>Amazon DCV 세션 ID입니다.</td> </tr> <tr> <td>aut</td> <td>문자열</td> <td>예</td> <td>서버에 연결할 때 사용할 인</td> </tr> </tbody> </table>	이름	Type	필수	설명	ses	문자열	예	Amazon DCV 세션 ID입니다.	aut	문자열	예	서버에 연결할 때 사용할 인
이름	Type	필수	설명												
ses	문자열	예	Amazon DCV 세션 ID입니다.												
aut	문자열	예	서버에 연결할 때 사용할 인												

이름	Type	필수	설명			
			이름	Type	필수	설명
			ser	문 자 열	예	<p>증 토 권 입 니 다.</p> <p>실 행 중 인 Amazon DCV 서 버 의 호 스 트 이 름 및 포 트 형 식 은 다 음 과 같 습</p>

이름	Type	필수	설명			
			이름	Type	필수	니다. https:// d cv_host_a dddress:po rt 예: https:// m y- dcv- ser ver:8443
			base	문 자 열	예	SDK 파 일 을 로 드 할 절 대 또 는 상 대 URL 입 니 다.

이름	Type	필수	설명			
			이름	Type	필수	설명
			release	문자열	아니요 (기본값: "")	DCV 리소스에 액세스할 수 있는 절대 또는 상대 URL입니다.
			onDirect	함수	아니요 (기본값: {} => {})	Amazon DCV 서버와의 연결이

이름	Type	필수	설명			
			이름	Type	필수	설명
						해제될 때 콜백 함수가 간접적으로 호출되고, 연결이 종료됩니다.
			log	Log	아니요 (기본)	뷰어에서 사

이름	Type	필수	설명			
			이름	Type	필수	설명
					값: Log INF	용할 1. 로 그 수 존 입 니 다.
			obs	객체	아니요 (기본 값: {})	구현을 정의하기 위해 httpExtra HeadersCallback 및 httpExtra SearchParamsCallback을 포함할 객

이름	Type	필수	설명			
이름	Type	필수	설명			
						체입니다. The Extra SearchPar (연결 설정 (영종영)) URL 에 사용자 지정 쿼리 파라미터를 주

이름	Type	필수	설명			
			이름	Type	필수	설명
						<p>입하기 위해 직접적으로 호출되는 콜백 함수입니다.</p> <p>Extra Headers 설정 (정수 값) (HTTP 요</p>

이름	Type	필수	설명			
			이름	Type	필수	설명
						필수 (예) 사용자 지정 헤더를 추가하기 위해 직접적으로 호출 콜백 함수인

이름	Type	필수	설명								
			<table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>필수</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td>필수입니다.</td> </tr> </tbody> </table>	이름	Type	필수	설명				필수입니다.
이름	Type	필수	설명								
			필수입니다.								

uiConfig

이름	Type	필수	설명								
uiConfig	객체	아니요 (기본값: {})	<p>도구 모음 표시 여부와 도구 모음에 전체 화면 및 다중 모니터 버튼을 표시할지 여부를 구성하는 속성을 정의하는 객체입니다.</p> <table border="1"> <thead> <tr> <th>이름</th> <th>Type</th> <th>필수</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>too</td> <td>객체</td> <td>아니요 (기본값: {})</td> <td>도구 모음의 구성 옵션을 정</td> </tr> </tbody> </table>	이름	Type	필수	설명	too	객체	아니요 (기본값: {})	도구 모음의 구성 옵션을 정
이름	Type	필수	설명								
too	객체	아니요 (기본값: {})	도구 모음의 구성 옵션을 정								

이름	Type	필수	설명			
			이름	Type	필수	설명
						의하는 객체입니다. 이 필수는 true 또는 false 값을 갖는 (true) 할지 숨길지 여부를 정의하는 옵션입니다.

이름	Type	필수	설명			
이름	Type	필수	설명			
						type 선택 선 입 니 다. screen button 요 (width value) 화 면 버 튜 을 표 시 할 지 슝 길 지 여 부 를 정 의 하

이름	Type	필수	설명			
이름	Type	필수	설명			
						<p>이 필수는 옵션입니다.</p> <p>이 Button 요 (웹 페이지) 모니터 버튼을 표시할지 숨길지 여부를</p>

이름	Type	필수	설명			
			이름	Type	필수	설명
<div data-bbox="1485 352 1550 892" style="border: 1px solid black; padding: 5px;"> 이 필드에는 정의를 하는 옵션이 없습니다. </div>						

Amazon DCV 웹 클라이언트 SDK에 대한 릴리스 정보 및 문서 기록

이 페이지에서는 Amazon DCV 웹 클라이언트 SDK에 대한 릴리스 정보와 문서 기록을 제공합니다.

주제

- [Amazon DCV 웹 클라이언트 SDK 릴리스 정보](#)
- [문서 기록](#)

Amazon DCV 웹 클라이언트 SDK 릴리스 정보

이 섹션에서는 출시일별 Amazon DCV 웹 클라이언트 SDK의 릴리스 정보를 제공합니다.

주제

- [1.13.2 — 2026년 2월 19일](#)
- [1.10.1 - 2025년 10월 22일](#)
- [1.9.100 - 2025년 7월 2일](#)
- [1.8.7 - 2024년 10월 31일](#)
- [1.8.4 - 2024년 10월 1일](#)
- [1.5.10 - 2023년 12월 19일](#)
- [1.5.6 — 2023년 11월 9일](#)
- [1.4.4 — 2023년 6월 29일](#)
- [1.4.0 — 2023년 3월 28일](#)
- [1.3.1 — 2022년 12월 9일](#)
- [1.3.0 — 2022년 11월 11일](#)
- [1.2.1 — 2022년 7월 21일](#)
- [1.2.0 — 2022년 6월 29일](#)
- [1.1.3 — 2022년 5월 23일](#)
- [1.1.2 — 2022년 5월 19일](#)
- [1.1.1 — 2022년 3월 23일](#)
- [1.1.0 — 2022년 2월 23일](#)

- [1.0.4 — 2021년 12월 20일](#)
- [1.0.3 — 2021년 9월 1일](#)
- [1.0.2 — 2021년 7월 30일](#)
- [1.0.1 — 2021년 5월 31일](#)
- [1.0.0 — 2021년 3월 24일](#)

1.13.2 — 2026년 2월 19일

빌드 번호	새로운 기능
<ul style="list-style-type: none"> • 의미 체계 버전: 1.13.2 • 빌드: 1074 	<ul style="list-style-type: none"> • DCVViewer 구성 요소에 대한 httpExtraHeadersCallback 및 httpExtraSearchParamsCallback 정의를 포함하도록 observers 객체 파라미터를 추가했습니다.

1.10.1 - 2025년 10월 22일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> • 의미 체계 버전: 1.10.1 • 빌드: 1011 	<p>다음과 같은 기능이 추가되었습니다.</p> <ul style="list-style-type: none"> • 모바일 브라우저 지원 추가(Android의 Chrome, iOS의 Chrome 및 Safari) • gestureEvent 연결 구성 콜백 함수 추가 • 터치를 트랙패드 모드로 활성화/비활성화하는 새 API setTrackpadMode • 엔드포인트를 프로브하는 새 API probe • 게임패드 지원 추가 	<ul style="list-style-type: none"> • 이제 마이크를 사용하는 원격 애플리케이션이 있는 경우에만 마이크를 가져오고 처리합니다(2025.0 서버 필요). • Alt 키 및 Firefox 버그 수정 • 휠 스크롤 경험 개선 • Firefox에서 웹캠 지원 활성화

빌드 번호	새로운 기능	변경 및 버그 수정
		<p>정보 수준의 로그에 오디오 통계 추가</p> <ul style="list-style-type: none"> 크기가 지원되지 않는 경우 커서 크기 조정 수정

1.9.100 - 2025년 7월 2일

빌드 번호	새로운 기능	
<ul style="list-style-type: none"> 의미 체계 버전: 1.9.100 빌드: 952 	<ul style="list-style-type: none"> Amazon DCV 서버에 대한 WebSocket 연결을 설정할 때 URL을 사용자 지정하는 <code>httpExtraSearchParamsCallback</code> 연결 구성 콜백 함수가 추가되었습니다(예: SigV4 추가). HTTP 요청에 사용자 지정 헤더를 추가하는 <code>httpExtraHeadersCallback</code> 연결 구성 콜백 함수가 추가되었습니다(예: SigV4). 	

1.8.7 - 2024년 10월 31일

빌드 번호	변경 및 버그 수정	
<ul style="list-style-type: none"> 의미 체계 버전: 1.8.7 빌드: 858 	<ul style="list-style-type: none"> Firefox 130 이상의 렌더링 수정 	

1.8.4 - 2024년 10월 1일

빌드 번호	새로운 기능	변경 및 버그 수정
<ul style="list-style-type: none"> 시맨틱 버전: 1.8.4 빌드: 840 	<p>다음과 같은 기능이 추가되었습니다.</p> <ul style="list-style-type: none"> 'Amazon DCV 웹 클라이언트 SDK'로 이름 변경 높은 dpi 디스플레이를 위한 새로운 API <code>enableHighPixelDensity</code> 추가 호환되는 브라우저에서 마이크를 선택할 수 있는 실험 API <code>setMicrophone</code> 추가 새로운 연결 오류 <code>GATEWAY_BUSY</code>, <code>UNSUPPORTED_CREDENTIAL</code> 및 <code>TRANSPORT_ERROR</code> 추가 새로운 종료 사유 <code>EXTERNAL_PROTOCOL_CONNECTION_EVICTED</code> 및 <code>DISCONNECTION_REQUESTED</code> 추가 	<ul style="list-style-type: none"> 웹캠 처리 개선 오디오 재생 처리 개선 WebCodecs 처리 개선 마이크와 웹캠의 플러그 및 언플러그 기능 개선 다중 모니터 사용 시 원격 창 드래그 개선 이제 파일 스토리지 업로드 및 다운로드 권한이 올바르게 전파됨 렌더링에 대한 사소한 수정

1.5.10 - 2023년 12월 19일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.5.10 빌드: 684 	<p>변경 및 버그 수정</p> <ul style="list-style-type: none"> 스트림 디코딩 오류 해결

1.5.6 — 2023년 11월 9일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.5.6 빌드: 659 	<p>변경 및 버그 수정</p> <ul style="list-style-type: none"> 스트림 디코딩 및 렌더링의 성능 개선 Internet Explorer 11 지원 종료

1.4.4 — 2023년 6월 29일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.4.4 빌드: 573 	<p>변경 및 버그 수정</p> <ul style="list-style-type: none"> 이제 뷰어 UI 구성 요소는 이를 지원하는 브라우저의 <code>navigator.keyboard.lock</code> API를 사용하여 전체 화면에서 특수 키를 처리합니다. Chrome 114 이상을 사용할 때 잘못된 색상이 표시될 수 있는 문제를 해결했습니다. WebCodecs 감지가 개선되었습니다. 창에 들어갈 때 마우스 버튼 상태와 관련된 문제를 해결했습니다. macOS에서 보조 키가 계속 눌러 있는 문제를 해결했습니다. 네트워크 성능 저하에 대한 오디오 안정성을 개선했습니다.

버전	릴리스 노트
	<ul style="list-style-type: none"> • 메모리 누수를 해결했습니다. • 시간과 수준을 포함하도록 로그를 개선했습니다.

1.4.0 — 2023년 3월 28일

버전	릴리스 노트
<ul style="list-style-type: none"> • 시맨틱 버전: 1.4.0 • 빌드: 476 	<p>새로운 기능</p> <ul style="list-style-type: none"> • 여러 파일을 업로드하는 새 <code>uploadFiles</code> 메서드를 <code>FileStorage</code> 객체에 추가했습니다. • 이제 뷰어 UI 구성 요소에서 드래그 앤 드롭으로 파일 업로드를 시작할 수 있습니다. • WebCodecs 브라우저 API는 이제 오디오와 웹캠에도 사용됩니다. <p>변경 및 버그 수정</p> <ul style="list-style-type: none"> • 동일한 페이지에서의 반복 연결과 관련된 메모리 누수가 해결되었습니다. • 이제 <code>setUploadBandwidth</code> 에서 최대 1Gbps의 값을 허용합니다. • UI 구성 요소의 렌더링이 최적화되었습니다. •

버전	릴리스 노트
	<p>Windows의 애니메이션 커서 지원이 해결되었습니다.</p> <ul style="list-style-type: none"> 동일한 작업에 텍스트와 이미지 데이터가 모두 있을 때 클립보드 지원 문제가 해결되었습니다. 웹캠 API의 안정성 향상: 요청이 이미 진행 중인 동안에는 설정을 변경할 수 없으며, 이제 <code>webcam.setEnabled</code> 에서 진행 중인 요청에 대한 디바이스 ID를 추적하고 Promise를 반환합니다. 뷰어 UI 구성 요소가 오류 발생 시 알림을 표시합니다.

1.3.1 — 2022년 12월 9일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.3.1 빌드: 413 	<p>변경 및 버그 수정</p> <ul style="list-style-type: none"> 시간대 리디렉션 UI가 서버와 동기화되지 않는 문제를 해결했습니다. 여러 번 재연결 후 메모리 누수가 발생하는 문제를 해결했습니다. 연결이 끊길 때 빈 페이지가 표시되는 문제를 해결했습니다. 오디오 디코더 종료 시 콘솔 경고가 발생하는 버그가 수정되었습니다.

1.3.0 — 2022년 11월 11일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.3.0 빌드: 407 	<p>새로운 기능</p> <ul style="list-style-type: none"> UI 뷰어 구성 요소에 Cloudscape(https://cloudscape.design)를 도입했습니다. 시간대 리디렉션에 대한 지원을 추가했습니다. <p>변경 및 버그 수정</p> <ul style="list-style-type: none"> DCV 뷰어에 포커스가 맞춰져 있을 때 비동기 클립보드에서 업데이트가 누락되는 문제를 해결했습니다. 클라이언트 측에서 디스플레이 크기를 조정할 때 <code>setDisplayScale</code> 함수가 더 이상 필요하지 않습니다. 이제 <code>DCVViewer</code> 구성 요소가 마운트 해제되면 <code>disconnect()</code> 를 자동으로 직접 호출합니다.

1.2.1 — 2022년 7월 21일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.2.1 	<p>변경 및 버그 수정</p> <ul style="list-style-type: none">

버전	릴리스 노트
<ul style="list-style-type: none"> 빌드: 358 	<p>Amazon DCV 서버 2019.1 및 이전 버전에 연결하지 못하는 문제를 해결했습니다.</p>

1.2.0 — 2022년 6월 29일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.2.0 빌드: 352 	<p>변경 및 버그 수정</p> <ul style="list-style-type: none"> 수신된 프레임이 지원되는 최대 해상도(4,096 x2,160)보다 클 때 발생하는 충돌 버그가 수정되었습니다. 이제 리소스 객체(fileDownload 및 filePrinted 관찰자에게 인수로 전달됨)에 객체에 대해 각각 리소스를 다운로드하고 삭제할 수 있는 accept 및 decline 메서드가 있습니다. 연결을 끊을 때 클립보드 자동 동기화에 대한 사소한 버그가 수정되었습니다.

1.1.3 — 2022년 5월 23일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.1.3 빌드: 329 	<p>변경 및 버그 수정</p> <ul style="list-style-type: none"> web-url-path 옵션을 지정할 때 제대로 연결되지 않는 문제를 해결했습니다.

1.1.2 — 2022년 5월 19일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.1.2 빌드: 322 	<p>변경 및 버그 수정</p> <ul style="list-style-type: none"> 연결 후 입력이 제대로 작동하지 않는 문제를 해결했습니다. 스케일 비율이 1보다 클 때의 마우스 좌표를 수정했습니다.

1.1.1 — 2022년 3월 23일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.1.1 빌드: 309 	<p>변경 및 버그 수정</p> <ul style="list-style-type: none"> 서버와의 통신 제한 시간이 초과되면 Transport Error를 보고합니다. 큰 해상도를 스트리밍할 때 반복되는 디코딩 오류를 해결했습니다.

1.1.0 — 2022년 2월 23일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.1.0 빌드: 295 	<p>새로운 기능</p> <ul style="list-style-type: none">

버전	릴리스 노트
	<p>DCVViewer React 구성 요소가 포함된 Amazon DCV 웹 UI SDK 라이브러리를 출시합니다.</p> <ul style="list-style-type: none"> • Amazon DCV 웹 클라이언트 SDK를 UMD 및 ES 모듈로 내보낼 수 있습니다. • 높은 색상 정확도 지원이 추가되었습니다. • 세션에 연결된 클라이언트를 나열하고 상호 작용하는 기능이 추가되었습니다. 연결 및 연결 해제에 대한 알림을 추가했습니다. <p>변경 및 버그 수정</p> <ul style="list-style-type: none"> • WebCodecs 디코딩 지원이 개선되었습니다. • 다양한 키보드 개선 사항이 있습니다. • 클립보드가 비활성화되었을 때 두 번째 화면이 열리지 않던 버그가 수정되었습니다.

1.0.4 — 2021년 12월 20일

버전	릴리스 노트
<ul style="list-style-type: none"> • 시맨틱 버전: 1.0.4 • 빌드: 249 	<p>새로운 기능</p> <ul style="list-style-type: none"> • 동일한 페이지에서 여러 개의 연결을 열 수 있도록 지원합니다. •

버전	릴리스 노트
	CDN에서 SDK 로드를 지원합니다.

1.0.3 — 2021년 9월 1일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.0.3 빌드: 202 	<p>새로운 기능</p> <ul style="list-style-type: none"> WebCodecs에 대한 실험적 지원입니다. 이 기능은 기본적으로 비활성화되어 있으며 새 속성 <code>enableWebCodecs</code> 를 사용하는 <code>ConnectionConfig</code> 객체를 통해 활성화해야 합니다. 클립보드: Chromium 기반 브라우저의 <code>image/png</code> 데이터 유형 지원이 추가되었습니다. 서버의 스크린샷을 PNG 이미지로 가져오는 관찰자 및 콜백을 추가했습니다(Amazon DCV 서버 2021.2 필요). <p>변경 및 버그 수정</p> <ul style="list-style-type: none"> 키보드 보조 키 처리가 개선되었습니다.

1.0.2 — 2021년 7월 30일

버전	릴리스 노트

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.0.2 빌드: 167 	<ul style="list-style-type: none"> 스타일러스 이벤트의 압력 감지 기능이 수정되었습니다. Chrome의 한국어 키보드 레이아웃 지원이 개선되었습니다.

1.0.1 — 2021년 5월 31일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.0.1 빌드: 141 	<ul style="list-style-type: none"> 연결 오류 및 종료 사유 전파 수정 파일 스토리지 청크 진행률 업데이트 수정 웹캠 처리 개선 오디오 입력 처리 개선

1.0.0 — 2021년 3월 24일

버전	릴리스 노트
<ul style="list-style-type: none"> 시맨틱 버전: 1.0.0 빌드: 81 	Amazon DCV 웹 클라이언트 SDK의 초기 릴리스입니다.

문서 기록

다음 표는 본 Amazon DCV 웹 클라이언트 SDK 릴리스 관련 설명서를 소개합니다.

변경	설명	Date
Amazon DCV 웹 클라이언트 SDK 버전 1.13.2	이제 Amazon DCV 웹 클라이언트 SDK 1.13.2를 사용할 수 있습니다. 자세한 내용은 SDK v.1.13.2 를 참조하세요.	2026년 2월 19일
Amazon DCV 웹 클라이언트 SDK 버전 1.10.1	이제 Amazon DCV 웹 클라이언트 SDK 1.10.1을 사용할 수 있습니다. 자세한 내용은 SDK v.1.10.1 을 참조하세요.	2025년 10월 22일
Amazon DCV 웹 클라이언트 SDK 버전 1.9.100	이제 Amazon DCV 웹 클라이언트 SDK 1.9.100을 사용할 수 있습니다. 자세한 내용은 SDK v.1.9.100 을 참조하세요.	2025년 7월 2일
Amazon DCV 웹 클라이언트 SDK 버전 1.8.7	이제 Amazon DCV 웹 클라이언트 SDK 1.8.7을 사용할 수 있습니다. 자세한 내용은 SDK v.1.8.7 을 참조하세요.	2024년 10월 31일
Amazon DCV 웹 클라이언트 SDK 버전 1.8.4	이제 Amazon DCV 웹 클라이언트 SDK 1.8.4를 사용할 수 있습니다. 자세한 내용은 SDK v.1.8.4 를 참조하세요.	2024년 10월 1일
Amazon DCV 웹 클라이언트 SDK 버전 1.5.6	이제 Amazon DCV 웹 클라이언트 SDK 1.5.6을 사용할 수 있습니다. 자세한 내용은 SDK v.1.5.6 을 참조하세요.	2023년 11월 9일
Amazon DCV 웹 클라이언트 SDK 버전 1.4.4	이제 Amazon DCV 웹 클라이언트 SDK 1.4.4를 사용할 수	2023년 6월 29일

변경	설명	Date
	있습니다. 자세한 내용은 SDK v.1.4.4 를 참조하세요.	
Amazon DCV 웹 클라이언트 SDK 버전 1.4.0	이제 Amazon DCV 웹 클라이언트 SDK 1.4.0을 사용할 수 있습니다. 자세한 내용은 SDK v.1.4.0 을 참조하세요.	2023년 3월 28일
Amazon DCV 웹 클라이언트 SDK 버전 1.3.1	이제 Amazon DCV 웹 클라이언트 SDK 1.3.1을 사용할 수 있습니다. 자세한 내용은 SDK v.1.3.1 을 참조하세요.	2022년 12월 9일
Amazon DCV 웹 클라이언트 SDK 버전 1.3.0	이제 Amazon DCV 웹 클라이언트 SDK 1.3.0을 사용할 수 있습니다. 자세한 내용은 SDK v.1.3.0 을 참조하세요.	2022년 11월 11일
Amazon DCV 웹 클라이언트 SDK 버전 1.2.0	이제 Amazon DCV 웹 클라이언트 SDK 1.2.0을 사용할 수 있습니다. 자세한 내용은 SDK v.1.2.0 을 참조하세요.	2022년 6월 29일
Amazon DCV 웹 클라이언트 SDK 버전 1.1.0	이제 Amazon DCV 웹 클라이언트 SDK 1.1.0을 사용할 수 있습니다. 자세한 내용은 SDK v.1.1.0 을 참조하세요.	2022년 2월 23일
Amazon DCV 웹 클라이언트 SDK 버전 1.0.1	일부 오타가 수정되었습니다. 사소한 버그가 수정되었습니다. SDK v.1.0.1 을 참조하세요.	2021년 5월 31일
초기 릴리스	이 내용의 첫 번째 발행입니다.	2021년 3월 24일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.