Co-Browsing 22.1.0000

User Guide

TOC

Overview

Co-Browsing (also known as collaborative browsing) is a technology that allows peer-to-peer communication between two browsers and delivers instant contextual communication.

Co-browsing could be used in a number of digital journeys, both for banking and insurance use cases:

- Customer onboarding;
- Loan application;
- Mortgages;
- Compliance related processes;
- Claims handling and many more.

With Co-Browsing, customers and operators can interact in real-time, streamlining an in-person experience. Operators can see the customers' screens in real-time and guide them remotely though forms, transactions, and processes by either highlighting relevant areas on the customers' screens or by taking control of the customers' web sessions and performing actions on behalf of the customers (if the customer allows it).

Co-Browsing uses WebRTC (Web Real Time), an open framework for the web that enables Real-Time Communications (RTC) capabilities in the browser. This provides a 100% web-based experience that requires no downloads, installations, or plugins.

FintechOS enables you to add Co-Browsing to your digital journeys allowing you to connect operators and customers in real-time. This increases operators' efficiency by reducing call-handling time and provides faster customer service, enhanced customer satisfaction, and improved business performance.

Features

- **Enhanced User Experience**. High quality and scalable video communications for your FintechOS web and mobile applications;
- **Quality of Service**. Dynamically prioritize audio traffic over video on slow network connections;
- **Adaptive Layout**. Adapt stream layout and display based on audio detection;
- **Browser sharing in real time**. Operators and customers share their active web session while in a video call;
- **Visual drawing tool**. Operators can highlight a specific area on the customer's screen to instantly point the customer to specific actions;
- **Chat**. Customer and operator can interact via text messages;
- **Control switching**. The leader or the participant(s) can take control or request control of the session in seconds, and aid the other call participant by navigating on the screen in real-time.

Security

Secure co-browsing sessions are important, especially when taking into account that personal data is exchanged in banking or insurance digital journeys. The FintechOS Co-browsing capability ensures data protection and security:

- **Data in transit is encrypted**. Co-browsing sessions use HTTPS connections, SHA-256 SSL certificate and AES 128-bit encryption to protect sensitive data in transit.

- **No data is stored**. FintechOS does not store co-browsing session data. The data lives in the memory during the co-browsing session.
- **Behavior control.** The customer controls the operator's permissions.
- **Isolated control.** The person who controls the session can hide sensitive elements from other session participants.
- **Data Privacy.** Sensitive customer data is protected using field masking. The fields containing customer sensitive data are obfuscated, not shown in plain text to the operators and other session participants (if any were invited).
- **White/Black Listing**. Allow or deny access and privileges to specific members.
- **Action audit**. Track all actions performed during the co-browsing session.

Co-Browsing Streaming Flow

Co-Browsing is the continuous transmission of video content between a customer and a server. The customer can start a co-browsing session with a bank consultant or call center operator, receive real-time assistance during onboarding journeys by browsing and filling out information together with the consultant or operator.

Find below the co-browsing streaming flow:

1. The customer initiates a co-browsing session.
2. The FintechOS gateway sends instructions to the Co-browsing Service to create a co-browsing session.
3. The Co-browsing Service creates a session and sends the session ID back to the FintechOS gateway.
4. The FintechOS gateway creates a unique token and sends the session information to the FintechOS Portal and the customer is added to the queue.
5. When an operator picks up the customer's session from the queue, the FintechOS gateway sends the session ID and token to the customer.
6. The customer uses the token to connect to the session.
7. The co-browsing session starts and the participants (customer and operator) can start interacting in real-time.

Example

Let's take an example of an loan origination journey. The customer, John launched the journey on his laptop at home. After going through several steps, including identity verification, John now needs to choose the best loan offer for him. However, John needs help with picking an offer. John follows

the next steps:

1. From the customer journey, John initiates a co-browsing request.
2. The bank operator picks up the co-browsing call from the queue.
3. The bank operator then clicks **Take call** to join the co-browsing session.

During the session, John can add more users to the session, like his wife if he decides to add a coborrower to the journey. John can also switch control to the bank operator, thus allowing them to assist by explaining each options available in the flow's step.

When given control, the bank operator gets the same permissions as the customer.

With real-time assistance from a bank operator, the customer is able to get all the information they need and complete the journey in good time.

Installing Co-Browsing

1 Install the SysPacks

Make sure you have the **SysPack v22.1.0000** installed on your system. To do so:

1. Using a web browser, log in to your FintechOS Community account.
2. Select the **Release Hub**.
3. Open the **FintechOS 22.R2** folder.
4. Download the **SySDigitalSolutionPackages v22.1.0000.zip** archive.
5. Unzip the archive and follow the instructions in the **How to install FintechOS SysPacks v22.1.0000.pdf** file to install the SysPacks.

NOTE Make sure the following packages are deployed:

- 10_01 FTOS Cognitive Processor OperatorDM - v20.2.11x1
- 10_02 FTOS Cognitive Processor OperatorScripts - 20.2.11x1
- 100 DFP Common Scripts - 21.2.11x0

2 Set up the Co-Browsing Service Subscription Key

IMPORTANT! Make sure that the Vault configurations are done and that the Webhook is created.

In Vault

Key Path	Key	Value
kv/<environment>/<FintechOS Portal instance>/app-settings	FTOSServiceCobrowsingEndpoint	DCS web app endpoint URL provided by FintechOS.
kv/<environment>/<FintechOS Portal instance>/app-settings	FTOSServicesCobrowsingAppId	ID for the Surfly co-browsing service subscription.
kv/<environment>/<FintechOS Portal instance>/app-settings	FTOSServicesCobrowsingSubscriptionKey	Subscription key for the Surfly co-browsing service.

The configuration required by Co-browsing consists of the following:

- Serilog Configurations
- EbsSqlServer: Connection string pointing to the database where configurations are stored.

In order to correctly identify the sub-account configuration, Co-browsing requires an extra configuration to be made. This helps retrieve the Surfly Api Key based on the subscription key. For this, an entry in the FintechOS management instance, the ApiKeyRelation entity has to be edited accordingly, similar to the image below:

Field	Description
ApiTypeId	The ID type. It needs to be Cobrowsing.
Name	The name given session name. Must be unique.
SubscriptionKey	Represent the display name of the given subscription key. Must be different from Name.
ApiKey	The API key to be sent to the co-browsing session.

3 Set up the Processor Settings

Configure the following **ProcessorSetting**:

```
{
    "SourceEntityName": "Entity_Name",
    "QueueParameters": [
        {
            "ParamName": "Name",
            "ParamValue": "VideoQueue",
        }
    ]
}
```

- `SourceEntityName` is the entity from which the co-browsing session is initiated.
- `QueueParameters`, the `ParamValue` is the queue value item of the co-browsing session.

Adding Co-Browsing to a Digital Journey

The Co-browsing automation block can be added at any step of the customer journey. Follow the steps below to integrate it in a flow:

1. In **Innovation Studio**, create a custom journey to define a button to call the Co-browsing automation block. For information on how to create a custom journey, section Custom Flows.
2. Go to the form driven journey or form step on which you want to add the button to call the Co-browsing automation block.
3. Click the **Advanced** tab.
4. Click the **After Events** tab (for Steps, it is displayed by default, being the only tab available).

5. In the JavaScript field, provide the following code to call the co-browsing processor:

```
var csl = ebs.importClientScript("FTOS.DFP.Cobrowsing")
if(typeof(__surfly) === 'undefined'){

    $("#cobrowsingButton").on("click", function() {
        csl.createCobrowsingSessionEvent(
            formData,
            window.location,
            sessionStorage.B2CSessionIdParam,
            {
                settings: {
                    // these settings change depending on
what entity your form edits
                    recordId:
formData.model.myAccountApplicationid,
                    ProcessorSettings: "CobrowsingQueue",
                    // this is just an example, your
processor/flow settings
                    // can be named however you wish
                }
            }
        )
    })
}
else {
//$("button#cobrowsingButton").hide();
//if the button isn't hidden, use the other selector
$("button#cobrowsingButton")
    $("#cobrowsingButton").hide();
}
```

6. Click the **Save and Close** button () at the top right corner to save your digital journey.

IMPORTANT!

- If the journey contains multiple processors, if the Co-browsing session is closed in a step including a processor, then the flow is automatically redirected to the first step containing a processor.
- After the Co-Browsing session ends, the user needs to be redirected to the step in which the session was closed. For this, the journey needs to have a Business Workflow behind it, as well as a custom code for redirecting.

Troubleshooting

- if `ebs.importClientScript` does not load in a B2C environment and you get the following error message: "clientScriptLibray not found: FTOS.DFP.Cobrowsing", then use the following workaround:
    - On the form driven flow entity call "`formScope.csl = ebs.importClientScript("FTOS.DFP.Cobrowsing`".
    - On the button, instead of "`csl.createCobrowsingSessionEvent`" call "`formScope.csl.createCobrowsingSessionEvent`".
- for *recordid*, instead of `formData.model.myAccountApplicationid`, you must add the Id of the entity used for launching the co-browsing session. The following can also be used: `recordId: ebs.getCurrentEntityId()`.
- `ProcessorSettings` value is the name of the file configured after installation here.
- In the `FTOS.DFP.Cobrowsing` client script library, at the `buildURLFromFormData` and `buildURLFromCustomAction` functions, for obtaining the URL you must take into account if the `location.host` has proxy or not. The initial code is:

```
location.protocol +
                "//" +
                location.host +
                "/Main" +
                continueToUrl;
```

If the environment's URL contains "`proxy`" or "`b2cproxy`", then the following must be added:

```
continueToUrl = location.protocol +
                "//" +
                location.host +
                "/proxy/Main" +
                continueToUrl;
// or
continueToUrl = location.protocol +
                "//" +
                location.host +
```

```
                "/b2cproxy/Main" +
                continueToUrl;
```

- On the environment, the IPs obtained from this URL must be whitelisted.