

Creating a JSGF Grammar File

Syn Speech has a good support for JSGF files (unlike SRGS which is partially supported). The JSGF file parser is extremely sensitive to file encoding and characters. Any illegal character in the file may render the entire grammar useless.

To create a JSGF Grammar file a developer should first create a file with the name of the Grammar he intends to use. For example if your Grammar file contains sentences like “**hi there**”, “**whats up**” and “**how are you**”. You may name your grammar file `hello.gram` where `hello` is the name of the grammar and `.gram` is the extension of the grammar file.

It is highly recommended that you use lower-case naming convention when it comes to grammar files and names. This prevents any *File not found* exception from occurring in different platforms where file names may be case-sensitive (Linux).

The following demonstrates a grammar file that helps the speech recognizer recognize the aforementioned sentences (*hi there, whats up, how are you*).

```
#JSGF V1.0;  
grammar hello;  
public <command> = ( hi there | whats up | how are you );
```

Grammar Header

The first line is the grammar header and specifies the version of JSGF. The grammar header should always begin with **#JSGF** followed by the version.

Grammar Name

The second line declares the name of the grammar i.e. `hello`. The format is **grammar grammar-name;**. Every grammar file should declare a grammar name. So for example if you wish to name your grammar `helloSpeech`. You'll do so by typing **grammar helloSpeech;**

Grammar Body

COMMUNITY LOBBY ([HTTPS://GITTER.IM/SYNCOMMUNITY/LOBBY](https://gitter.im/syncommunity/lobby))

The last line is the Grammar body where a grammar rule is defined. In the above example we've created a simple grammar for 3 sentences. Each sentence separated by a vertical bar and a space character. All the sentences are then encapsulated within round brackets.

Grammar body declaration format: **public = ruleExpansion;**

An example would be **public = hello;** where name of the rule and hello is the expansion. It is important to note that a single JSGF Grammar file must contain just 1 Grammar. This is one of the limitations of JSGF.

Individual tokens in a grammar can be separated using vertical bars (|)

Example **public = hello | hey;**. So now if the user says either hello or hi. The speech will be recognized immediately.

You can even glue words or sentences using parenthesis.

Example **public = hey (machine | computer);** which will match inputs like hey machine and hey computer.

JSGF also has optional grouping via square brackets. For example if you want a word or sentence to be optional you can group it within a square bracket.

Example **public = [Hello] World;** This will match the inputs hello world or world.

Comments

To add single line comment in your grammar file use double forward-slashes. Example **// Single lined comment**

If you want to add multiple lines comment enclose the comment within **/* */**.

If you intend to use grammar heavily with Syn Speech we recommend you going through W3C's JSGF specification (<http://www.w3.org/TR/jsgf/>)

COMMUNITY LOBBY ([HTTPS://GITTER.IM/SYNCOMMUNITY/LOBBY](https://gitter.im/syncommunity/lobby))