Basics of Java Speech Grammar Format (JSGF)

by Puneet Kalra | Mar 12, 2010 | All, Tutorials | 21 comments

Hello Everyone,

This post is my response to those 4 help requests that i received in last few days. This one is going to be very basic and essentials of Java Speech Grammar Format (JSGF).

Every single file defines a single grammar only. Each grammar contains two parts:

The **grammar header** and the **grammar body**.

Grammar header format : **#JSGF version char-encoding locale;**

"#JSGF" is required and "version char-encoding locale;" is optional.

Grammar header example : **#JSGF V1.0;** & **#JSGF V1.0 ISO8859-5 en;**

After declaring Grammar Header, We need to specify the Grammar name.

Grammar name format : **grammar grammarName;**

This is a mandatory line. Else javax.speech.recognition.GrammarException will be thrown. Not only on Grammar Name, you will get GrammarExpection If you made any kind of mistake in grammar file.

Grammar Name Example : **grammar helloWorld;**

Once you are done with above part, you next step will be defining Grammar body.

The grammar body defines rules. you can define a rule only once. If you declare same rule twice then it will overwritten.

Rule Definition Format : **public <ruleName> = ruleExpansion;**

"public" is optional and remaining part is mandatory.

The rule expansion defines how the rule may be spoken. It is a logical combination of tokens (text that may be spoken) and references to other rules.

Rule Example : **public <greet> = Hello;**

The rule **<greet>** refers to a single token **Hello**. So to say rule **<greet>**, User must say word "Hello".

A simple rule expansion can refer to one or more tokens or rules.

public <greet> = Hello;

public <completeGreet> = <greet> World;

Now, rule **<completeGreet>** refers to rule **<greet>** and token **World**. To say rule **<completeGreet>**, User must say “Hello World”.

Lets complete a simple “Hello World” grammar file.

#JSGF V1.0;

grammar simpleExample;

public <greet> = Hello;
public <completeGreet> = <greet> World;

This grammar file will allow you SR application to recognize 2 sentences. “Hello” and “Hello World”.

Now lets play a little bit with rule expansions.

Alternatives : “|”

public <greet> = Hello | Hey | Hi;

To say rule **<greet>**, User must say “Hello” or “Hey” or “Hi” But ONLY one of these three words.

Parentheses : “( )”

public <greet> = Hello (World | User | Friend);

public <command> = (Open | Close) (Door | Window);

To say rule **<greet>**, User must say “Hello World” or “Hello User” or “Hello Friend”.

And, to say rule **<command>**, User must say “Open Door” or “Open Window” or “Close Door” or “Close Window”.

Optional Grouping : “[ ]”

public <greet> = [Hello] World;

To say rule **<greet>**, User must say “Hello World” or “World”. As “Hello” is defined inside the Optional

Grouping. So user may say it or not but "World" is mandatory.

Kleene Star : "*"

public <greet> = (Hello | Hey | Hi)* World;

Any group or expansion followed by asterisk symbol indicates that it may be spoken zero or more times. For example "Hey Hello World" or "World" or "Hello Hello Hello World".

Plus Operator : "+"

public <greet> = (Hello | Hey | Hi)+ World;

A Plus Operator works same as "Kleene Star", The only thing that makes difference is any group or expansion followed by plus symbol indicates that it may be spoken one (NOT ZERO) or more times.

You can also add comments in grammar file.

// One line comment

/* Multiple lines comment*/

/**

*** Documentation comment**

*** @author Puneet Kalra**

***/**

Hope this tutorial clears all your doubts on JSGF.

More updates to come soon!

Regards,