

Логирование

Логгер по умолчанию, включение и отключение

По умолчанию, лог выполнения операций WebSDK выводится в консоль браузера с уровнем INFO

09:46:56 INFO webrtc - Initialized	flashphoner.js:35631
09:46:56 INFO websocket - Initialized	flashphoner.js:35631
09:46:56 INFO core - Initialized	flashphoner.js:35631

Начиная со сборки [0.5.28.2753.131](#) (исходный код на GitHub доступен по тэгу [05cb5bd](#)), логирование может быть полностью отключено на этапе инициализации API

```
Flashphoner.init({flashMediaProviderSwfLocation: '../../../../../media-provider.swf', logger: null});
```

или в коде приложения вызовом функции

```
Flashphoner.getLogger().setEnableLogs(false);
```

Затем, при необходимости, логирование может быть включено

```
Flashphoner.getLogger().setEnableLogs(true);
```

Изменение уровня логирования

Уровень логирования может быть изменен на этапе инициализации API

```
Flashphoner.init({flashMediaProviderSwfLocation: '../../../../../media-provider.swf', logger: {severity: "WARN"}});
```

или в коде приложения вызовом функции

```
Flashphoner.getLogger().setLevel("WARN");
```

Поддерживаются следующие уровни логирования:

Level text constant	Description
---------------------	-------------

Level text constant	Description
ERROR	Только ошибки
WARN	Ошибки и предупреждения
INFO	Информация о работе функций WebS DK (по умолчанию)
DEBUG	Отладочная информация
TRACE	Трассировка выполнения

Отправка лога на сервер

По умолчанию, лог клиента не отправляется на сервер. При необходимости, эта возможность может быть включена на этапе инициализации API

```
Flashphoner.init({flashMediaProviderSwfLocation: '../..../media-provider.swf', logger: {push: true}});
```

или в коде приложения вызовом функции

```
Flashphoner.getLogger().setPushLogs(true);
```

В этом случае лог клиента будет отправлен на WCS сервер по Websocket и будет передан сервером на бэкэнд при помощи [REST hook /pushLogs](#):

```
10:16:03,335 INFO          RestClient - API-ASYNC-pool-12-thread-5 SEND
REST OBJECT ==>
URL:http://localhost:8081/apps/EchoApp/pushLogs
OBJECT:
{
  "nodeId" : "vdUfWbDQUa9TIFfYSwGmXhDs3zp1vH4p@192.168.0.111",
  "appKey" : "defaultApp",
  "sessionId" : "/192.168.0.100:50627/192.168.0.111:8443-966a2082-8033-4982-9bcb-fecc1bd46169",
  "logs" : "10:15:55 INFO webrtc - \"Initialized\\n\",
  "origin" : "https://test.flashphoner.com:8888"
}
```

При закрытии Websocket соединения лог клиента будет выведен в серверный лог:

 [Client log example in server logs](#)



Использование собственного логгера

Начиная со сборки [0.5.28.2753.131](#) (исходный код на GitHub доступен по тэгу [05cb5bd](#)), можно определить собственный логгер

```
var customLogger = {
  error: function (text) {
    console.log("custom logger: ERROR:", text);
  },
  warn: function (text) {
    console.log("custom logger: WARN:", text);
  },
  info: function (text) {
    console.log("custom logger: INFO:", text);
  },
  debug: function (text) {
    console.log("custom logger: DEBUG:", text);
  },
  trace: function (text) {
    console.log("custom logger: TRACE:", text);
  }
};
```

и указать его при инициализации API

```
Flashphoner.init({flashMediaProviderSwfLocation: '../.../.../media-provider.swf', logger: {customLogger: customLogger}});
```

или в коде приложения вызовом функции

```
Flashphoner.getLogger().setCustomLogger(customLogger);
```

Пример логгера, приведенный выше, выведет в консоль браузера

custom logger: INFO: Initialized	two_way_streaming.js:15
custom logger: INFO: Initialized	two_way_streaming.js:15
custom logger: INFO: Initialized	two_way_streaming.js:15
Create new session with url wss://test1.flashphoner.com:8443	two_way_streaming.js:54
custom logger: INFO: ▶ {audio: true, video: {...}, customStream: undefined}	two_way_streaming.js:15
custom logger: INFO: FOUND WEBRTC CACHED INSTANCE, id 8185c8c0-2081-11ea-b1ef-9de17262c752-LOCAL_CACHED_VIDEO	two_way_streaming.js:15

Для собственного логгера, как и для логгера по умолчанию, работают включение, отключение, изменение уровня логирования и отправка лога на сервер.

Определение параметров логирования для сессии, потока или звонка

В сборке WebSDK 2.0.215 добавлена возможность определять параметры логирования не только для приложения в целом, но и для сессии, потока или звонка. При этом на каждом из уровней используется свой экземпляр объекта логирования. Например, создадим собственные логгеры отдельно для публикации и проигрывания в приложении Two Way Streaming:

1. Определим логгеры для публикации

```
var publishCustomLogger = {
  error: function (text) {
    console.log("publish: ERROR:", text);
  },
  warn: function (text) {
    console.log("publish: WARN:", text);
  },
  info: function (text) {
    console.log("publish: INFO:", text);
  },
  debug: function (text) {
    console.log("publish: DEBUG:", text);
  },
  trace: function (text) {
    console.log("publish: TRACE:", text);
  }
};
```

и для проигрывания

```
var playCustomLogger = {
  error: function (text) {
    console.log("play: ERROR:", text);
  },
  warn: function (text) {
    console.log("play: WARN:", text);
  },
  info: function (text) {
    console.log("play: INFO:", text);
  },
  debug: function (text) {
    console.log("play: DEBUG:", text);
  },
  trace: function (text) {
    console.log("play: TRACE:", text);
  }
};
```

2. Указываем собственный логгер при создании потока для публикации

```
function publishStream() {
  ...
  session.createStream({
    name: streamName,
    display: localVideo,
```

```

    ...
    logger: {customLogger: publishCustomLogger}
    ...
  }).publish();
}

```

и для проигрывания

```

function playStream() {
  ...
  session.createStream({
    name: streamName,
    display: remoteVideo,
    logger: {customLogger: playCustomLogger}
    ...
  }).play();
}

```

3. При публикации и проигрывании потока в модифицированном примере Two Way Streaming в консоль будут выведены сообщения от логгеров

13:09:10	INFO	webrtc - Initialized	flashphoner.js:14432
13:09:10	INFO	websocket - Initialized	flashphoner.js:14432
13:09:10	INFO	core - Initialized	flashphoner.js:14432
		Failed to load resource: the server responded with a status of 404 (Not Found)	:8081/favicon.ico:1
		Create new session with url ws://localhost:8080	two_way_streaming.js:75
13:09:22	INFO	webrtc - ▶ {audio: true, video: {...}, customStream: undefined}	flashphoner.js:14432
		publish: INFO: FOUND WEBRTC CACHED INSTANCE, id c9433740-aa6f-11ec-b31c-1321b67d9aed-LOCAL_CACHED_VIDEO	two_way_streaming.js:19
		publish: INFO: Set video track contentHint to detail	two_way_streaming.js:19
		GET http://localhost:8081/favicon.ico 404 (Not Found)	favicon.ico:1
		Resize from 320x240 to 320x240	utils.js:185
		play: INFO: FOUND WEBRTC CACHED INSTANCE, id cd0a2a50-aa6f-11ec-b31c-1321b67d9aed-REMOTE_CACHED_VIDEO	two_way_streaming.js:36
		Resize from 320x240 to 320x240	utils.js:185