Screen Sharing

- Screen sharing example
 - Extension for Google Chrome with publication in Chrome Store
 - Create a Google account
 - Register as Chrome Web Store Developer
 - Customization for your domain
 - Package the extension
 - Publish the extension
 Extension Inline Install
 - Extension Inline Installation
 Configuring the client
 - Configuring the clie
 Media source parameters
 - Parameter list
 - System sound capture in Chrome browser
 - Capture source (screen or window) management in Firefox browser
 - Screen sharing without extension
 - Firefox browser
 - Chromium based browsers
 - Safari browser in MacOS
 - Known limits
- Code of the example
- Analyzing the code

Screen sharing example

Screen sharing is available for Chrome and Firefox. To share screen in Chrome before 73 the extension need to build and install, in the latest Chrome versions (73 and newer), Firefox and Safari screen can be shared without extension. Today, it is not recommended to use extension.

		Screen Sharing)	
FPS 30	Width 640	Height 480		Two-way
	Use Mic? ₽	Mic Default - Microphone (Reall]	Loat
	Two-way Streaming	Name And And And And And And And And And And	→	PUBLISHING
	DOURNAL AND HOW DOURNAL AND HOU DOURNAL AND HO			DOWNLOAD NOW
wss://test2 flas	My Sc hphoner.com:8443/3480c6			Preview
133.116342.11d3	PUBLIS			

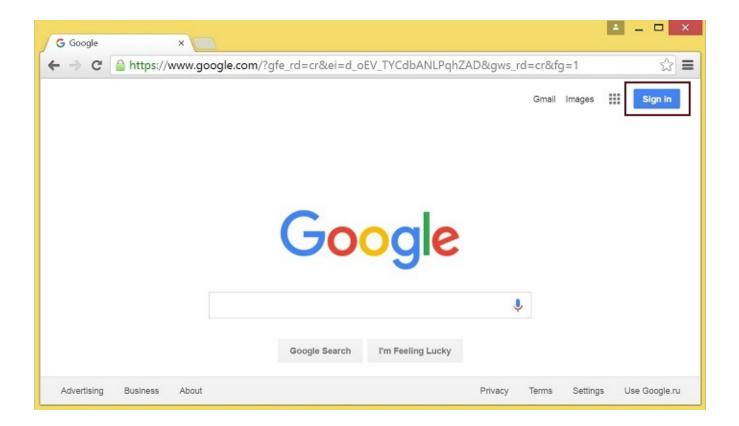
Extension for Google Chrome with publication in Chrome Store

Source code for the extension is available by the following link:

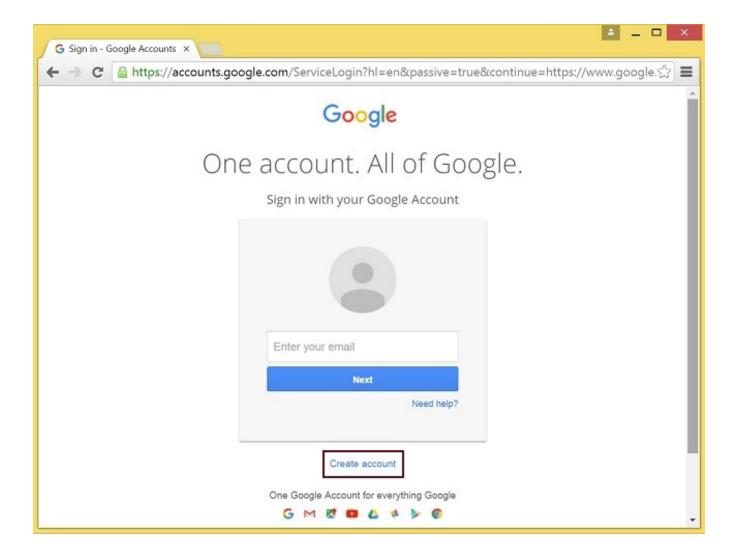
Chrome Screen Sharing Extension

Create a Google account

1. Go to google.com and click 'Sign in' button



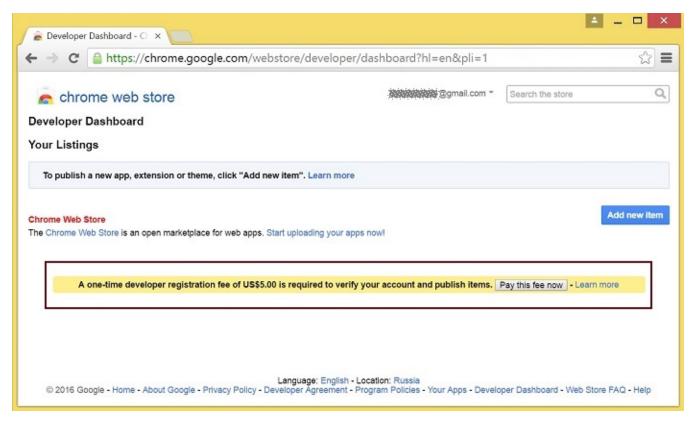
2. On the 'Sign in' page, click 'Create account' link



3. 'Create your Google Account' page will be opened Fill the required fields and click 'Next step' button to create the account.

Register as Chrome Web Store Developer

1. Sign in to Chrome Developer Dashboard with the created Google account 2. Pay a one-time \$5 developer signup fee



Customization for your domain

Follow the procedures described below to use the extensions with your domain. Edit manifest file manifest.json of the Chrome extension.

Change:

- name
- author
- description
- homepage_url
- under "externally_connectable":"matches" change flashphoner.com to your domain

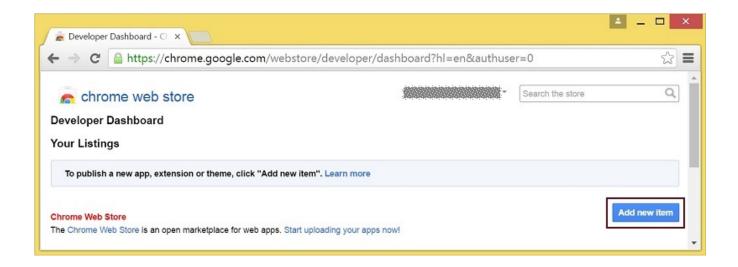
Save your icons for the extension to chrome-extension directory and edit the file names in "icons" and "web_accessible_resources". (For more information, see Manifest - Icons and Supplying Images.)

Package the extension

Package files from chrome-extension folder into ZIP archive.

Publish the extension

- Sign in to Chrome Developer Dashboard
 In the dashboard, click 'Add new item' button



3. Accept the developer agreement

C C https://chrome.google.com/webst	tore/developer/terms?hl=en&continue=ht	tps://chrome.google.	.com/wೕನ್ನ
chrome web store		Search the store	(
oogle Chrome Web Store Developer A	greement	*	
finitions			
oogle: Google Inc., a Delaware corporation with principal place ew, CA 94043, United States.	of business at 1600 Amphitheatre Parkway, Mountain		
veloper or You: Any person or company who is registered to de b Store in accordance with the terms of this Agreement.	evelop Products for publication and distribution on the		
ab Store: The Google Chrome Web Store site operated by Goo tribution to users of Google Chrome.	gle, where Developers can publish Products for direct		
yment Processor(s): Any party authorized by Google to provide th optional Payment Accounts to charge Google Chrome users	e payment processing services that enable Developers for Products distributed via the Web Store.	•	
ter-friendly version			

4. On the 'Upload' page, choose the chrome-extension.zip file and click 'Upload' button

				4	_ 🗆 🗙
🖉 🖉 Upload - Developer Dashb	×				
← → C 🔒 https://cl	hrome.google.com/w	ebstore/developer/	update?hl=en&authuser=1	&publisherId=g1111	631047☆ 🔳
n chrome web st				Search the store	٩
Developer Dashboard > Add ner		Remove			
Upload a ZIP file of your iter Include a well-designed proc	duct icon in your manifest (m out creating and packaging a	ore info). apps.			

5. When the extension is uploaded, the page for editing the extension draft will be opened Edit the extension as required and click 'Save draft and return to dashboard' button at the bottom of the page.

🖉 🚡 Test Screen Sh	aring - Edit X	□	×
		ore/developer/edit/hjjhafbcdlcbmjkbjogdlhdofkpnggmj?hl=en ゲ	3 =
	e web store ard > Test Screen Sharing	Search the store	2
Edit Item	In-app Products		
Upload Detailed description	Test Screen Sharing Short name: Not specified in m Version 1.4 by Upload Updated Package This Chrome extension is developed for <yo< th=""><th></th><th>y</th></yo<>		y
lcon	Upload new icon		

Inline install	☑ This item uses inline install.	Select this checkbox to allow inline install for your item (note that this may delay publishing). More info
Analytics	Google Analytics ID UA (example: 1234567-1)	
lequirements	No requirements	
Visibility options	Public Everyone can see It	
	Only people with the link can see it.	
	 Private Only trusted testers from your developer dashboard can see it. You can also include members of a Google Group that you own or manage. 	

6. The extension will appear in the developer dashboard Click 'Publish' link to publish the extension.

	,	iere, actoroper	, adonoodra,	915521150450	755806720?hl=en	2
n chrome web store				-	Search the store	Q
eveloper Dashboard						
Check out new Chrome Web Store Multip		which reduces the	download size of	f your application by o	distributing only relevant files t	o the
target system based on its NaCI architect	ure.					
our Listings (1 - 1 of 1)	Created	Last published	Weekly users	Status		
Test Screen Sharing					r	Dublish Edit
Version 1.4	4/6/16		0	Draft	L	Publish Edit More info
Target users in other languages. More in	nfo 🔻					
Target users in other languages. More in	nfo 🔻					

A published extension will have status 'Published' as on the image below.

	J	tore/develope	r/dashboard	?hl=en-US&auth	iusei – i	F 🗘
n chrome web store				*	Search the stor	e Q
eveloper Dashboard						
our Listings (1 - 1 of 1)	Created	Last published	Weekly users	Status		
	ng					Stats Unpublish Edit

For more information, see Chrome Web Store Publishing Tutorial.

Extension Inline Installation

Extension installation can be initiated by clicking a link on the Screen Sharing client page. Note that the extension to be installed should be customized, published and approved before inline installation can be used.

Follow the procedures described below to use the client with your extensions.

1. When publishing, select 'Inline Install' option

Inline install	☑ This item uses inline install.	Select this checkbox to allow inline install for your item (note that this may delay publishing). More info
Analytics	Google Analytics ID UA (example: 1234567-1)	
equirements	No requirements	
Visibility options	Public Everyone can see it	
	Olly people with the link can see it.	
	 Private Only trusted testers from your developer dashboard can see it. You can also include members of a Google Group that you own or manage. 	

2. Verify and add website with your domain to the extension - Click 'Add a New Site'

→ C 🔒 h	t ps://chrome.google.com /webstore/developer/edit/hjjhafbcdlcbmjk	bjogdlhdofkpnggmj?hl=en&authuser=0 🖓
Websites	Verify that this is an official item for a website you own: None Add a new site Refresh list Link to website for your item (optional)	If your item is associated with a website that you own, select that website from the list. You can register as the owner for a site using Google Webmaster Tools. More info Providing the URLs for description and support pages can improve your users' experience and
	② Link to support & FAQ for your item (optional)	help make this item's ratings and comments more meaningful. More info

- Google Search Console page will open in a new tab Enter URL with your domain.

🖉 Developer Dashboard - 🔾	× 🖶 Search Console - Home 🛛 ×	÷ _	
← → C 🔒 https://v	ww.google.com/webmasters/tools/home?hl=en		☆ 〓
Google		0	S
Search Console	Help ~	\$	•
Home All Messages Other Resources	Welcome to Search Console Get the data, tools, and diagnostics needed to create and maintain Google-friendly websites and apps. To get started, just add your site or app now. Image:	l mobile	e

- A page with the instruction for the site verification will open Follow the procedure steps and click 'Verify' button.

🖉 🚡 Developer Dashboard - 🗆 🗙	Search Console ×
← → C 🔒 https://www.g	$oogle.com/webmasters/verification/verification?hl=en&authuser=0&the \Im$
Google	
Search Console	Help ~
Verify your ownership of https://m.	flashphoner.com/. Learn more.
Recommended method	Alternate methods
Recommended: HTML file of Upload an HTML file to your site.	upload
1. Download this HTML verificat	tion file. [google4b93d2d3a1de58ff.html]
2. Upload the file to https://m.fla	shphoner.com/
	/ visiting https://m.flashphoner.com/google4b93d2d3a1de58ff.html in your browser.
 Click Verify below. To stay verified, don't remove the 	e HTML file, even after verification succeeds.
VERIFY Not now	

- If the verification is passed, a page confirming successful verification will be opened

🚡 Developer Dasht	board - C × 🕞 Search Console ×	1		×
← → C 🔒 h	https://www.google.com/webmasters/verification/verification-success?hl=er	n&site	Url=<	3 =
Google		0	s	Î
Search Cor	nsole	Help	*	
Congrat Continue	tulations, you have successfully verified your ownership of https://m.flashphoner.com/.			÷

- The website will appear in the list in the extension options and the extension can be associated with the site

C 🥚	https://chrome.google.com/webstore/developer/edit/hjjhafbcdlcbm	njkbjogdlhdofkpnggmj?hl=en&authus			
Websites	Verify that this is an official item for a website you own:	If your item is associated with a website that you own, select that website from the list. You can register as the owner for a site using Google Webmaster Tools. More info			
	https://m.flashphoner.com/				
	Add a new site Refresh list				
	Link to website for your item (optional)	Providing the URLs for description and suppor			
		pages can improve your users' experience and help make this item's ratings and comments			
	② Link to support & FAQ for your item (optional)	more meaningful. More info			

Configuring the client

Edit Screen-sharing.html and Screen-sharing.js

- In Screen-sharing.html the chrome-webstore-item parameter should point to your extension in the Chrome Store
- In Screen-sharing is replace the value of the chromeScreenSharingExtensionId parameter to the ID of your extension

To get the ID of the extension, click 'More info' of this extension inChrome Developer Dashboard.

🚊 Developer Dashboard - Cl. 🗙 🦲					= -		×
← → C 🔒 https://chrome.god	ogle.com/web:	store/develo	per/dashb	oard?hl=en		ŝ	≡
Flashphoner Screen Sharin Version 1.4	2/23/16	2/25/16	14	Published	Stats Unpublish Edit More info		
Target users in other languages. More	info 🔻				Item ID: nibaajpipmleofphigmgaifhoikjmbkg		

Media source parameters

To configure screen media source parameters, use parameters of the Configuration object passed to the init() method upon initializing of the Flashphoner API instance.

```
var f = Flashphoner.getInstance();
var configuration = new Configuration();
....
configuration.screenSharingVideoWidth = 1920;
configuration.screenSharingVideoHeight = 1080;
configuration.screenSharingVideoFps = 10;
f.init(configuration);
```

Parameter list

Parameter	Description				
screenSharingVideoWidth	Screen media source width				
screenSharingVideoHeight	Screen media source height				
screenSharingVideoFps	Screen media source framerate				

These parameters set marginal values of resolution and framerate (FPS). For instance, screenSharingVideoWidth = 1080 means the width of the source video cannot be more than 1080 pixels, but can be less. (i.e. when sending a stream of an app window that has the width of 720 pixels).

System sound capture in Chrome browser

In Chrome browser, there is ability to translate audio stream from system sound source while capturing a screen. The feature is useful in screencasting, for example. To capture system sound, set "Share audio" option in Chrome extension dialog window while choosing streaming source window or browser tab:

Your Entire Screen	Application Window	Chrome Tab
nation → <mark>O</mark> randaria <u>10 (m.</u>	ngaan angan • • ja ⊒ ⊒ 0 ⊒ ≟ 2 € 2	
Corecer Sharing The State of		
	de (Kanaka	
eriter e bess it no	1400 (H100)	

```
Chrome extension code:
```

callback({sourceId: sourceId, systemSoundAccess: opts.canRequestAudioTrack});

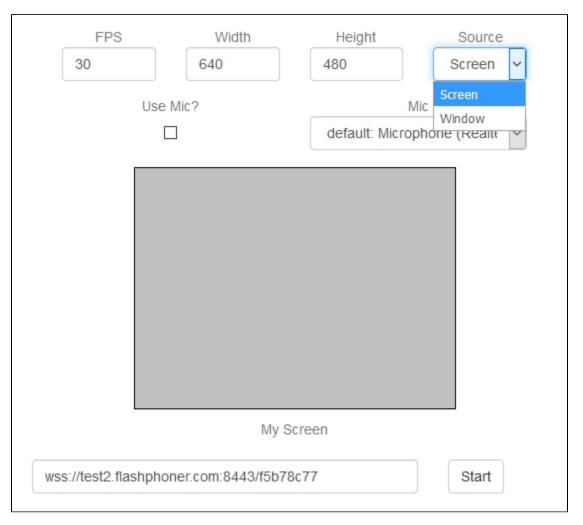
Capture source (screen or window) management in Firefox browser

In Firefox browser, all the screen or some program window can be chosen as video stream source with constraints.video.mediaSource parameter

code:

<pre>constraints.video.type = "screen";</pre>
if (Browser.isFirefox()){
<pre>constraints.video.mediaSource = \$('#mediaSource').val();</pre>
}
<pre>session.createStream({</pre>
name: streamName,
display: localVideo,
constraints: constraints
})

Source selection interface example:



Program window capture



Screen capture

			Screen	Sharing	g				
	FPS 30	Width 640	Height 480	Source Screen			C Screen Staring ← → C @	•× +	් පි 🔒 https://esi2.flat
	Us	e Mic?	M default: Microp					20 EDS	640 Use Mic?
	$\mathbf{v}_{\pm} = 0$	Screen Shering				>			
		The second secon						wss.iNest2 flash	My Scri iphone.com 64.836567627 PUBLISE
		My Sc	reen				< # 🕮 😭	9	9 🚯 🔍
WSS	://test2.flashph	noner.com:8443/f5b780	77	Stop					Preview
		PUBLIS	HING						

Screen sharing without extension

Firefox browser

Firefox browser does not use extension to share screen

Chromium based browsers

Since Chrome 73 and Flashphoner WebSDK0.5.28.2753.86screen sharing is possible without extension. To do this constraints.video.withoutExt ension parameter should be passed when stream is created

code

```
if ($("#woChromeExtension").prop('checked')) {
    constraints.video.withoutExtension = true;
}
```

Safari browser in MacOS

Since Safari 13and Flashphoner WebSDK0.5.28.2753.152screen sharing is possible without extension. To do this constraints.video.withoutExtension parameter should be passed when stream is created

code

```
if ($("#woChromeExtension").prop('checked') || Browser.isSafari()) {
    constraints.video.withoutExtension = true;
}
```

Known limits

1. In Chrome browser, picture resolution and FPS are set by source dimensions (screen, windows or browser tab) and by real picture updating speed, not by constraints/ This issue is fixed sinceFlashphoner WebSDK build0.5.28.2753.152

2. System sound capture only works starting from Crome 74

Code of the example

This example uses Flashphoner extensions for work with domain *.flashphoner.com. For work with your domain, build up and publish your extensions as described above.

As temporary solution, IP address of your WCS server can be added to file C:\Windows\System32\drivers\etc\hosts (on Windows OS) as test. flashphoner.com. That way, you will be able to test your WCS server with domain test.flashphoner.com till the extensions for your domain are built up.

With Chrome, the example works only by https://.

The path to the source code of the example on WCS server is:

/usr/local/FlashphonerWebCallServer/client/examples/demo/streaming/screen-sharing

```
screen-sharing.css - file with styles
screen-sharing.html - page of the screen sharing streamer
screen-sharing.js - script providing functionality for the streamer
```

This example can be tested using the following address:

https://host:8888/client/examples/demo/streaming/screen-sharing/screen-sharing.html

Here host is the address of the WCS server.

```
For Chrome, link to the extension is specified directly in file screen-sharing.htmlline 17
```

```
<link rel="chrome-webstore-item" href="https://chrome.google.com/webstore/detail
/nlbaajplpmleofphigmgaifhoikjmbkg">
```

Analyzing the code

1. Initialization of the API.

Flashphoner.init()code

Flashphoner.init({screenSharingExtensionId: extensionId});

2. Connection to WCS server.

Flashphoner.createSession()code

```
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function(session){
    //session connected, start streaming
    startStreaming(session);
}).on(SESSION_STATUS.DISCONNECTED, function(){
    setStatus(SESSION_STATUS.DISCONNECTED);
    onStopped();
}).on(SESSION_STATUS.FAILED, function(){
    setStatus(SESSION_STATUS.FAILED);
    onStopped();
});
```

3. Receiving the event confirming successful connection

ConnectionStatusEvent ESTABLISHEDcode

```
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function(session){
    //session connected, start streaming
    startStreaming(session);
}).on(SESSION_STATUS.DISCONNECTED, function(){
    ...
}).on(SESSION_STATUS.FAILED, function(){
    ...
});
```

4. Stream constraints setting

resolution and fpscode

```
var constraints = {
    video: {
        width: parseInt($('#width').val()),
        height: parseInt($('#height').val()),
        //WCS-2014. fixed window/tab sharing
        frameRate: parseInt($('#fps').val())
    }
};
```

michrophone usagecode

```
if ($("#useMic").prop('checked')) {
    constraints.audio = {
        deviceId: $('#audioInput').val()
    };
}
```

video source type and Chrome screen sharing without extensioncode

```
constraints.video.type = "screen";
if ($("#woChromeExtension").prop('checked')) {
    constraints.video.withoutExtension = true;
}
```

Firefox media sourcecode

```
if (Browser.isFirefox()){
    constraints.video.mediaSource = $('#mediaSource').val();
}
```

5. Video streaming

session.createStream(), publish()code

```
session.createStream({
    name: streamName,
    display: localVideo,
    constraints: constraints
    ...
}).publish();
```

6.Receiving the event confirming successful streaming

StreamStatusEvent PUBLISHINGcode

When the screen sharing stream is published, preview video stream is created with method session.createStream(), and function play() is called to start playback of the stream in <div> element 'remoteVideo'.

```
session.createStream({
   name: streamName,
   display: localVideo,
   constraints: constraints
}).on(STREAM_STATUS.PUBLISHING, function(publishStream){
   /*
    * User can stop sharing screen capture using Chrome "stop" button.
     * Catch onended video track event and stop publishing.
     */
   document.getElementById(publishStream.id()).srcObject.getVideoTracks()[0].onended = function (e) {
       publishStream.stop();
    };
   document.getElementById(publishStream.id()).addEventListener('resize', function(event){
       resizeVideo(event.target);
   });
   setStatus(STREAM_STATUS.PUBLISHING);
   //play preview
   session.createStream({
       name: streamName,
       display: remoteVideo
        . . .
    }).play();
}).on(STREAM_STATUS.UNPUBLISHED, function(){
   . . .
}).on(STREAM_STATUS.FAILED, function(){
    . . .
}).publish();
```

7. Receiving the event confirming successful preview stream playback

StreamStatusEvent PLAYINGcode

```
session.createStream({
    name: streamName,
    display: remoteVideo
}).on(STREAM_STATUS.PLAYING, function(previewStream){
    document.getElementById(previewStream.id()).addEventListener('resize', function(event){
        resizeVideo(event.target);
    });
    //enable stop button
    onStarted(publishStream, previewStream);
}).on(STREAM_STATUS.STOPPED, function(){
    ...
}).on(STREAM_STATUS.FAILED, function(){
    ...
}).play();
```

8. Preview stream playback stop

stream.stop()code

```
function onStarted(publishStream, previewStream) {
   $("#publishBtn").text("Stop").off('click').click(function(){
      $(this).prop('disabled', true);
      previewStream.stop();
   }).prop('disabled', false);
}
```

9. Receiving the event confirming successful playback stop

StreamStatusEvent STOPPEDcode

On receiving the event, publishStream.stop() is called to stop screen sharing streaming

```
session.createStream({
    name: streamName,
    display: remoteVideo
}).on(STREAM_STATUS.PLAYING, function(previewStream){
    ...
}).on(STREAM_STATUS.STOPPED, function(){
    publishStream.stop();
}).on(STREAM_STATUS.FAILED, function(){
    ...
}).play();
```

10. Screen sharing streaming stop by click on Chrome extension button

publishStream.stop().code

```
document.getElementById(publishStream.id()).srcObject.getVideoTracks()[0].onended = function (e) {
    publishStream.stop();
};
```

11. Receiving the event confirming successful streaming stop

StreamStatusEvent UNPUBLISHEDcode

```
session.createStream({
    name: streamName,
    display: localVideo,
    constraints: constraints
}).on(STREAM_STATUS.PUBLISHING, function(publishStream){
    ...
}).on(STREAM_STATUS.UNPUBLISHED, function(){
    setStatus(STREAM_STATUS.UNPUBLISHED);
    //enable start button
    onStopped();
}).on(STREAM_STATUS.FAILED, function(){
    ...
}).publish();
```