RemoteCall is a remote support solution for enterprises that diagnoses and resolves issues by accessing PCs or mobile devices remotely. As a pioneer in introducing the concept of remote support, RemoteCall has set the industry standard. Introducing RemoteCall improves both customer ROI and satisfaction by reducing consultation and resolution time.

RemoteCall SaaS

WebViewer SDK Integration User Guide

RemoteCall SaaS WebViewer SDK Integration User Guide
Ver. 1.0.

2025.8.19

Notice

RSUPPORT Co., Ltd. hereafter referred to as "RSUPPORT", does not take any responsibility for any marketability nor compatibility apart from the usage described in the document. Under no circumstances will RSUPPORT be liable to direct, indirect, incidental nor accidental damages due to errors in the manual.

The copyright and intellectual property rights in this document are held by RSUPPORT Co., Ltd. and are protected by the copyright laws of the Republic of Korea and international copyright treaties. Reproducing or copying any part or the entire contents of this Guide without the prior written consent of RSUPPORT Co., Ltd. is a violation of international copyright law.

Windows 7 / 8 / 8.1 / 10 / 11, Edge are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Android, Chrome are trademarks of Google LLC.

macOS is a trademark of Apple Inc., registered in the United States and other countries.

IOS is a trademark of Cisco.

The above trademarks are protected by the respective companies' copyrights.

Other third-party registered trademarks and terms protected by copyright mentioned in this manual are used solely for reference.

©2025 All Rights Reserved by RSUPPORT Co., Ltd.

Table of Contents

1. Overview

1.1. About the Service

RemoteCall is a remote support service developed and provided by RSUPPORT Co., Ltd.

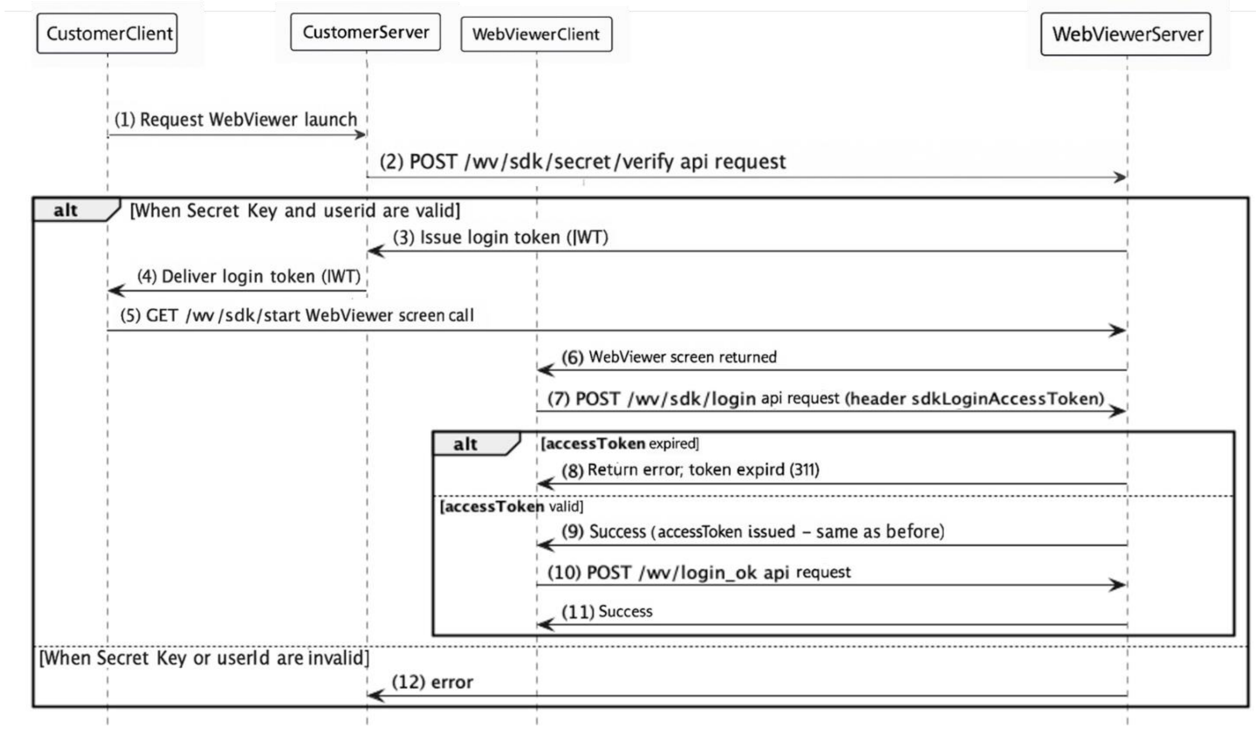
The service can be accessed and used directly through a browser without installing separate programs. RemoteCall uses secure connection code, allowing customers to use features such as explicit consent for remote control to start remote support on their PC or mobile device.

RemoteCall is a cloud-based service, enabling users to access the service anytime, anywhere from internet-enabled PCs or mobile devices.

This document explains how to use RemoteCall directly from customer systems without accessing www.remotecall.io.

1.2. Integration Service Configuration

To use RemoteCall, an account must be issued and token-based authentication is used when calling APIs with the issued account credentials. Customer systems can obtain tokens using API (Application Programming Interface) authentication information registered with RemoteCall, and use the provided APIs with the Secret Key required for WebViewer remote support execution using the issued tokens.


CustomerClient
CustomerServer
WebViewerClient
WebViewerServer
(1) Request WebViewer launch
(2) POST /wv/sdk/secret/verify api request
alt
[When Secret Key and userid are valid]
(3) Issue login token (JWT)
(4) Deliver login token (JWT)
(5) GET /wv/sdk/start WebViewer screen call
(6) WebViewer screen returned
(7) POST /wv/sdk/login api request (header sdkLoginAccessToken)
alt
[accessToken expired]
(8) Return error; token expird (311)
[accessToken valid]
(9) Success (accessToken issued – same as before)
(10) POST /wv/login_ok api request
(11) Success
[When Secret Key or userId are invalid]
(12) error

1.2.1. Key Features

1) RemoteCall features include remote support, file transfer, drawing, screen capture, and recording.

1.2.2. Support Scope

1) WebViewer
   a) Browser : Chrome, Edge (Chromium), Safari, Firefox, Opera, Samsung Browser, Whale. *Other browsers not supported.
   b) OS: Windows, Mac OS, Android, iOS, Chromium OS. *Other OS not supported.
2) Host
   a) Windows / MacOS / Android / IOS

1.3. Service Usage Environment

This section describes the environment for RemoteCall service and integration.

Environment requirements and policy content may change during regular updates. Users will be notified through helpdesk and email before updates.

1.3.1. User PC Environment

Category	Minimum Specifications	Recommended Specifications

OS	Windows 7, Mac OS 13	Windows 10, Mac OS 15
CPU	Dual core 2.4GHz	Quad core Processor
RAM	4 GB	8 GB
HDD	5 GB or higher	100 GB or higher
Display	HD resolution	FHD resolution
Web Camera	SD class (360p)	HD class (720p)
Network	10 Mbps or faster	Broadband network (100 Mbps or faster)

*** Minimum requirements may not provide optimal remote support quality and speed.**

*** Even with recommended requirements, quality may be affected when used with resource-intensive programs.**

*** OS minimum requirements align with each OS provider's End of Life Policy.**

1.3.2. User PC Web Browser Environment

Category	Chrome	Edge	Whale	Safari	Firefox
Minimum Ver.	100	100	3.14	15.4	99
Recommended Ver.	Latest	Latest	Latest	Latest	Latest

1.3.3. User PC Environment (Native APP)

Category	Windows	MacOS
Version	7 and later	13 and later

1.3.4. User Mobile Experience (Native APP)

Category	Android OS	iOS
Version	8.0 and later	16 and later

*** Mobile minimum requirements align with each platform’s End of Life Policy.**

*** Does not work on some Android devices that do not support the H.264 codec.**

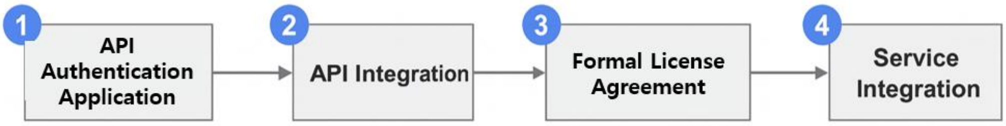
1.3.4. Firewall and Proxy Environment

Whitelist Registration	
Open Port	UDP 443, TCP 443

*** To register IP information as exceptions when using firewalls, please contact helpdesk or sales specialists.**

2. Integration Progress

To use RemoteCall API, customer information must be pre-registered in the system. By following the procedure described below, API authentication information for testing can be issued within 24 hours (business days). For full API authentication information, please contact the sales specialists specified on the last page of this document.



2.1. API Authentication Application

1) Please complete the application form according to the required format. Once the API authentication information application is approved, the information will be sent via email.
2) Survey Form:

2.2. API Integration

1) Server Information : https://www.remotecall.com (For inquiries, see "5. Contact Us" page)
2) Please review the API overview and list, then integrate with your customer system.
3) API authentication information issued for public use **automatically expires according to license information**. To request an extension, please contact your sales specialist.
4) Customers who have not proceeded with formal contracts **can extend only once**.

2.3. Formal License Agreement

1) Once all verification is complete, please proceed with the formal license agreement process. A formal license will be issued upon contract completion with your sales specialist.

2.4. Service Integration

1) The RemoteCall API is available under contractual terms.

3. API Overview

3.1. Connection Information

- https://remotecall.io

3.2. Authentication

URL

POST / wv/sdk/secret/verify HTTP/1.1
Host : remotecall.io

HTTP Headers

Accept: application/json
Authorization: Bearer **SecretKey**
Content-Type: application/json

Parameter

userId : value

* SecretKey: Issued for authentication

* userId : User's unique key

* Response status for successful authentication

Status : 200

* Response status for failed authentication (unavailable SecretKey)

Status : 401

* Response status for failed token issuance due to failed authentication

Status : 403

3.3. Error Handling

REST API returns standard HTTP status codes to indicate request success or failure. Response format is sent in JSON format as follows:

```
{
  "retcode": "997",
  "message": "secretKey not matched."
}
```

Returns return codes and messages containing problem details and solutions.

Error Response

Code	Type	Description
Retcode	String	Response code (3 digits)
Message	String	Error messages

* For more information on error codes, see the Error Code Specification in the appendix.

Error Page

* Service error codes, excluding HTTP status errors, are displayed as shown above.

4. API Overview

4.1. WebViewer SDK User Authentication

This method calls the remote support page from the customer system to invoke the consultation standby screen (page invocation method). When the request is completed successfully, the screen shown below is displayed, allowing users to call the consultation standby screen.

API

URI	/wv/sdk/secret/verify		
Features	SDK user authentication	Method	POST
Request Type	application/json	Response Type	application/json
Description	Verify if the customer is an SDK license user.		

Request Header

No	Name	Parameters	Type	Required	Description
1	SecretKey	Authorization	String	O	SecretKey

Request Data

No	Name	Parameters	Type	Required	Description
1	User’s unique key	userId	String	O	User’s unique key

Request Sample

Request Headers

Accept: application/json

Authorization: Bearer 0869f8f0ce2cbef5a2a261953ce089acff1bb78855d77909c86d0e953baf68ec404ccac4bf811373bd2ab8f6cefddb9abbf531e9b7c73086096911b24a040934

Content-Type: application/json

Request Body

POST

```
{ "userId": "000000007ff6c47e01801c0ee3490034" }
```

Response Data

No	Name	Type	Required	Description
1	Retcode	String	O	100: Success, otherwise failed. For more information on error codes, see the Error Code Specification in the appendix.
2	Message	String	O	SUCCESS: Success / Others: Failed
3	sdkLoginAccessToken	String	O	Tokens required for SDK validation
4	requestId	String	O	ID for SDK API history

Response Sample

```
{
  "sdkLoginAccessToken":
"eyJhbGciOiJIUzI1NiJ9.eyJleHAiOjE3NTE4NjMyODIsInVzZXJJZCI6IjAwMDAwMDAwN2ZmNmM0
N2UwMTgwMWMwZWUzNDkwMDM0IiwiaWF0IjoxNzUxODYyOTgyfQ.BHn43Tio7i7qBRqp2iuIL
QtYipERjFT4uyym4w_hRS4",
  "requestId": "8a82f4c197e25fa90197e32bb5cd0007",
  "retcode": "100",
  "message": "SUCCESS"
}
```

4.2. WebViewer SDK Integration

API

URI	/wv/sdk/start		
Features	Run SDK	Method	GET
Request Type	N/A	Response Type	text/html
Description	This method uses the token obtained through '4.1. WebViewer SDK User Authentication' to call the remote support page. The parameters listed below must be passed as query strings.		

Request Data

No	Name	Parameters	Type	Required	Description
1	SDK Login Access Token	sdkLoginAccessToken	String	O	The token returned in the response when the '4.1. WebViewer SDK User Authentication' API call succeeds.
2	Request ID	requestId	String	O	The unique request ID returned in the response when the '4.1. WebViewer SDK User Authentication' API call succeeds.
3	Screen Sharing Type	sss	String	X	Define the default type of screen sharing during remote support. - **Host (default):** The standard remote support mode where the customer's screen is shared. - **Viewer**: The mode where the support agent's screen is shared with the customer. (* Available only if the "Agent screen sharing" feature is enabled in Group settings)

					Note: If the 'Agent screen sharing' feature is disabled, the setting will automatically switch to Host mode even if Viewer is selected.

Request Sample

```
https://remotecall.io/wv/sdk/start?sdkLoginAccessToken=eyJhbGciOiJIUzI1NiJ9.eyJleHAiOjE3NT
E4NjMyODIsInVzZXJJZCI6IjAwMDAwMDAwN2ZmNmM0N2UwMTgwMWMwZWUzNDkwMDM0Iiw
iaWF0IjoxNzUxODYyOTgyfQ.BHn43Tio7i7qBRqp2iuILQtYipERjFT4uyym4w_hRS4&requestId=B8a
82f4c197e25fa90197e32bb5cd0007&sss= "ssshost
```

Call Method

When a session ends, the WebViewer automatically closes the window using the window.close() script. Refer to the descriptions below for each call method and choose the one that best suits your environment.

4.2.1. Invocation via HTML Tag

1-A. Static URL Configuration

```
<a href="https://remotecall.io/wv/sdk/start?sdkLoginAccessToken=...&requestId=..." target="_blank">Remote
Support</a>
```

1-B. Via <a> tag in JavaScript

```
<a id="remoteLink" href="#" target="_blank">Remote Support< /a>
<script>
    const requestUrl = `https://remotecall.io/wv/sdk/start?sdkLoginAccessToken=...&requestId=...`;
    document.getElementById('remoteLink').href = requestUrl;
</script>
```

Note: This method is simple but may not automatically close the window after the session ends, depending on the browser or its settings.

4.2.2. Invocation via JavaScript Event Handler (Recommended)

This method opens the window using the window.open( ) function when an event (such as a button click) occurs. It provides the most reliable operation of the automatic window-closing feature, and is therefore **strongly recommended**.

```
<button onclick="openWebViewer()">Remote Support</button>
<script>
   function openWebViewer() {
      const requestUrl = `https://remotecall.io/wv/sdk/start?sdkLoginAccessToken=...&requestId=...`;
      // with window.open() Opens a new window
      window.open(requestUrl,  '_blank', 'noopener,noreferrer');
   }
</script>
```

4.2.3. Invocation via Android Event Handler

When a button click event occurs, the data received from the server is added as URL query parameters. The value is then stored in an intent variable, and the window is opened by calling the startActivity(intent) function.

```
   val requestUrl = "https://remotecall.io/wv/sdk/start?sdkLoginAccessToken=...&requestId=..."

... Omitted...

btnOpenWebViewer.setOnClickListener {

         openWebViewer()

}

private fun openWebViewer() { {

         val intent  =  Intent(Intent.ACTION_VIEW,  Uri.parse(requestUrl))

         intent.setPackage("com.android.chrome") // Open a new window on Chrome

         startActivity(intent)

}
```

4.2.4. Invocation via iOS Browser

When a button click event occurs, the data received from the server is added as URL query parameters. The value is then passed to the UIApplication.shared.open() function to open the installed browser.

```
val requestUrl = "https://remotecall.io/wv/sdk/start?sdkLoginAccessToken=...&requestId=..."

... Omitted...

@IBAction func OpenWebBrowser(_ sender: Any) {

    DispatchQueue.main.async{

        self.openUrl(requestUrl)


    }

}


func openUrl(_ url:String){

    guard let nUrl = URL(string: url) else {

        return

    }

    UIApplication.shared.open(nUrl, options: [:])

}
```

5. Contact Us

For inquiries related to the use of the RemoteCall Service API, please refer to the contact information below.

Sales: 070-7011-0683
제휴/제품 문의 : sales.kr@rsupport.com
연동/개발 문의 : cs.asp@rsupport.com
제품 고객 센터 : https://help.remotecall.com

- End of Document -

Appendix: Error Code Specification

Code	Description
893	The SDK secretKey does not exist in the license
901	Required parameter missing
992	User or license not found
997	The provided secretKey does not match the one registered in the license
999	Unknown server error

Appendix: Glossary

Terms	Description
userId	Unique key assigned to the customer’s support agent
sdkLoginAccessToken	Tokens used for WebViewer SDK verification
requestId	Unique key for SDK API activity tracking
sss	Webviewer screen sharing type