title: JSON structure weight: 10

In your custom skills, communication between your client device, the core routing component of Watson Assistant Solutions, and your custom skills is implemented using evaluate and converse methods.

In figure 1, the flow of a conversation between your client device and your assistant is displayed. For more information about the flow of the conversation, see the *How routing works* topic.

Figure 1 - conversation flow

In sections 1-6, an example of the JSON structure at each step in the flow is presented.

The following high-level scenario is captured in the JSON examples:

Scenario

John has registered members of his family to use an assistant named Watson. John is currently at his home in London and is planning to travel to the city center if the temperature does not get too hot. He says "hello Watson" to wake up his device in the kitchen, and asks "What are the temperatures like today in London city center".

Internally, some context information is set as the request flows through Watson Assistant Solutions to a weather skill. John's user ID is sent in the request to the assistant. In the utterance context, `$locationName` is set to `at-home`.

The weather skill determines from the utterance that John is interested in temperatures specifically and takes note of this interest for future conversation turns with the weather skill. His interest in the city center might be of value to other skills. In an evaluation response, the weather skill adds `$weather_interest` to the skill context and sets it to `temperature`. In the session context, the skill sets `$zone` to city-center.

The weather skill responds with "In London city center, low temperature today will be 83 degrees fahrenheit and high temperature today will be 109 degrees fahrenheit". Watson plays the audio to John through his speaker. Because John is at home, the skill also returns a card with a URL to a temperature map image for London city. His smart device at home is multi-modal and allows him to view images as well as read text and hear audio. The device displays the temperature map.

---

1. Converse request from a client device to the routing core

The JSON structure of the converse request from the client device to the routing core is as follows:

```
{
  "text": "What are the temperatures like today in London city center",
  "language": "en-US",
  "userID": "john-001",
  "clientID": "home-speaker-001",
  "deviceType": "smart-speaker",
```

```
    "additionalInformation": {
      "context": {
        "locationName": "at-home",
        "locationLatitude": 36.169941,
        "LocationLongitude": -115.139829
      }
    }
  }
```

Table 1 - Converse request parameters

Parameter	Description	Type	Required
`text`	The user utterance.	string	yes
`language`	The language that the user utterance is in.	string	yes
`userID`	The ID of the end-user who made the utterance. For example, three family members share a smart speaker. The ID is of the user who is conversing with the smart speaker.	string	yes
`clientID`	The ID of the client device. For example, the ID of the smart speaker. The parameter is for future use.	string	no
`deviceType`	A value that represents the type of client or device from where the utterance was sent. The parameter is for future use.	string	yes
`Additional information`	Extra information about the context of the conversation. Only information that is added to the context object is sent to the skill.	object	yes

Table 2 - Converse request parameters - Additional information

Parameter	Description	Type	Required
`context`	The utterance context. For example, the utterance context might capture whether a user is at home or in his car. A skill might use a different response depending on the utterance context. When a user is at home and asks about expected temperatures, the skill might return a temperature map with the response. When the user is in the car, the temperature map is not returned. An empty context object is allowed. You can add any additional parameters that might be useful to your skill under `context`. In the example, latitude and longitude parameters are added to represent the location of the user.	object	yes

Important In the current implementation, the routing core does not send `deviceType` and `clientID` to the skill. However, you can add this information to the utterance context under additional information.

2. Evaluate request from the routing core to the skills

The JSON structure of the evaluate request from the routing core to the skills is as follows:

```
{
  "id": "001",
  "version": "1.0",
  "language": "en-US",
  "text": "what are the temperatures like today in london city center",
  "context": {
    "user": {
      "id": "john-001"
    },
    "session": {
      "id": "session-001",
      "new": "true",
      "attributes": {},
      "version": "1.0"
    },
    "application": {
      "id": "app-001",
      "attributes": {
        "locationName": "at-home",
      "locationLatitude": 36.169941,
      "LocationLongitude": -115.139829
      }
    }
  }
}
```

Table 3 - Evaluate request parameters

Parameter	Description
`id`	The request ID that is assigned by the routing core.
`version`	The request version that is assigned by the routing core. The version is always `1.0`.
`language`	The language that the user utterance is in.
`text`	The user utterance after the routing core has normalized the text. In all languages, uppercase text is converted to lowercase. In US English (en-US), further normalization techniques are applied, for example, numerals are converted to words, punctuation is removed.

Parameter	Description
`context`	Information about the context of the conversation with the user.

Table 4 - Evaluate request parameters - context

Parameter	Description
`user`	The user ID.
`session`	Information about the session.
`application`	The application ID and any utterance context information. The parameter is for future use.

Table 5 - Evaluate request parameters - user context

Parameter	Description
`id`	The unique ID of the user.

Table 6 - Evaluate request parameters - session context

Parameter	Description	Type	Required
`id`	The ID of the session that is assigned by the routing core.		
`new`	Specifies whether a conversation with the user is already in progress.		
`attributes`	Includes session context and skill context information. **Note**: Because the sample evaluate request is the first request in the conversation, no session or skill context attributes are included in the example.		
`version`	The version of the session that is assigned by the routing core. The version is always `1.0`.		

Table 7 - Converse request parameters - application context

Parameter	Description
`id`	The unique ID of the application. The parameter is for future use.
`attributes`	Includes utterance context information, if present.

3. Evaluate response from the skill to the routing core

The JSON structure of the evaluate response from a skill to the routing core is as follows:

```
{
  "responseCode": 200,
  "requestResult": "In London city center, low temperature today will be 83
degrees fahrenheit and high temperature today will be 109 degrees fahrenheit.",
  "handleUtterance": true,
  "context": {
    "user": {
      "id": "john-001"
    },
    "session": {
      "id": "session-001",
      "new": true,
      "skill": {
        "attributes": {
          "weather-interest": "temperature"
          }
      },
      "attributes": {
        "zone": "city-center"
        },
      "version": "1.0"
    },
    "application": {
      "id": "app-001",
      "attributes": {
        "locationName": "at-home",
        "locationLatitude": 36.169941,
        "LocationLongitude": -115.139829
      }
    }
  },
 "intentities": [
     {
      "name": "wcs",
      "entities": [
        {
          "entity": "weatherType",
          "value": "temperature",
          "confidence": 1
        },
        {
          "entity": "datePhrase",
          "value": "today",
          "confidence": 1
        },
        {
          "entity": "sys-location",
          "value": "london",
          "confidence": 0.962316
        }
      ],
      "intents": [
        {
```

```
            "intent": "get-temperature",
            "confidence": 0.85514235496521
          },
          {
            "intent": "get-forecast",
            "confidence": 0.75514235496521
                  }
              ]
          },
       {
        "name": "regexp",
        "entities": [
          {
            "entity": "sys-location",
            "value": "london",
            "confidence": 0.941245
          }
        ],
        "intents": [
          {
            "intent": "get-rainfall",
            "confidence": 0.63214235496521
          },
          {
            "intent": "get-warnings",
            "confidence": 0.43914235496521
                  }
              ]
          }
      ]
  }
```

Table 8 - Evaluate response parameters

Parameter	Description	Type	Required
`responseCode`	The status of the response. For example, 200 (OK) or 400 (bad request)	string	yes
`requestResult`	The response to the utterance if returned by the nlu engine. The regexp nlu engine does not return a response in an evaluation response.	string	yes
`handleUtterance`	Set to false if the skill is capable of handling the utterance but decides not to handle it. For example, a skill is designed to only display a map when the user is at home. When the skill detects that the user in a car, it sets `handleUtterance` to false.	boolean	yes
`context`	Information about the context of the conversation with the user.	object	yes

Parameter	Description	Type	Required
`intentities`	The intents and entities returned by the skill for each nlu engine.	array	yes

Table 9 - Evaluate response parameters - context

Parameter	Description	Type	Required
`user`	The user ID.	string	yes
`session`	Information about the session, including session context information.	object	yes
`application`	The application ID and utterance context information.	object	yes

Table 10 - Evaluate response parameters - user context

Parameter	Description	Type	Required
`id`	The unique ID of the user.	string	yes

Table 11 - Evaluate response parameters - session context

Parameter	Description	Type	Required
`id`	The ID of the session that is assigned by the routing core.	string	yes
`new`	Specifies whether a conversation with the user is already in progress.	string	yes
`skill`	Includes attributes representing the skill context.	object	yes
`attributes`	Includes any session context information. An empty attributes object is allowed.	object	yes
`version`	The version of the session. The version is always `1.0`.	string	yes

Table 12 - Evaluate response parameters - skill context

Parameter	Description	Type	Required
`attributes`	Includes any skill context information. An empty attributes object is allowed.	object	yes

Table 13 - Evaluate response parameters - application context

Parameter	Description	Type	Required
`id`	The unique ID of the application. The parameter is for future use.	string	yes

Parameter	Description	Type	Required
`attributes`	Includes any utterance context information. An empty attributes object is allowed.	object	yes

Table 14 - Converse request parameters - application attributes

Parameter	Description	Type	Required
`attributes`	Includes any utterance context information. An empty attributes object is allowed.	object	yes

Table 15 - Evaluate response parameters - Intentites

Parameter	Description	Type	Required
`name`	The nlu type that processed the evaluation request	string	yes
`entities`	The entities extracted by the nlu engine from the utterance.	array	no
`intents`	The intents returned by the nlu engine.	array	no

Table 16 - Evaluate response parameters - Entities

Parameter	Description	Type	Required
`entity`	The name of an entity extracted from the utterance by the nlu engine.	string	yes
`value`	The value of the extracted entity.	string	yes
`confidence`	A confidence value that is associated the entity value. **Note**: A confidence score is always returned by a skill for each entity. However, the confidence score for an entity is only considered if no intent is returned, that is, if entity-based routing is being used.	string	yes

Table 17 - Evaluate response parameters - Intents

Parameter	Description	Type	Required
`intent`	The name of an intent that was returned by an nlu engine.	string	yes
`confidence`	A confidence value that is associated the intent.	string	yes

4. Converse request from the routing core to the skill

The JSON structure of the converse request from the routing core to a skill is as follows:

```
{
  "id": "001",
  "version": "1.0",
  "language": "en-US",
  "text": "What are the temperatures like today in London city center",
  "retext": "what are the temperatures like today in london city center",
  "attributes": {
    "intent": "get-temperature"
      },
  "context": {
    "user": {
      "id": "john-001"
      },
    "session": {
      "new": true,
      "skill": {
      "attributes": {
        "weather-interest": "temperature"
        }
      },
      "attributes": {
        "zone": "city-center"
      },
      "version": "1.0"
    },
    "application": {
      "id": "app-001",
      "attributes": {
        "locationName": "at-home",
        "locationLatitude": 36.169941,
        "LocationLongitude": -115.139829
      }
    }
    },
  "skill": {
    "name": "weather",
    "intents": [
      {
        "intent": "get-temperature",
        "confidence": 0.85514235496521
      }
    ],
    "entities": [
      {
      "entity": "weatherType",
      "value": "temperature",
      "confidence": 1
    },
    {
      "entity": "datePhrase",
      "value": "today",
      "confidence": 1
    },
```

```
      {
        "entity": "sys-location",
        "value": "london",
        "confidence": 0.962316
      }
      ],
      "confidence": 0.85514235496521
    },
    "evaluationResponse": {
      "response": "In London city center, low temperature today will be 83 degrees
  fahrenheit and high temperature today will be 109 degrees fahrenheit.",
      "handleRequest": true,
      "context": {
        "user": {
          "id": "john-001"
        },
        "session": {
          "new": true,
          "skill": {
            "attributes": {
              "weather-interest": "temperature"
              }
            },
          "attributes": {
            "zone": "city-center"
          },
          "version": "1.0"
        },
        "application": {
          "id": "app-001",
          "attributes": {
            "locationName": "at-home",
            "locationLatitude": 36.169941,
            "LocationLongitude": -115.139829
          }
        }
      }
    }
  }
```


Table 18 - Converse request parameters

Parameter	Description
`id`	The request ID that is assigned by the routing core.
`version`	The request version that is assigned by the routing core. The version is always `1.0`.
`language`	The language that the user utterance is in.

Parameter	Description
`text`	The user utterance.
`retext`	The user utterance after the text is normalized. In all languages, uppercase text is converted to lowercase. In US English (en-US), further normalization techniques are applied, for example, numerals are converted to words, punctuation is removed.
`attributes`	Includes the intent with the highest confidence score. If no intent was returned, the entity with the highest confidence score is specified. Note: The intent or entity must have a confidence score that is above the confidence score threshold that is set in th manifest file of the skill. The default threshold in the NodeJS boilerplate is `0.85`.
`context`	Information about the context of the conversation with the user.
`skill`	
`evaluationResponse`	Information about response to the utterance that the skill returned with the highest confidence level.

Table 19 - Converse request parameters - attributes

Parameter	Description
`intent`	The name of the intent with the highest confidence score.

Table 20 - Converse request parameters - context

Parameter	Description
`user`	The user ID.
`session`	Information about the session.
`application`	The application ID and any utterance context information.

Table 21 - Converse request parameters - user context

Parameter	Description
`id`	The unique ID of the user.

Table 22 - Converse request parameters - session context

Parameter	Description
`new`	Specifies whether a conversation with the user is already in progress.
`skill`	Includes attributes representing the skill context

Parameter	Description
`attributes`	Includes any session context information.
`version`	The version of the session. Version information is set by the routing core. The version is always `1.0`.

Table 23 - Evaluate response parameters - skill context

Parameter	Description
`attributes`	Includes any skill context information.

Table 24 - Converse request parameters - application context

Parameter	Description
`id`	The unique ID of the application. The parameter is for future use.
`attributes`	Includes attributes representing utterance context information.

Table 25 - Converse request parameters - application attributes

Parameter	Description
`attributes`	Includes any utterance context information.

Table 26 - Converse request parameters - application attributes

Parameter	Description
`attributes`	Includes any utterance context information.

Table 27 - Converse request parameters - skill

Parameter	Description
`name`	The name of the skill.
`entities`	The entities that were extracted from the utterance.
`intents`	The intent of the skill that returned the highest confidence score.
`confidence`	The confidence score of the intent or entity that returned the highest confidence score.

Table 28 - Converse request parameters - entities

Parameter	Description
`entity`	The name of an entity extracted from the utterance.

Parameter	Description
value	The value of the extracted entity.
confidence	A confidence value that is associated the entity value.

Table 29 - Converse request parameters - intents

Parameter	Description
intent	The name of the intent with the highest confidence score.
confidence	The confidence score of the intent.

Table 30 - Converse request parameters - evaluationResponse

Parameter	Description
response	The response to the utterance.
handleRequest	Reflects the value of the handleUtterance parameter returned by the skill in the evaluation response.
context	Context information that was returned by the skill in the evaluation response. See Table 9 for a description of the context object.

5. Converse response from the skill to the routing core

The JSON structure of the converse response from a skill to the routing core is as follows:

```
{
  "reject": false,
  "error": 200,
  "deleteSkillSession": false,
  "captureInput": false,
  "speech": {
    "text": "In London city center, low temperature today will be 83 degrees
fahrenheit and high temperature today will be 109 degrees fahrenheit."
  },
  "card": {
    "type": "show-temp-map",
    "content": {
        "id": "134325"
        "image_url": "https://www.bbc.co.uk/weather/2643743"
     }
  },
  "skill": {
    "name": "weather",
    "intents": [
      {
        "intent": "get-temperature",
```

```
          "confidence": 0.85514235496521
        }
      ],
      "entities": [
        {
          "entity": "weatherType",
          "value": "temperature",
          "confidence": 1
          },
          {
            "entity": "datePhrase",
            "value": "today",
            "confidence": 1
          },
          {
            "entity": "sys-location",
            "value": "london",
            "confidence": 0.962316
          }
      ],
      "confidence": 0.85514235496521
    },
    "additionalInformation": {
      "context": {
        "application": {
          "id": "app-001",
          "attributes": {
            "locationName": "at-home",
            "locationLatitude": 36.169941,
            "LocationLongitude": -115.139829
            }
        },
        "session": {
          "id": "session-001",
          "new": true,
          "skill": {
              "attributes": {
                "weather-interest": "temperature",
                "inConversation": false
                }
              },
          "attributes": {
            "zone": "city-center"
        },
          "version": "1.0"
        }
      }
    }
  }
```


Table 31 - Converse response parameters

Parameter	Description	Type	Required
`reject`	Specifies that the skill has rejected the request.	boolean	yes
`error`	The status of the response. For example, 200 (OK) or 404 (not found).	string	yes
`deleteSkillSession`	Instructs the routing core to end the user session. When the user session is ended, the context information that is stored is deleted.	boolean	yes
`captureInput`	Instructs the audio client to continue to listen for an utterance. If set to `true`, the audio client should not wait for a wake-up command.	boolean	yes
`speech`	The response from the skill.	object	yes
`card`	If present, a card provides supplementary information that enhances the text or audio response. You might use a card to display an image, play music, or provide a more detailed text response. In the card object, you can provide the URL to the image or music. The client device determines how to present this information to the user.	object	no
`skill`	Information about the skill that processed the response.	object	yes
`additionalInformation`	Additional information about the conversation, including context information and whether the skill is in conversation.	object	yes

Table 32 - Converse response parameters - speech

Parameter	Description	Type	Required
`text`	The response from the skill.	string	yes

Table 33 - Converse response parameters - card

Parameter	Description	Type	Required
`type`	The type of action that the card object invokes. In the example, the action is named `show-temp-map`. The card is used to display a temperature map when the user is at home.	string	yes

Parameter	Description	Type	Required
`content`	The attributes of the card. For example, you might add `image-url` or `music-url` to specify the image to display or the music to play.	array	yes

Table 34 - Converse response parameters - skill

Parameter	Description	Type	Required
`name`	The name of the skill.	string	yes
`entities`	The entities that were extracted from the utterance.	array	yes
`intents`	The intent that processed the utterance.	array	yes
`confidence`	The confidence score of the intent or entity that processed the request.	string	yes

Table 35 - Converse response parameters - entities

Parameter	Description	Type	Required
`entity`	The name of an entity extracted from the utterance.	string	yes
`value`	The value of the extracted entity.	string	yes
`confidence`	A confidence value that is associated the entity value. **Note**: A confidence score is always returned by a skill for each entity.	string	yes

Table 36 - Converse response parameters - intents

Parameter	Description	Type	Required
`intent`	The name of the intent that processed the utterance.	string	yes
`confidence`	The confidence score of the intent that processed the utterance	string	yes

Table 37 - Converse response parameters - additional information

Parameter	Description	Type	Required
`context`	Information about the context of the conversation with the user.	object	yes

Table 38 - Converse response parameters - context

Parameter	Description	Type	Required

Parameter	Description	Type	Required
`application`	The application ID and utterance context information.	object	yes
`session`	Information about the session, including session context information and skill context information.	object	yes

Table 39 - Converse response parameters - Application context

Parameter	Description	Type	Required
`id`	The unique ID of the application. The parameter is for future use.	string	yes
`attributes`	Includes any utterance context information. An empty attributes object is allowed.	object	yes

Table 40 - Converse response parameters - Session context

Parameter	Description	Type	Required
`id`	The ID of the session that is assigned by the routing core.	string	yes
`new`	Specifies whether a conversation with the user is already in progress.	boolean	yes
`skill`	Includes attributes representing the skill context	object	yes
`attributes`	Includes any session context information. An empty attributes object is allowed.	object	yes
`version`	The version of the session. Version information is set by the routing core. The version is always `1.0`.	string	yes

Table 41 - Converse response parameters - skill context

Parameter	Description	Type	Required
`attributes`	Includes any session context attributes. Include `"inConversation": true` to specify that the skill is expecting a response from the end-user. Allows the response from the user to be routed to the same skill for processing. An empty attributes object is allowed.	object	yes

Table 42 - Converse response parameters - session context

Parameter	Description	Type	Required
`attributes`	Attributes representing the session context information. An empty attributes object is allowed.	object	no

6. Converse response from the routing core to a client device

The JSON structure of the converse response from the routing core to a client device is as follows:


```
{
"reject": false,
"error": 200,
"deleteSkillSession": true,
"captureInput": true,
"speech": {
  "text": "In London city center, low temperature today will be 83 degrees
fahrenheit and high temperature today will be 109 degrees fahrenheit.",
"card": {
  "type": "show-temp-map",
  "content": {
    "id": "134325",
    "image_url": "https://www.bbc.co.uk/weather/2643743"
  }
},
"skill": {
  "name": "weather",
  "intents": [
    {
      "intent": "get-temperature",
      "confidence": 0.85514235496521
    }
  ],
  "entities": [
    {
      "entity": "weatherType",
      "value": "temperature",
      "confidence": 1
      },
      {
        "entity": "datePhrase",
        "value": "today",
        "confidence": 1
      },
      {
        "entity": "sys-location",
        "value": "london",
        "confidence": 0.962316
      }
  ],
  "confidence": 0.85514235496521
},
"additionalInformation": {
  "context": {
    "locationName": "at-home",
    "locationLatitude": 36.169941,
    "LocationLongitude": -115.139829
      }
```

```
      }
    }
  }
```

Table 43 - Converse response parameters

Parameter	Description
`reject`	Specifies that the skill has rejected the request.
`error`	The status of the response. For example, 200 (OK) or 404 (not found).
`deleteSkillSession`	Indicates if the routing core is ending the conversation with the skill.
`captureInput`	Specifies whether an audio client must continue to listen for an utterance. If set to `true`, the audio client does not wait for a wake-up command.
`speech`	The response to the utterance.
`card`	If returned by the skill, a card provides supplementary information that enhances the text or audio response.
`skill`	Information about the skill that processed the
response.	
`Additional information`	Extra context information about the conversation.

Table 44 - Converse request parameters - speech

Parameter	Description
`text`	The response to the utterance.

Table 45 - Converse request parameters - card

Parameter	Description
`type`	The type of action that the card invokes. In the example, the action is named `show-temp-map`. The card is used to display a temperature map when the user is at home.
`content`	The attributes of the card. For example, `content` might add `image-url` or `music-url` to specify the image to display or the music to play.

Table 46 - Converse response parameters - skill

Parameter	Description

Parameter	Description
`name`	The name of the skill that processed the request.
`entities`	The entities that were extracted from the utterance.
`intents`	The intent that processed the utterance.
`confidence`	The confidence score of the intent or entity that processed the request.

Table 47 - Converse response parameters - entities

Parameter	Description
`entity`	The name of an entity extracted from the utterance.
`value`	The value of the extracted entity.
`confidence`	A confidence value that is associated the entity value.

Table 48 - Converse response parameters - intents

Parameter	Description
`intent`	The name of the intent that processed the utterance.
`confidence`	The confidence score of the intent that processed the utterance

Table 49 - Converse request parameters - Additional information

Parameter	Description
`context`	Contains the utterance context.