

雑談対話API インタフェース仕様書

第 1.0.6 版

2015年12月1日
株式会社NTTドコモ



改版履歴

日付	版	変更内容
2013/11/11	1.0	初版
2013/11/28	1.0.1	2.2.1. プロトコルの URL 名を訂正 2.2.2. クエリ・パラメータのキー名を訂正
2014/3/27	1.0.2	2.2.3. リクエストボディ部に上限値を追記 2.2.4. リクエスト・サンプル内の URL を訂正 2.5. HTTP ステータスコード一覧を訂正
2014/12/5	1.0.3	2.2.3. リクエストボディ部に新パラメータを追記 3.6 キャラクタの設定について追記
2015/1/6	1.0.4	2.2.2. クエリ・パラメータのキー名を訂正
2015/2/5	1.0.5	2.2.1. d アカウント有のプロトコルを追記
2015/12/1	1.0.6	2.2.3 リクエストボディ部の context について追記 2.2.3 リクエストボディ部の birthdateD について追記 「docomo ID」の名称を「d アカウント」に変更

目次

1.	はじめに	1
2.	雑談対話 API 仕様	2
2.1.	共通仕様	2
2.2.	リクエスト仕様	2
2.2.1.	プロトコル	2
2.2.1.1.	d アカウント利用無	2
2.2.1.2.	d アカウント利用有	2
2.2.2.	クエリ・パラメータ	2
2.2.3.	リクエスト・ボディ部	2
2.2.4.	リクエスト・サンプル	4
2.3.	レスポンス仕様	5
2.3.1.	レスポンス・ボディ部	5
2.3.2.	レスポンス・サンプル	5
2.4.	場所リスト	6
2.5.	HTTP ステータスコード一覧	6
3.	雑談対話 API を利用した会話例	7
3.1.	会話の継続	7
3.2.	システムへの質問	8
3.3.	ユーザのニックネーム等の利用	9
3.4.	しりとり開始	10
3.5.	しりとりの継続	11
3.6.	キャラクタの設定	12

1. はじめに

「雑談対話 API」は、ユーザの発話(テキスト)に対して返答を行う機能である。

発話(テキスト)を株式会社 NTT ドコモ(以下、ドコモ)の提供する雑談対話サーバ上の API(以下、雑談対話 API)に送信することで、発話に対する返答を取得できる。

下記に雑談対話 API を用いたアプリの会話例(図1)を示す。

本書はこの雑談対話機能における HTTP インタフェースに関する仕様を示したものである。

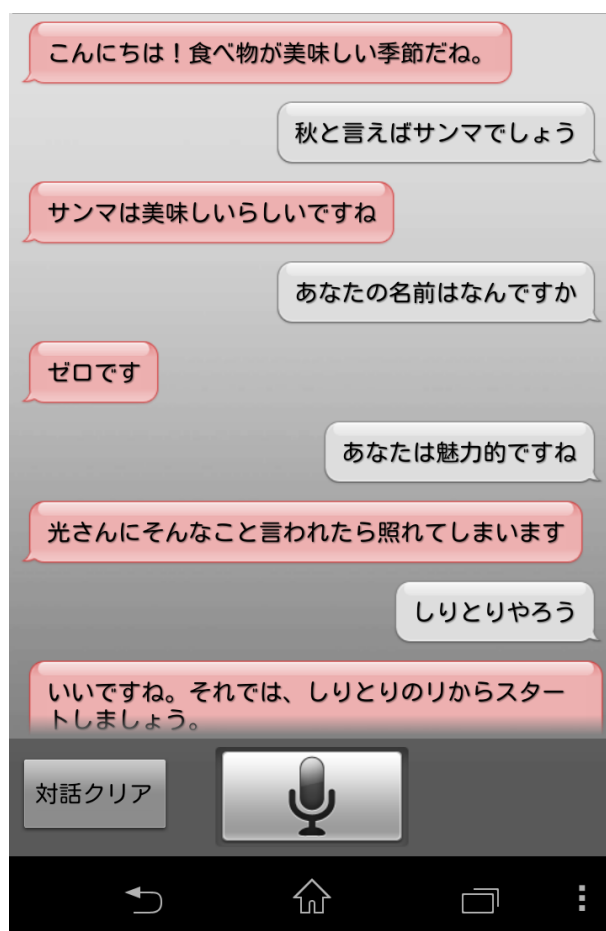


図1. 雑談対話 API を用いたアプリでの会話例

2. 雑談対話 API 仕様

本章では雑談対話 API の利用にあたってのリクエスト/レスポンスの仕様を記載する。

2.1. 共通仕様

プロトコル	HTTP
文字コード	UTF-8
Content-type	application/json

2.2. リクエスト仕様

2.2.1. プロトコル

2.2.1.1. d アカウント利用無

URL	https://api.apigw.smt.docomo.ne.jp/dialogue/v1/dialogue
メソッド	POST

2.2.1.2. d アカウント利用有

URL	https://api.apigw.smt.docomo.ne.jp/dialogue/v2/dialogue
メソッド	POST

2.2.2. クエリ・パラメータ

項番	項目名	キー	型	必須	説明	備考
1	API キー	APIKEY	string	○	API キー。アプリケーションに対してユニークに付与	

2.2.3. リクエスト・ボディ部

項番	項目名	キー	型	必須	説明	備考
1	ユーザ発話	utt	string	○	ユーザの発話 設定例「こんにちは」	255 文字以下
2	コンテキスト ID	context	string	—	システムから出力された context を入力することにより会話を継続 d アカウントを利用する場合は、最初は必ず空で送信し、その後はシステムから出力された context を利用すること	255 文字以下

					設定例「aaabbb111222」	
3	ユーザのニック ネーム	nickname	string	—	ユーザのニックネームを設 定 設定例「光」	全角 10 文字(半角 10 文字)以下
4	ユーザのニック ネームの読み	nickname_y	string	—	ユーザのニックネームの読 みを設定 設定例「ヒカリ」	全角 20 文字以下 (カタカナのみ)
5	ユーザの性別	sex	string	—	ユーザの性別を設定 設定例「女」	男または女
6	ユーザの血液 型	bloodtype	string	—	ユーザの血液型を設定 設定例「B」	A、B、AB、O のいず れか
7	ユーザの誕生 日(年)	birthdateY	int	—	ユーザの誕生日(年)を設定 設定例「1997」	1～現在の都市まで のいずれかの整数
8	ユーザの誕生 日(月)	birthdateM	int	—	ユーザの誕生日(月)を設定 設定例「5」	1～12 までのいず れかの整数
9	ユーザの誕生 日(日)	birthdateD	int	—	ユーザの誕生日(日)を設定 設定例「30」 birthdateD を入力する場 合は、birthdateM も入力必須	1～31 までのいず れかの整数
10	ユーザの年齢	age	int	—	ユーザの年齢を設定 設定例「16」	正の整数
11	ユーザの星座	constellations	string	—	ユーザの星座を設定 設定例「双子座」	牡羊座、牡牛座、双 子座、蟹座、獅子 座、乙女座、天秤 座、蠍座、射手座、 山羊座、水瓶座、魚 座のいずれか
12	ユーザの場所	place	string	—	ユーザの地域情報を設定 設定例「東京」	2.4「場所リスト」に含 まれるもののいず れか
13	モード	mode	string	—	システムから出力された mode を入力することによりし りとりを継続 設定例「dialog」	dialog または srtr
14	キャラクタ ID	t	int	—	キャラクタを設定 なし: デフォルト	20 または 30 で関西 弁または赤ちゃんキ

					20: 関西弁 30: 赤ちゃん	キャラクタ設定。t の設定なしでデフォルトキャラとなる
--	--	--	--	--	---------------------	-----------------------------

2.2.4. リクエスト・サンプル

```
POST/dialogue/v1/dialogue?APIKEY=<APIキー> HTTP/1.1
```

```
Host: api.apigw.smt.docomo.ne.jp
```

```
Connection: keep-alive
```

```
Content-Length: 28
```

```
Accept: application/json
```

```
Origin: <リクエスト生成元URI>
```

```
User-Agent: <各クライアント>
```

```
Content-Type: application/json
```

```
Accept-Encoding: gzip,deflate,sdch
```

```
Accept-Language: en-US,en;q=0.8
```

```
{
  "utt": "あいうえお"
}
```

2.3. レスポンス仕様

2.3.1. レスポンス・ボディ部

項番	項目名	キー	型	必須	説明	備考
1	システムの返 答	utt	string	—	システムからの返答 返答例「こんにちは光さん」	
2	音声合成用読 み出力	yomi	string	—	音声合成にシステム返答を読ませるための出力 返答例「こんにちはヒカリさん」	
3	モード	mode	string	—	対話のモード ※この値をそのまま入力とする 返答例「dialog」	
4	対話番号	da	int	—	ユーザとシステムの対話に対してサーバが付与し た番号 返答例「0」	
5	コンテキスト ID	context	string	—	自動的にシステムより出力される ID ※この値をそのまま入力とする 返答例「aaabbb111222」	

2.3.2. レスポンス・サンプル

```

HTTP/1.1 200 OK
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: <リクエスト生成元URI>
asyncServiceInvoke: false
Content-Type: application/json;charset=UTF-8
Date: Wed, 06 Nov 2013 02:18:46 GMT
Server: Apache-Coyote/1.1
X-Powered-By: Servlet 2.5; JBoss-5.0/JBossWeb-2.1
Content-Length: 104
Connection: keep-alive

{
  "utt": "かきくけこ",
  "yomi": "かきくけこ",
  "mode": "dialog",
  "da": "2",
  "context": "v-2Wcqz6fDAcGqJZvpgM5A"
}

```


2.4. 場所リスト

稚内	山形	銚子	福井	広島	中津
旭川	米沢	館山	大野	呉	日田
留萌	酒田	横浜	敦賀	福山	佐伯
網走	新庄	小田原	金沢	庄原	熊本
北見	仙台	甲府	輪島	下関	阿蘇乙姫
紋別	古川	河口湖	大津	山口	牛深
根室	石巻	長野	彦根	柳井	人吉
釧路	白石	松本	津	萩	宮崎
帯広	福島	諏訪	上野	高松	油津
室蘭	郡山	軽井沢	四日市	徳島	延岡
浦河	白河	飯田	尾鷲	池田	都城
札幌	小名浜	静岡	京都	日和佐	高千穂
岩見沢	相馬	網代	舞鶴	松山	鹿児島
倶知安	若松	石廊崎	奈良	新居浜	阿久根
函館	田島	三島	風屋	宇和島	枕崎
江差	宇都宮	浜松	和歌山	高知	鹿屋
青森	大田原	御前崎	潮岬	室戸岬	種子島
弘前	水戸	新潟	大阪	清水	名瀬
深浦	土浦	津川	神戸	福岡	沖永良部
むつ	前橋	長岡	姫路	八幡	那覇
八戸	みなかみ	湯沢	洲本	飯塚	名護
秋田	さいたま	高田	豊岡	久留米	久米島
横手	熊谷	相川	鳥取	佐賀	南大東島
鷹巣	秩父	富山	米子	伊万里	宮古島
盛岡	東京	伏木	岡山	長崎	石垣島
二戸	大島	岐阜	津山	佐世保	与那国島
一関	八丈島	高山	松江	厳原	海外
宮古	父島	名古屋	浜田	福江	
大船渡	千葉	豊橋	西郷	大分	

2.5. HTTP ステータスコード一覧

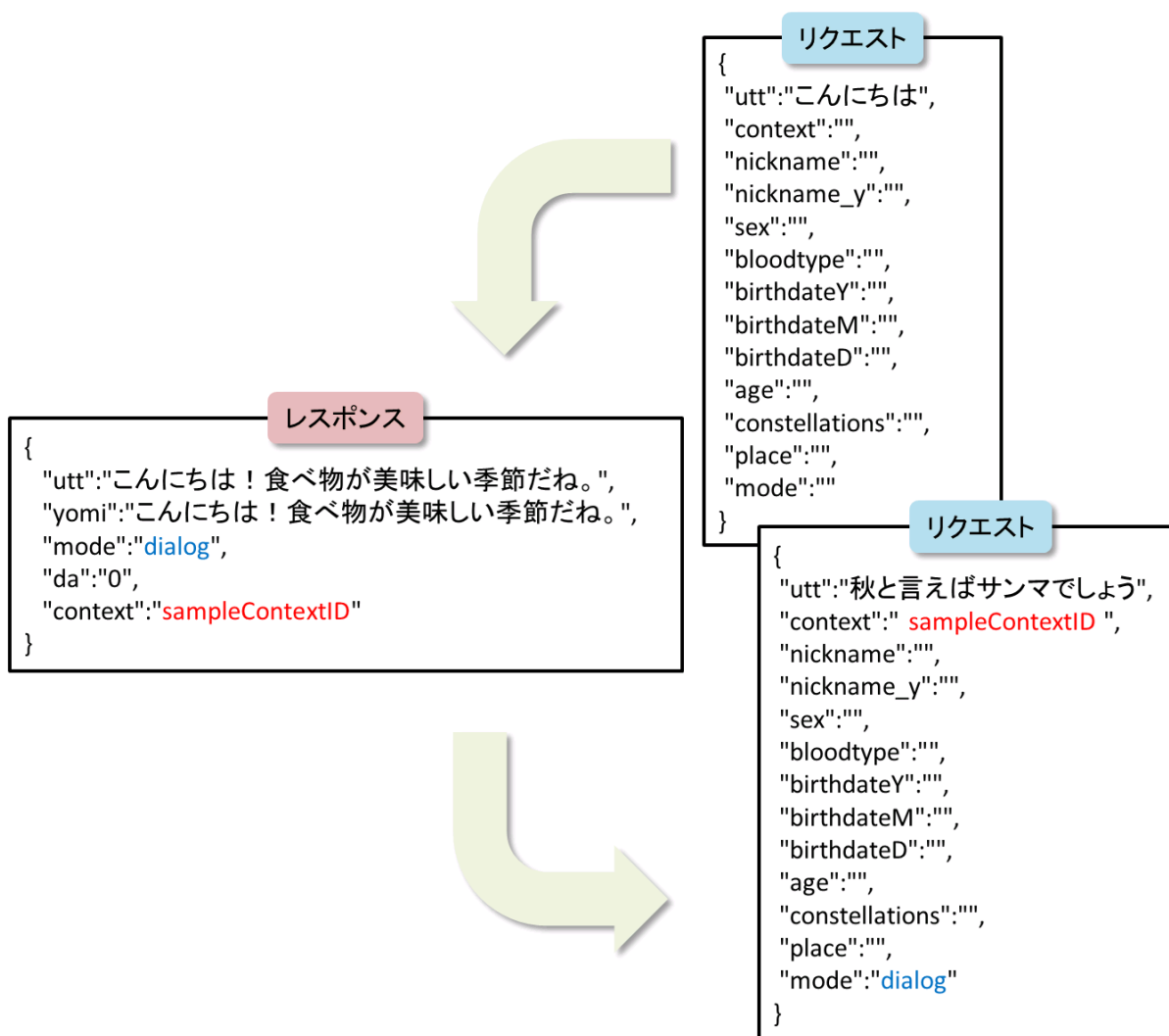
項番	ステータスコード	内容
1	200	正常に応答を返却した場合
2	400	要求パラメータ不良の場合
3	500	サーバ内で予期せぬエラーが発生した場合

3. 雑談対話 API を利用した会話例

本章では雑談対話 API を利用した会話の事例を紹介する。

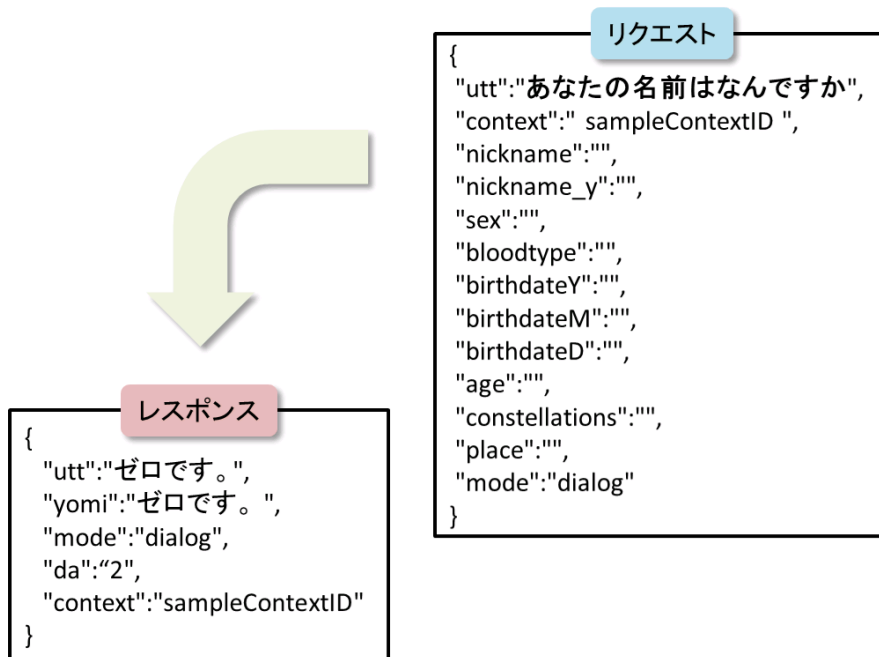
3.1. 会話の継続

レスポンスの「モード(mode)」と「コンテキスト ID(context)」の内容を引き継ぐことにより、対話を継続することができる。



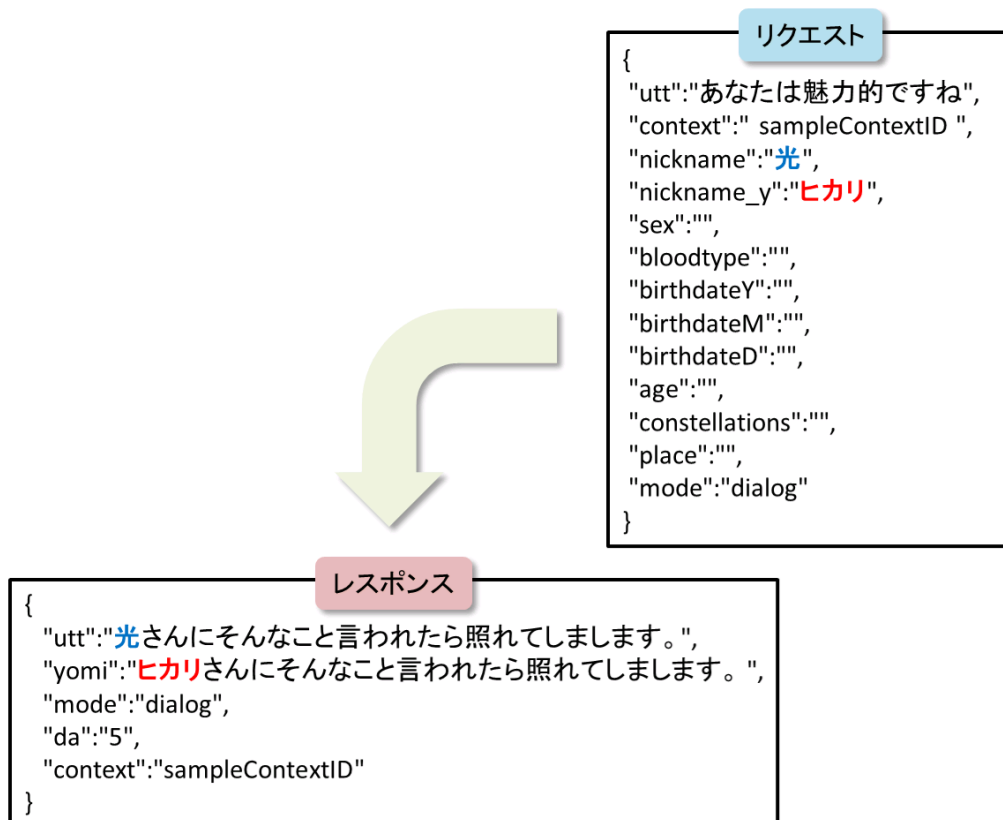
3.2. システムへの質問

システム自身への質問をすることも可能である。



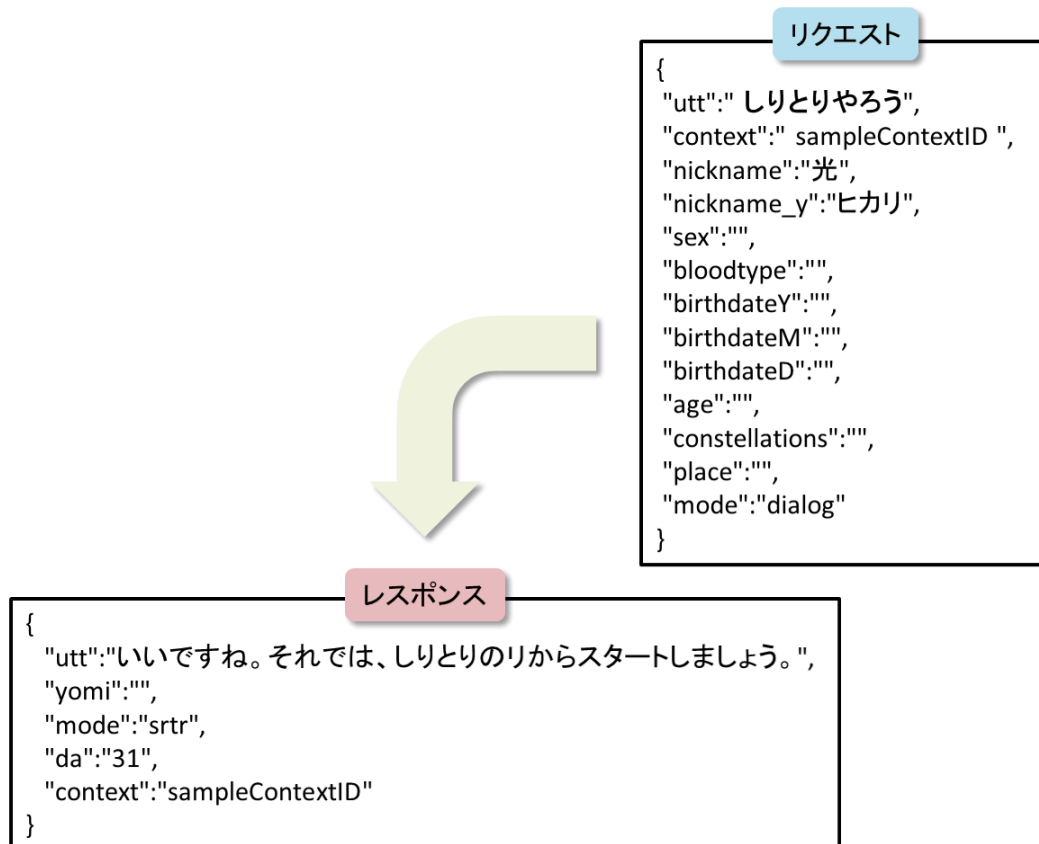
3.3. ユーザのニックネーム等の利用

「ユーザのニックネーム(nickname)」等のユーザの属性を入力することにより返答が変化する場合があります。



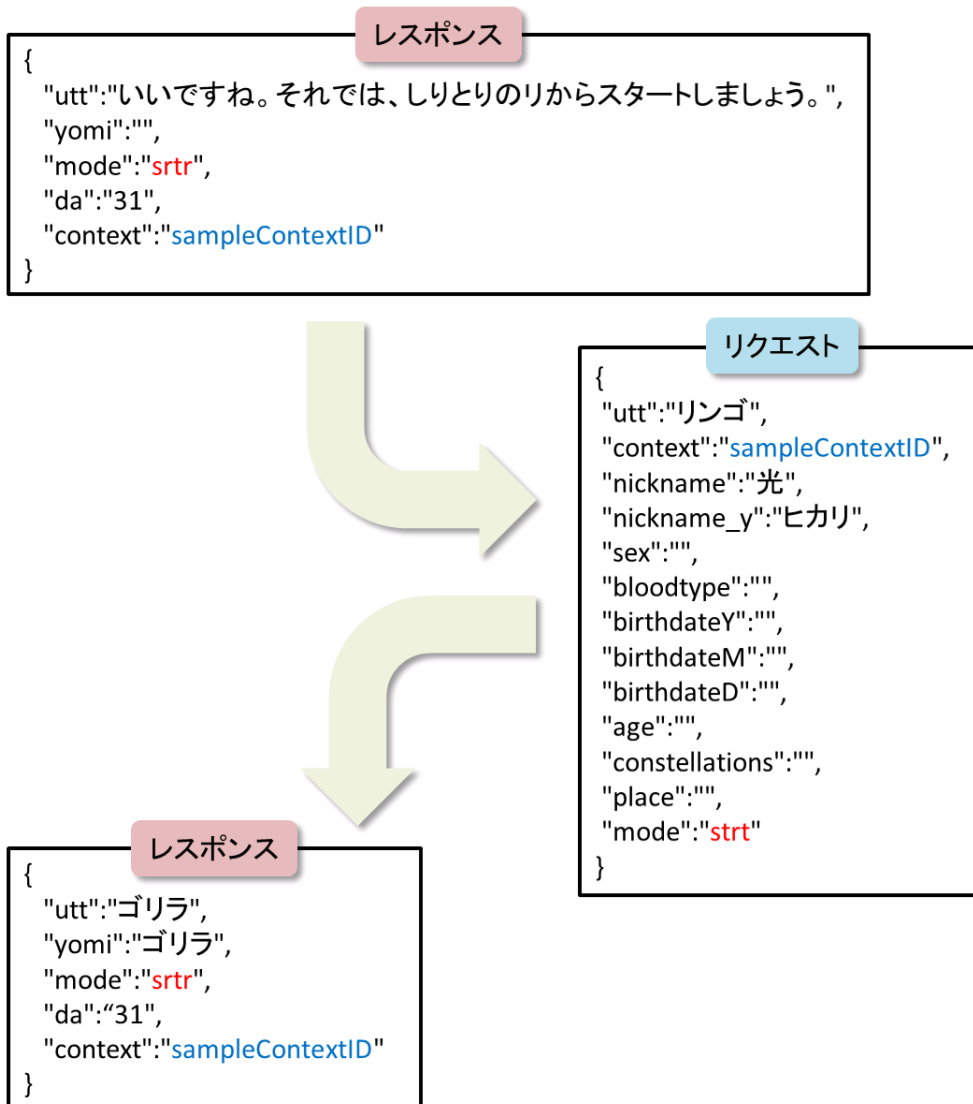
3.4. しりとり開始

「しりとりやろう」等、しりとり開始を示唆する発話を行うことでしりとりを開始することができる。



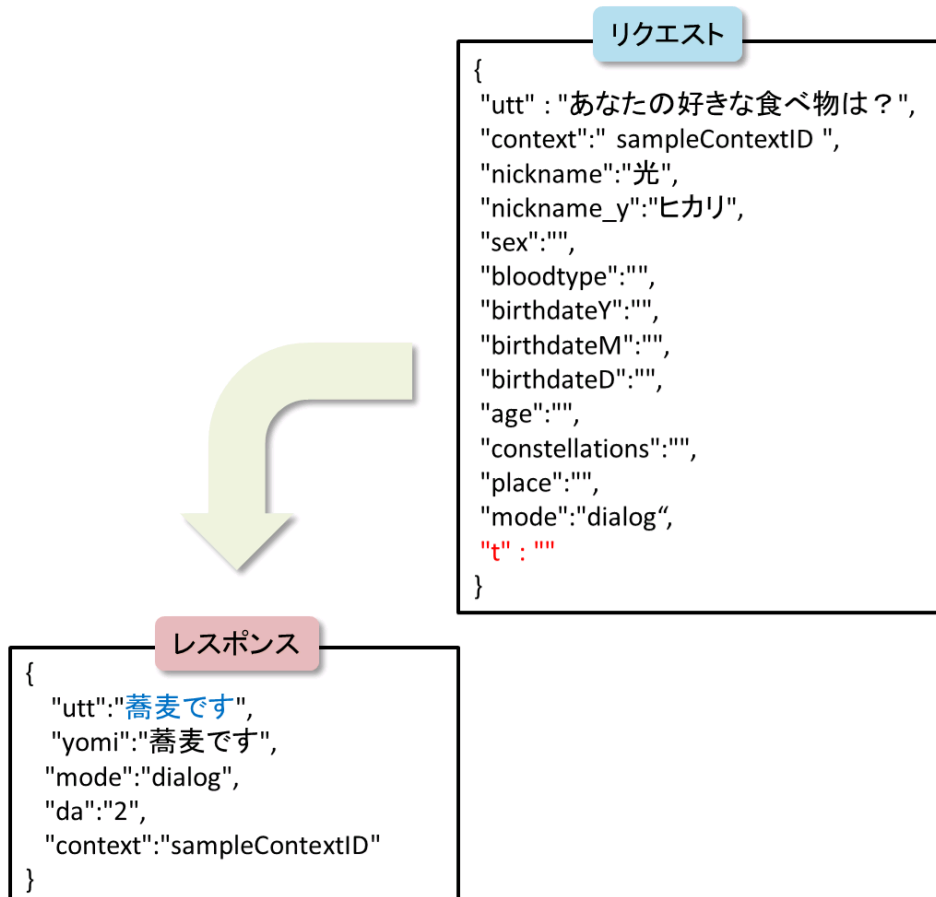
3.5. しりとりの継続

システムからのレスポンスの「モード(mode)」と「コンテキスト ID(context)」を引き継ぐことにより、しりとりを継続することができる。



3.6. キャラクタの設定

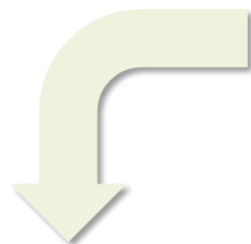
tを設定しない場合はデフォルトのキャラクタとなる。



tを設定するとキャラクターが変わる。

リクエスト

```
{  
  "utt" : "あなたの好きな食べ物は？",  
  "context": " sampleContextID ",  
  "nickname": "光",  
  "nickname_y": "ヒカリ",  
  "sex": "",  
  "bloodtype": "",  
  "birthdateY": "",  
  "birthdateM": "",  
  "birthdateD": "",  
  "age": "",  
  "constellations": "",  
  "place": "",  
  "mode": "dialog",  
  "t" : "30"  
}
```



レスポンス

```
{  
  "utt": "色々食べたいでちゅ",  
  "yomi": "色々食べたいでちゅ",  
  "mode": "dialog",  
  "da": "2",  
  "context": "sampleContextID"  
}
```