Ananya
MoTeCH - IVR Services

Revisions:

No	Date	Author
1	16/03/2012	R Vijayaraj

Ananya IVR Services

The following are the remote services used by the existing IVR callflows.

1. DynamicJsController
2. CertificateCourseCallDataController
3. JobAidCallStateController

VoiceXML callflows executing on OnMobile Platform talk with these MoTeCH services via HTTP protocol.
These services provide and receive data over the course of a call flow. The upcoming sections will describe the methods in detail. The current call flows work as a combination of voicexml, javascript and these services hosted by Motech.

1. DynamicJsController

1.1 Provide caller state details for certificate course call flow.

This method provides the information regarding the caller current state in Certificate Course. It provides the registration status, bookmark location and current scores.

URL: http://motech-server/ananya/generated/js/dynamic/caller_data.js?callerId={msisdn}&operator={operator}
Method: HTTP GET

Request Parameters:
a) callerId - msisdn of caller.
b) operator - operator of caller.

Response: JSON String. Sample shown below.

```
var callerData = {
  "isRegistered" : "true",
  "bookmark" : {"type":"lesson","chapterIndex":3,"lessonIndex":0},
  "scoresByChapter" : { "0" : 2, "1" : 1, "2" : 0 }
};
```

1.2 Provide caller state details for jobaid call flow.

This method provides the information regarding the caller current state in Jobaid.
It provides the registration status, current usage in milliseconds, maximum allowed usage in milliseconds and the counts of prompts heard.

URL: http://motech-server/ananya/generated/js/dynamic/caller_data.js?callerId={msisdn}&operator={operator}
Method: HTTP GET

Request Parameters:
a) callerId - msisdn of caller.
b) operator - operator of caller.

Response: JSON String. Sample shown below.

```
var callerData = {
   "isRegistered" : "false",
   "currentJobAidUsage" : 1230000,
   "maxAllowedUsageForOperator" : 2340000,
   "promptsHeard" : { "0001_prep_for_child_birth.wav":2, "0005_inst_delivery.wav":1 }
};
```

1.3 Provide Certificate Course datastructure.

This method provides the entire certificate course as a tree-style data structure. This datastructure drives the existing call flow navigation via javascript and the caller data fetched (*ref* method 1.1).

URL: http://motech-server/ananya/generated/js/certification_course_data.js
Method: HTTP GET

Request Parameters: NIL
Response: JSON String. Sample shown below.

```
var courseData =
{"name":"CertificationCourse","data":{"type":"course"},
"children":[{"name":"Chapter 1","data":{"type":"chapter"},
        "children":[
                {
                "name":"Chapter 1 Lesson 1",
                "data":{"type":"lesson"},
                "children":[],
                "contents":[
                        {"value":"0003_ch1_l1.wav","language":"hindi","name":"lesson","type":"StringContent"},
                        {"value":"0004_ch1_l1_op.wav","language":"hindi","name":"menu","type":"StringContent"}],
                        "childrenIds":[],
                        "type":"Node"
                },
                {
                "name":"Chapter 1 Lesson 2","data":{"type":"lesson"},
                "children":[],
                "contents":[
                        {"value":"0005_ch1_l2.wav","language":"hindi","name":"lesson","type":"StringContent"},
                        {"value":"0006_ch1_l2_op.wav","language":"hindi","name":"menu","type":"StringContent"}],
                "childrenIds":[],
                "type":"Node"
                },
                {"name":"Chapter 1 Lesson 3"
      ....
}
```

1.4 Provide Jobaid Level datastructure

This method provides the 'Level' of jobaid course as a tree-style datastructure. This datastructure is an aggregation of the jobaid lessons and menus belonging to a particular level. Each node in the datastructure comprises of the audio to be played, the shortcode without the basenumber and duration of audio.

URL: http://motech-server/ananya/generated/js/jobaid_level_data.js?levelNumber={level_number}
Method: HTTP GET

Request Parameters:
levelNumber - Jobaid level number (1 to 5)

Response: JSON String. Sample shown for Level 1 below.

```
courseData.children[0] =
 {"name":"level 1","data":{"number":"1","type":"Level"},
"children":[
        {"name":"Level 1 Chapter 1",data":{"number":"1","type":"Chapter"},
        "children":[
                {"name":"Level 1 Chapter 1 Lesson1","data":{"shortcode":"11","number":"1","type":"Lesson"},
                "children":[],
                "contents":[{"value":"0001_prep_for_child_birth.wav","language":"hindi","name":"lesson","metadata":
{"duration":"96734"},"type":"StringContent"}],
                "childrenIds":[],
                "type":"Node"},
                {"name":"Level 1 Chapter 1 Lesson2","data":{"shortcode":"12","number":"2","type":"Lesson"},
                "children":[],
                "contents":[{"value":"0002_ifa_and_tt.wav","language":"hindi","name":"lesson","metadata":
{"duration":"120453"},"type":"StringContent"}],
                "childrenIds":[],"type":"Node"}
                .... }
```

1.5 Provide Jobaid Root datastructure

This method provides only the top level node of jobaid course as a tree-style data structure. This datastructure drives the existing call flow navigation via javascript and the caller data fetched (*ref* method 1.2). Jobaid levels data fetched (*ref* method 1.4) are attached to this root node.

URL: http://motech-server/ananya/generated/js/jobaid_course_data_without_levels.js
Method: HTTP GET

Request Parameters: NIL
Response: JSON String. Sample shown below.

```
var courseData = {"name":"JobAidCourse","data":{"type":"Level"},"children":[],"contents":
[{"value":"0002_select_level.wav","language":"hindi","name":"menu","metadata":
{"duration":"42787"},"type":"StringContent"}],"childrenIds":[],"type":"Node"};
```

2. CertificateCourseCallDataController

2.1 Receive Call Data for Certificate Course

This method receives call data as json during certificate course call flow. In the current call flow, it is called multiple times to send these data tokens to the motech server.

URL: http://motech-server/ananya/transferdata/
Method: HTTP POST

Request Parameters:

callId	-	unique call id
callerId	-	msisdn of the caller.
dataToPost	-	json token string. Sample shown below

```
[
{"token":0,"type":"callDuration","data":{"callEvent":"CALL_START","time":1331211295810}},
{"token":1,"type":"callDuration","data":{"callEvent":"CERTIFICATECOURSE_START","time":1331211297476}},
{"token":2,"type":"ccState",
"data":"result":null,"questionResponse":null,"contentId":"0cccd9b516233e4bb1c6c04fed6a66d5","contentType":"lesson","
certificateCourseId":"","contentData":null,"interactionKey":"lessonEndMenu","courseItemState":"end","contentName":"Ch
apter 1 Lesson 1","time":"2012-03-08T12:54:57Z","chapterIndex":0,"lessonOrQuestionIndex":0}}
];
```

Response: String

2.1 Receive Call Data for Certificate Course Disconnect

This method receives call data as json during disconnect event of certificate course call flow. In the current call flow, it is called only once.

URL: http://motech-server/ananya/transferdata/disconnect/
Method: HTTP POST

Request Parameters:

callId	-	unique call id
callerId	-	msisdn of the caller.
dataToPost	-	json token string. Sample shown below

```
 [{"token":28,"type":"callDuration","data":{"callEvent":"DISCONNECT","time":1331211652263}}];
```

Response: String

3. JobAidCallStateController

3.1 Receive Prompts played data for JobAid

This method receives updates regarding the number of times an audio prompt is played during jobaid callflow.

URL: http://motech-server/ananya/jobaid/updateprompt?callId={callId}&callerId={callerId}&promptList=['prompt1', 'prompt2']"
Method: HTTP GET

Request Parameters:

callId	-	unique call id
callerId	-	msisdn of the caller.
promptList	-	json string array of prompt names.

Response: String

3.2 Receive updated caller usage for JobAid

This method receives updates on the current usage of the caller.

URL: http://motech-server/ananya/jobaid/updateusage?callId={callId}&callerId={callerId}&usage={usage}
Method: HTTP GET

Request Parameters:

callId	-	unique call id
callerId	-	msisdn of the caller.
usage	-	current usage of the caller.

Response: String