



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Web Real-Time Communications Deployment Guide

Setting up STUN and TURN servers

Setting up STUN and TURN servers

Depending on the type of deployment model, either a STUN server or a TURN server may need to be deployed.

There are open source projects that provide working implementations of STUN and TURN servers. The following projects are known to be compatible with the Genesys WebRTC Service and Google Chrome:

- STUN—<http://sourceforge.net/projects/stun/>
- TURN—<https://code.google.com/p/rfc5766-turn-server/>

In order for a STUN or TURN server to work with the WebRTC Service when using the WebRTC JavaScript API, its URI must be provided in the configuration parameters.

STUN example:

```
var configuration = {
  'webrtc_gateway': 'http://webrtc.genesyslab.com:8080',
  'stun_server': 'stun.genesyslab.com:3478'
};
var grtcClient = new Grtc.Client(configuration);
```

Access to a TURN Server requires the use of a credential. Note that this credential is employed to ensure that the TURN server hosted on the Internet only provides access to your own web applications, which must use these credentials. Unauthorized web applications that do not have access to the credentials will not be able to request TURN bindings.

```
var configuration = {
  'webrtc_gateway': 'http://WebRTC.genesyslab.com:8080',
  'turn_server': 'stun.genesyslab.com:3478',
  'turn_username': 'genesys',
  'turn_password': 'password'
};
var grtcClient = new Grtc.Client(configuration);
```