

Techmo TTS
Synthesis User Guide

Contents

- 1. Introduction** **4**
 - 1.1. SSML 4
 - 1.2. Interpretation control 5

- 2. Text normalization** **6**
 - 2.1. Numbers 6
 - 2.1.1. Interpret-as cardinal 7
 - 2.1.2. Interpret-as ordinal 7
 - 2.1.3. Interpret-as decimal / number 7
 - 2.1.4. Interpret-as fraction 8
 - 2.2. Dates 8
 - 2.2.1. Interpret-as date 9
 - 2.3. Time 9
 - 2.3.1. Interpret-as time 10
 - 2.4. Currencies 10
 - 2.4.1. Interpret-as currency 11
 - 2.5. Units 11
 - 2.5.1. Interpret-as unit 12
 - 2.6. Address numbers 13
 - 2.7. Email addresses 13
 - 2.7.1. Interpret-as email 13
 - 2.8. URL addresses 13
 - 2.8.1. Interpret-as url 14
 - 2.9. Common abbreviations 14
 - 2.10. Other 15
 - 2.10.1. Interpret-as telephone 15
 - 2.10.2. Interpret-as postal 16
 - 2.10.3. Interpret-as characters / spell-out 16
 - 2.10.4. Interpret-as verbatim 17

- 3. Pronunciation lexicon** **18**
 - 3.1. Lexicon selection 20

- 4. Voice control** **21**



4.1. Voice selection	21
4.2. Voice modulation	22
4.2.1. Emphasis	22
4.2.2. Break	22
4.2.3. Prosody	23
5. Audio samples	24
6. Advanced synthesis examples	25



1. Introduction



This user guide describes methods of building TTS requests for Polish language. Each example number corresponds to an attached audio file.

Example 1.1: A basic request and a transcription of its response

 Dzień dobry. Jestem Michał - twój głosowy asystent.
 *dzień dobry. jestem michał- twój głosowy asystent.*

In order to ensure the most convenient and intuitive use of the TTS service, the system allows sending text in a commonly used form and tries to adjust the expansion to the full vocabulary by itself.

Example 1.2: Sample text normalization

 Nasz adres: ul. Torfowa 1, 30-384 Kraków.
 *nasz adres: ulica torfowa jeden, trzydzieści, trzysta osiemdziesiąt cztery kraków.*

For more complex or ambiguous texts SSML tags may be useful.



1.1. SSML

SSML (Speech Synthesis Markup Language) [1] is a markup language for speech synthesis applications. It defines a set of control tags designed to manipulate synthesized speech. Techmo TTS service supports a part of the SSML standard. By contrast to text normalization, SSML tags give users full control on where to occur.

SSML has a strictly defined syntax:

- every text influenced by SSML has to be contained within opening (**<tag-name>**) and closing tag (**</tag-name>**)
- opening tags may list optional parameters in form **parameter="value"**
- tags that occur between text and do not affect it, may be simplified to just a single one **<tag-name/>**
- every SSML request has to be contained within an overarching tag – **speak**

Example 1.3: A basic SSML request

 `<say-as format="text" detail="text" data-bbox="143 112 684 130"/> Tekst, który obsługuje znaczniki. </say-as>`
 *tekst, który obsługuje znaczniki.*

1.2. Interpretation control

Interpretation control tag indicates how to interpret text. Each **interpret-as** case is considered as a separate one because **format** and **detail** parameters follow it.

Tag name:	say-as
Attributes:	interpret-as, format, detail
Allowed values:	<p>interpret-as</p> <ul style="list-style-type: none"> - label: "cardinal", "ordinal", "decimal" or "number", "fraction", "currency", "unit", "date", "time", "telephone", "postal", "email", "url", "characters" or "spell-out", "verbatim" <p>format (optional)</p> <ul style="list-style-type: none"> - interpret-as dependent <p>detail (optional)</p> <ul style="list-style-type: none"> - interpret-as dependent

Example 1.4: The syntax of say-as tag usage

```
<say-as interpret-as="interpretation" format="format" detail="detail">
```

The **format** attribute specifies format for "date" and "time", while all other interpretations ignore it.

The **detail** attribute can provide an optional grammatical case and gender for numerals or grouping rules for telephone numbers. If the case and gender are not specified, default values are chosen.

The following formats are supported:

- case (for all interpretations with declension),
- case:gender (for numerals only)

The case can be any of the following:

- nom – nominative (mianownik, kto? co?)
- gen – genitive (dopełniacz, kogo? czego?)
- dat – dative (celownik, komu? czemu?)
- acc – accusative (biernik, kogo? co?)
- inst – instrumentalis (narzędnik, z kim?, z czym?)
- loc – locative (miejscownik, o kim?, o czym?)
- voc – vocative (wołacz)

The gender can be any of the following (description in Polish):

- m1 – męski osobowy (mężczyzna, chłopiec)
- m2 – męski zwierzęcy (pies, słoń)
- m3 – męski rzeczowy (chleb, dąb)
- f – żeński (kobieta, książka)
- n1 – nijaki zbiorowy (dziecko, szczenię)
- n2 – nijaki zwykły (słońce, okno)
- p1 – przymnogi męskoosobowy (mężczyźni, chłopcy)
- p2 – przymnogi zwykły (kobiety, dzieci, książki)

IMPORTANT: The format and detail attributes cannot be used without the interpret-as attribute.

2. Text normalization

2.1. Numbers

Numbers are defined for automatic normalization as any rational signed number written with a dot or a comma.

Example 2.1: Numbers



3.54



trzy i pięćdziesiąt cztery setne.

Example 2.2: Decimal fractions



-0,345





minus trzysta czterdzieści pięć tysięcznych.

2.1.1. Interpret-as cardinal



Cardinal say-as tag interprets text as a possibly signed number (integers only).

Tag name:	say-as interpret-as="cardinal"
Attributes:	detail

Example 2.3: Interpret-as cardinal

 `<say-as interpret-as="cardinal">-123</say-as></speak>`
 *minus sto dwadzieścia trzy.*

Example 2.4: Interpret-as cardinal



 `<say-as interpret-as="cardinal" detail="acc:m1"> 23
</say-as></speak>`
 *dwudziestu trzech.*

2.1.2. Interpret-as ordinal



Ordinal say-as tag interprets text as a possibly signed number (integers only).

Tag name:	say-as interpret-as="ordinal"
Attributes:	detail

Example 2.5: Interpret-as ordinal

 `<say-as interpret-as="ordinal"> -123 </say-as></speak>`
 *minus sto dwudziesty trzeci.*

Example 2.6: Interpret-as ordinal



 `<say-as interpret-as="ordinal" detail="gen:f"> 32
</say-as></speak>`
 *trzydziestej drugiej.*

2.1.3. Interpret-as decimal / number



Decimal and number say-as tags interpret text as a signed number with an optional decimal fraction, where the decimal point can be either a dot (.) or a comma (,). The keywords can be used interchangeably.

Tag name:	say-as interpret-as="decimal" / say-as interpret-as="number"
Attributes:	detail

Example 2.7: Interpret-as decimal / number

 `<say-as interpret-as="decimal"> -12.45 </say-as></speak>`
 *minus dwanaście i czterdzieści pięć setnych.*

Example 2.8: Interpret-as decimal / number



 `<say-as interpret-as="decimal" detail="dat:m3"> 3.75
</say-as></speak>`
 *trzem i siedemdziesięciu pięciu setnym.*

2.1.4. Interpret-as fraction



Fraction say-as tag interprets text as an optional cardinal number with a common fraction.

Tag name:	say-as interpret-as="fraction"
Attributes:	detail

Example 2.9: Interpret-as fraction

 `<say-as interpret-as="fraction"> 3/4 </say-as></speak>`
 *trzy czwarte.*

Example 2.10: Interpret-as fraction

 `<say-as interpret-as="fraction" detail="inst:n1"> 3/4
</say-as></speak>`
 *trzema czwartymi.*

2.2. Dates

Dates are defined for automatic normalization as three numbers divided with a dot (.), slash (/) or dash (-) in the format of **DD.MM.YYYY** or **YYYY.MM.DD**. Other date formats can be handled using SSML.

Example 2.11: Date



10.06.2016



dziesiąty czerwca dwa tysiące szesnasty.

Example 2.12: Date



2016-06-10



dziesiąty czerwca dwa tysiące szesnasty.

2.2.1. Interpret-as date

Date say-as tag interprets text as a date with optionally specified year, month, and/or day with dash (-), slash (/) and dot (.) separators.

Tag name:	say-as interpret-as="date"
Attributes:	detail, format
Allowed values:	<p>format</p> <p>- label: "ymd" (year-month-day), "dmy" (day-month-year), "mdy" (month-day-year; american form, avoid using it in polish texts), "ym" (year-month without day), "my" (month-year without day), "md" (month-day without year), "dm" (day-month without year), "y" (year only), "m" (month only), "d" (day only)</p>

Example 2.13: Interpret-as date



`<say-as interpret-as="date" format="mdy"> 5.23.2001`

`</say-as></speak>`



dwudziesty trzeci maja dwa tysiące pierwszy.

Example 2.14: Interpret-as date



`<say-as interpret-as="date" format="md" detail="gen"> 12.09`

`</say-as></speak>`





dziewiątego grudnia.



2.3. Time

Time is defined for automatic normalization as two numbers separated by a colon.

Example 2.15: Time with hours and minutes

 13:05
 *trzynasta pięć.*

Example 2.16: Time with hours and minutes



 09:40
 *dziewiąta czterdzieści.*

2.3.1. Interpret-as time



Time say-as tag interprets text as a time in **hh:mm** format. Time format can be switched to 12h mode using format attribute. Additionally, in the 12h mode, the expressions like "za pięć trzecia" or "wpół do czwartej" are used.

Tag name:	say-as interpret-as="time"
Attributes:	detail, format
Allowed values:	format - label: "24h", "12h"



Example 2.17: Interpret-as time

 `<say-as interpret-as="time"> 16:45 </say-as></speak>`
 *szesnasta czterdzieści pięć.*

Example 2.18: Interpret-as time

 `<say-as interpret-as="time" format="12h"> 16:45 </say-as></speak>`
 *za piętnaście piąta.*



Example 2.19: Interpret-as time

 `<say-as interpret-as="time" detail="loc"> 9:55 </say-as></speak>`
 *dziewiątej pięćdziesiąt pięć.*



2.4. Currencies

Currencies are defined for automatic normalization as numbers with a currency symbol.



Example 2.20: Currency

 3 zł
 *trzy złote.*

Example 2.21: Currency

 12.34 EUR
 *dwanaście euro trzydzieści cztery centy.*

Example 2.22: Currency

 46,12 £
 *czterdzieści sześć funtów dwanaście pensów.*



Supported currency symbols are: zł, PLN, \$, USD, €, EUR, £, GBP, CHF, CNY.

2.4.1. Interpret-as currency



Currency say-as tag interprets text as a decimal number followed by a currency symbol. The tag handles the same symbols as the automatic normalization.

Tag name:	say-as interpret-as="currency"
Attributes:	detail

Example 2.23: Interpret-as currency

 `<say-as interpret-as="currency"> 3.79zł </say-as></speak>`
 *jeden dolar amerykański kosztuje aktualnie trzy złote siedemdziesiąt dziewięć groszy.*

Example 2.24: Interpret-as currency

 `<say-as interpret-as="currency" detail="inst"> 3.57 zł </say-as> ? </speak>`
 *co ja mogę zrobić z trzema złotymi pięćdziesięcioma siedmioma groszami?*

2.5. Units

Units are defined for automatic normalization as numbers followed by a unit symbol.

Example 2.25: Units



3.7 km/h



trzy i siedem dziesiątych kilometra na godzinę.

Supported units are:

nm, μ m (um), mm, cm, m, km	nanometer, micrometer, millimeter, centimeter, meter, kilometer
ns, μ s (us), ms, s	nanosecond, microsecond, milisecond, second
min	minute
h	hour
mg, g, dag (dkg), kg	miligram, gram, decagram, kilogram
t, kt	tonne, kilotonne
l	litre
°, °C	degree, degree Celsius
%, ‰	percent, permille

Furthermore, each unit can be expressed with one of change per time unit suffixes i.e. with "/s", "/min" or "/h".

2.5.1. Interpret-as unit

Unit say-as tag interprets text as a decimal number followed by a unit symbol. The tag handles the same units as the automatic normalization.

Tag name:	say-as interpret-as="unit"
Attributes:	detail

Example 2.26: Interpret-as unit



`<say-as interpret-as="unit"> 317 km/h
</say-as></say-as>`



rekord prędkości to trzysta siedemnaście kilometrów na godzinę.

Example 2.27: Interpret-as unit



`<say-as interpret-as="unit" detail="gen"> 317 km/h </say-as></say-as>`





uzyskał rekordową prędkość trzystu siedemnaście kilometrów na godzinę.

2.6. Address numbers

Address numbers are defined for automatic normalization as the expressions $n[a]/m[b]$, where: n, m – unsigned integers; a, b – optional letters.



Example 2.28: Address number

 12c/3
 *dwanaście ce przez trzy.*

2.7. Email addresses

Email addresses are defined for automatic normalization as two parts connected by "." and separated by "@" sign, optionally ending with ".pl", ".com", ".org" etc.

Example 2.29: Email address



 iksinski@techmo.pl
 *iksinski małpa techmo kropka pe el.*

2.7.1. Interpret-as email

Email say-as tag interprets text as an email address.

Tag name:	say-as interpret-as="email"
Attributes:	-



Example 2.30: Interpret-as email

 <say-as interpret-as="email"> jan.kowalski@techmo.pl
</say-as></speak>
 *jan kropka kowalski małpa techmo kropka pe el.*

2.8. URL addresses

URL addresses are defined for automatic normalization as words separated by dot (.) and ending with ".pl", ".com", ".org" etc.

Example 2.31: URL address



 `https://www.techmo.pl`
 *hatetepees wwwuuu kropka techmo kropka pe el.*

2.8.1. Interpret-as url

URL say-as tag interprets text as a network address.

Tag name:	say-as interpret-as="url"
Attributes:	-

Example 2.32: Interpret-as url

 `<say-as interpret-as="url"> https://www.techmo.pl`
`</say-as></say-as>`
 *hatetepees wwwuuu kropka techmo kropka pe el.*

2.9. Common abbreviations

Autodetected and automatically expanded are the following abbreviations:

adm.	admiraża
al.	aleja / aleje
bt.	btogostawionego / btogostawionej
bp.	biskupa
bulw.	bulwar
gen.	generaża
im.	imienia
kard.	kardynaża
ks.	księdza
marsz.	marszałka
mjr.	majora
os.	osiedle
pl.	plac
płk.	pułkownika
ppłk.	podpułkownika
s.a.	es a
sp. z o.o.	spółka z o o
św.	świętego / świętej
ul.	ulica
wyb.	wybrzeże

Moreover, due to popularity in Poland, the extra exception is defined:

jana pawła ii	jana pawła drugiego
---------------	---------------------

IMPORTANT: If there is more than a single expansion listed, default one is chosen.

2.10. Other


2.10.1. Interpret-as telephone

Telephone say-as tag interprets text as a sequence of digits, optionally separated with spaces (), dashes (-) or hashes (#, in special numbers), with optional international prefix (+) or special number prefix (* or #).


By default a number is spelled as groups of digits as suggested by the user if dashes or spaces are used, or grouped by packs of three (preferable) or three and two automatically. Optionally spelling can be forced with the detail attribute.

Tag name:	say-as interpret-as="telephone"
Attributes:	detail, format
Allowed values:	detail - label: "grouping-digits", "separate-digits"


Example 2.33: Interpret-as telephone

 `<say-as interpret-as="telephone"> +48 123-456-789`


`</say-as></say-as>`

 *plus czterdzieści osiem, sto dwadzieścia trzy, czterysta pięćdziesiąt sześć, siedemset osiemdziesiąt dziewięć.*

Example 2.34: Interpret-as telephone

 `<say-as interpret-as="telephone" detail="separate-digits">`

`+48 123-456-789 </say-as></say-as>`


 *plus cztery osiem jeden dwa trzy cztery pięć sześć siedem osiem dziewięć.*


2.10.2. Interpret-as postal

Postal say-as tag interprets text as sequences of digits separated with dashes (-).

Tag name:	say-as interpret-as="postal"
Attributes:	-

Example 2.35: Interpret-as postal

 `<say-as interpret-as="postal"> 20-034 </say-as></say-as>`


 *dwadzieścia, zero trzydzieści cztery.*


2.10.3. Interpret-as characters / spell-out

Characters and spell-out say-as tags interpret text as an arbitrary text spelled out as separate characters with spaces omitted. The keywords can be used interchangeably.

Tag name:	say-as interpret-as="characters" / say-as interpret-as="spell-out"
Attributes:	detail
Allowed values:	detail - label: "output-quotations", "output-quotations-commas"


Example 2.36: Interpret-as characters / spell-out


 `<say-as interpret-as="characters"> AWS123`
`</say-as></say-as>`

 *twój numer to: a wu es jeden dwa trzy.*


Parameter **detail** can be used to slow the pronunciation down.


Example 2.37: Interpret-as characters / spell-out

 `<say-as interpret-as="characters"`
`detail="output-quotations"> AWS123 </say-as></say-as>`

 *twój numer to: "a" "wu" "es" jeden dwa trzy.*

Example 2.38: Interpret-as characters / spell-out

 `<say-as interpret-as="characters"`
`detail="output-quotations-commas"> AWS123 </say-as></say-as>`


 *twój numer to: "a", "wu", "es", "jeden, dwa, trzy.*


2.10.4. Interpret-as verbatim

Verbatim say-as tag interprets text as a word from letters only, passed through text normalizer as-is. It is useful to avoid any unwanted transformations, autodetection, etc.


Tag name:	say-as interpret-as="verbatim"
Attributes:	-


Example 2.39: Unusual phrase without use of interpret-as verbatim

 `Dokupiłem do pasieki nowy ul. W sumie mam już 8.`

 *dokupiłem do pasieki nowy ulica w sumie mam już osiem.*

Example 2.40: Unusual phrase with use of interpret-as verbatim

 `<say-as interpret-as="verbatim">`
`ul </say-as>. W sumie mam już 8. </say-as>`

 *dokupiłem do pasieki nowy ul. w sumie mam już osiem.*

3. Pronunciation lexicon

The pronunciation of foreign words, proper names or acronyms is defined by the pronunciation lexicons. The default *lexicon.xml* file is supplied together with the rest of the service's resources.

Pronunciation lexicons are defined according to PLS [2]. However, the TTS service interprets only a subset of what is allowed by PLS:

- **<lexeme>** tag has to contain exactly one **<alias>** or **<phoneme>** child
- **alphabet**, **prefer**, and **role** attributes are silently ignored

Example 3.1: A lexicon entry definition with alias

```
<lexeme>
  <grapheme>business</grapheme>
  <alias>biznes</alias>
</lexeme>
```

Additionally, **<grapheme>** tags can have a custom, techmo-specific **match** attribute, which adjusts the way in which graphemes are matched against the text. The **match** attribute is optional, and has the format: *match-type*[, *match-case*]. The *match-type* can be one of:

- **full**
- **partial**
- **full-partial**
- **partial-full**
- **regexp**

and *match-case* can be:

- **icase** (case insensitive matching)
- **case** (case sensitive matching)

The **full** match means that a matched phrase cannot be the part of a word, i.e. it has to begin and end at the word boundaries, which is the matching required by PLS. The **partial** match means that phrases which are parts of words are matched as well. The **full-partial** and **partial-full** match modes means that phrase can be matched to word's beginning or word's end, respectively.

Alternatively, the **regexp** match means that matching is performed using regular expressions, according to ECMA [3] script syntax. The default matching mode is **full, icase** (case insensitive full phrase match).

Example 3.2: A lexicon entry definition with a detailed match-type

```
<lexeme>
  <grapheme match="full, case">AC</grapheme>
  <alias>a-ce</alias>
</lexeme>
```

Some sequences of letters are read in Polish in a specific, not always expected ways. These sequences can be split using pipe (|) symbol.

Example 3.3: A lexicon entry definition without pipe

```
<lexeme>
  <grapheme match='full, case'>Jersey</grapheme>
  <alias>dżerzi</alias>
</lexeme>
```

Example 3.4: A lexicon entry definition with pipe

```
<lexeme>
  <grapheme match='full, case'>Jersey</grapheme>
  <alias>dżer|z|i</alias>
</lexeme>
```

Similar rules apply to the location of stress in Polish. There are two main ways to modify it:

- splitting words using punctuation (like whitespace () or dash (-))
- adding stress symbol (') directly before a vowel that should be stressed

Example 3.5: A lexicon entry definition without splitting

```
<lexeme>
  <grapheme>dvd</grapheme>
  <alias>diwidi</alias>
</lexeme>
```

Example 3.6: A lexicon entry definition with splitting

```
<lexeme>
  <grapheme>dvd</grapheme>
  <alias>di-wi-di</alias>
</lexeme>
```

Example 3.7: A lexicon entry definition without stress

```
<lexeme>
  <grapheme>liechtenstein</grapheme>
  <alias>lichtensztajn</alias>
</lexeme>
```

Example 3.8: A lexicon entry definition with stress

```
<lexeme>
  <grapheme>liechtenstein</grapheme>
  <alias>l'ichtensztajn</alias>
</lexeme>
```

There are some rare foreign words where above methods prove insufficient (like "ę" at the end of a word). For such cases it's best to use **<phoneme>**. Note that **partial**, **full-partial** or **partial-full** attributes do not work with **<phoneme>** though.

Example 3.9: A lexicon entry definition with insufficient alias

```
<lexeme>
  <grapheme>leroy merlin</grapheme>
  <alias>lerɨ'a merl'ę</alias>
</lexeme>
```

Example 3.10: A lexicon entry definition with phoneme

```
<lexeme>
  <grapheme>leroy merlin</grapheme>
  <phoneme>lerw'ã merl'ẽw̃</phoneme>
</lexeme>
```

Any changes to the pronunciation lexicon should be made carefully, especially those that use the **partial** or **regexp** match-type.

Lexicons are loaded at the start of the service. To handle the newly added changes, the service must be restarted.

3.1. Lexicon selection



Lexicon tag maps a lexicon URI to a custom identifier.

Tag name:	lexicon
Attributes:	xml:id, uri
Allowed values:	<p>xml:id</p> <ul style="list-style-type: none"> - custom identifier of a lexicon <p>uri</p> <ul style="list-style-type: none"> - URI of a lexicon

Lookup tag invokes a mapped lexicon. This makes switching between multiple lexicons possible.

Tag name:	lookup
Attributes:	ref
Allowed values:	<p>ref</p> <ul style="list-style-type: none"> - an identifier declared with the preceding lexicon tag

Example 3.11: Declaring and selecting a lexicon

 `<speak><lexicon xml:id="l2" uri="lex_en"/> Amerykanie mówią na to
<lookup ref="l2"> pies </lookup></speak>`
 *amerykanie mówią na to pajs.*

NOTE: The example above is a demonstration of the functionality. In order to use it, it would be necessary to add a second pronunciation lexicon named "lex_en".

IMPORTANT: Requesting to use a lexicon with mismatched language results in selecting no lexicon.

4. Voice control

4.1. Voice selection

Voice selection tag allows to switch voice on the fly, if a TTS service has more than one voice loaded.

Tag name:	voice
Attributes:	name, variant
Allowed values:	<p>name</p> <ul style="list-style-type: none"> - a name of a voice <p>variant</p> <ul style="list-style-type: none"> - a number representing a variant of a voice (1 is default)

Example 4.1: Changing voice



```
<speaK>
  <voice name="Michal" variant="3"> Poczekaj. </voice>
  <voice name="Masza"> Dobrze. Poczekam. </voice>
  <voice name="Michal" variant="1"> Czy chciałbyś połączyć się z
konsultantem? </voice>
</speaK>
```

poczekaj. dobrze. poczekam. czy chciałbyś połączyć się z konsultantem?

4.2. Voice modulation

4.2.1. Emphasis

Emphasis tag adds emphasis to contained text.

Tag name:	emphasis
Attributes:	level
Allowed values:	level - label: "strong", "moderate", "none", "reduced"

Example 4.2: Using emphasis



```
<speaK> Uwaga! <emphasis level="moderate"> Powtarzam zdanie głośno
i wyraźnie. </emphasis></speaK>
```



uwaga! powtarzam zdanie głośno i wyraźnie.

4.2.2. Break

Adds a pause in a selected place in an utterance.

Tag name:	break
Attributes:	time, strength
Allowed values:	time - duration in seconds or milliseconds (e.g. "2400ms") strength - label: "none", "x-weak", "weak", "medium", "strong", "x-strong"

Example 4.3: Using break

 `<speak> Proszę chwilę poczekać. <break time="1s"/> Gotowe.`
`</speak>`
 *proszę chwilę poczekać. <pauza> gotowe.*



IMPORTANT: Text that occurs inside this tag gets omitted.

4.2.3. Prosody



Prosody tag controls pitch, range, speaking rate and volume of speech.

Tag name:	prosody
Attributes:	pitch, range, rate, volume
Allowed values:	<p>pitch</p> <ul style="list-style-type: none"> - frequency in Hz (e.g. "250Hz") - relative change in percents (e.g. "-25%") - relative change in semitones (e.g. "+5st") - labels: "x-low", "low", "medium", "high", "x-high", "default" <p>range</p> <ul style="list-style-type: none"> - frequency in Hz (e.g. "250Hz") - relative change in percents (e.g. "-25%") - relative change in semitones (e.g. "+5st") - labels: "x-low", "low", "medium", "high", "x-high", "default" <p>rate</p> <ul style="list-style-type: none"> - non-negative percentage (e.g. "120%", "75%") - labels: "x-slow", "slow", "medium", "fast", "x-fast", "default" <p>volume</p> <ul style="list-style-type: none"> - relative change in decibels (e.g. "-10dB") - labels: "silent", "x-soft", "soft", "medium", "loud", "x-loud", "default"



Example 4.4: Changing voice volume

 `<speak><prosody volume="-6dB"> Przykładowe zdanie, ale ciszej.`
`</prosody></speak>`
 *przykładowe zdanie, ale ciszej.*



Example 4.5: Changing pitch of the voice

 `<speack><prosody pitch="-5%"> Głos obniżony o 5% </prosody></speack>`
 *głos obniżony o pięć procent.*

Example 4.6: Changing speaking rate

 `<speack><prosody rate="fast"> Tym razem mówię szybciej niż zwykle.
</prosody></speack>`
 *tym razem mówię szybciej niż zwykle.*



Example 4.7: Changing pitch range (variability) of the voice

 `<speack><prosody range="x-low"> Głos o mocno wypłaszczonej melodii.
</prosody></speack>`
 *głos o mocno wypłaszczonej melodii.*

5. Audio samples

To make utterances more realistic or express emotional responses, audio samples may be worked into the synthesized text. To use them a special tokens are defined.

Example 5.1: A request with a sound icon

 `:haha2: Bardzo śmieszne, ale czy moglibyśmy wrócić do tematu
rozmowy?`
 `<śmiech> bardzo śmieszne, ale czy moglibyśmy wrócić do tematu
rozmowy?`

Using custom audio files is possible with SSML audio tag. The selected file must be available on the filesystem used by the service. Therefore, for service instances running on Docker, they should be placed inside the 'resources' directory, with the path specified as "/tts-dnn/resources/..."

Example 5.2: A request with an audio sample selected from a disk



```
<speak>
```

```
  <audio src="file://  
    /tts-dnn/resources/predefined_recordings/entuzjastyczne_powitanie.wav  
  "/>
```

```
  Jak mogę ci pomóc?
```

```
</speak>
```

witaj! jak mogę ci pomóc?

If an audio sample is not found, text enclosed by an audio tag gets synthesized.

Example 5.3: A request with an audio sample that is not found



```
<speak><audio src="#invalid"> aha </audio>. Już rozumiem. </speak>
```

aha. już rozumiem.

6. Advanced synthesis examples

Examples shown in this section combine all previously described methods to build requests that return accurate refined responses.

Example 6.1: Advanced input text before SSML



```
  Jeszcze nie widzę twojego potwierdzenia. Zaloguj się do aplikacji  
  mobilnej. Wejdź w menu w lewym górnym rogu i Wybierz mobilną autoryzację.  
  Na samej górze listy znajdziesz operację numer 3 - Zatwierdź ją.
```

*jeszcze nie widzę twojego potwierdzenia. zaloguj się do aplikacji
 mobilnej. wejdź w menu w lewym górnym rogu i wybierz mobilną autoryzację.
 na samej górze listy znajdziesz operację numer trzy- zatwierdź ją.*

Example 6.2: Advanced input text after SSML



```
<speak>
  <voice variant="1">
    <prosody rate="110%" range="high" pitch="-3%">
      Jeszcze nie widzę twojego potwierdzenia.
    </prosody>
  </voice>
  <voice variant="3">
    <prosody range="x-low">
      Zaloguj się do aplikacji mobilnej.
    </prosody>
  </voice>
  <break time="1s"/>
  <voice variant="1">
    <prosody pitch="-3%">
      Wejdź w menu w lewym górnym rogu i Wybierz mobilną autoryzację.
    </prosody>
    Na samej górze listy znajdziesz operację numer 3 - Zatwierdź ją.
  </voice>
</speak>
```

jeszcze nie widzę twojego potwierdzenia. zaloguj się do aplikacji mobilnej. wejdź w menu w lewym górnym rogu i wybierz mobilną autoryzację. na samej górze listy znajdziesz operację numer trzy- zatwierdź ją.

Example 6.3: Advanced example of interpret-as



```
<speak>
```

```
  W przeciągu ostatnich
```

```
  <say-as interpret-as="unit" detail="gen">
```

```
    24h
```

```
  </say-as>
```

```
  cena bitcoina wzrosła o 5.73% do
```

```
  <say-as interpret-as="currency" detail="gen">
```

```
    34141.03 USD
```

```
  </say-as>
```

```
</speak>
```

w przeciągu ostatnich dwudziestu czterech godzin cena bitcoina wzrosła o pięć i siedemdziesiąt trzy setne procent do trzydziestu czterech tysięcy stu czterdziestu jeden dolarów trzech centów.

References

- [1] Baggia P., Bagshaw P., Bodell M. et al. (2010), *Speech Synthesis Markup Language (SSML) Version 1.1*, W3C.
- [2] Bagshaw P., Burnett D. C., Carter J. et al. (2008), *Pronunciation Lexicon Specification (PLS) Version 1.0*, W3C.
- [3] Ecma International (2011), *RegExp (Regular Expression) Objects*.