API Reference

The RequestInfo JSON Format

The RequestInfo JSON Format is a subset of JSON values used to encode the JSON sent to the SoundHound Hound servers by clients.
When the client makes a request of a SoundHound Hound server, it sends a set of extra information in addition to the text or audio of the query itself. This is called the "request info". This JSON type specifies what should be in that request info.

All the fields in this type are optional, so clients can send a raw text or speech request without providing any additional information. However, the server can do a better job if it has more of this information, so clients should be constructed to provide all the fields specified here that the client is able to provide.

If clients have additional information that might be useful to the server, they may provide it in additional fields here and future versions of the server may be able to make use of that information.

The RequestInfo format uses JSON objects with the following fields:

Field Name	Type	Optional?	Default
`"Latitude"`	`-90 <= rational="" <="90" span="">`	optional	no default
	This is the client's best guess about the latitude of the user's current position, in degrees north of the equator. Negative values indicate positions south of the equator. If the client doesn't have a way to know its position, this field should be omitted. If the client has an approximate location, even if it is not very accurate, it should still use that approximate location to fill in this field. This field is assumed to be in the WSG84 coordinate system.		

"Longitude"	**-180 <= rational="" <="180" span="">**	optional	no default
	This is the client's best guess about the longitude of the user's current position, in degrees west of the prime meridian. Negative values indicate positions east of the prime meridian. If the client doesn't have a way to know its position, this field should be omitted. If the client has an approximate location, even if it is not very accurate, it should still use that approximate location to fill in this field. This field is assumed to be in the WSG84 coordinate system.		
"PositionTime"	**finite integer**	optional	no default
	This is the time at which the client got the position fix used for the location in the "Latitude" and "Longitude" fields, in Unix time (i.e. seconds since midnight January 1, 1970 UTC not counting leap seconds). The motivation here is that often on mobile devices GPS is available but expensive in terms of battery power, so a mobile client will often be designed to only turn on the GPS to get a position fix periodically. So the position information may be stale, and this field tells the server just how stale it is.		
"PositionHorizontalAccuracy"	**rational >= 0.000000**	optional	no default
	This field provides the client's best estimate of the accuracy, in meters, of the position reported in the "Latitude" and "Longitude" fields.		
"Street"	**string**	optional	no default
	Sometimes clients have location information available not just in the form of a latitude and longitude but also have the street address available. Clients that have this information available can provide the street in this field.		
"City"	**string**	optional	no default
	Sometimes clients have location information available not just in the form of a latitude and longitude but also have the street address, or at least the city, available. Clients that have this information available can provide the city name in this field.		

"State"	**string**	optional	no default
	Sometimes clients have location information available not just in the form of a latitude and longitude but also have the street address, or at least the state, available. Clients that have this information available can provide the state name in this field. Note that this should only be used for locations that are within states of the United States. For other locations, this field should be omitted.		
"Country"	**string**	optional	no default
	Sometimes clients have location information available not just in the form of a latitude and longitude but also have the street address, or at least the country, available. Clients that have this information available can provide the country name in this field.		
"ControllableTrackPlaying"	**boolean**	optional	false
	This field specifies whether the client is currently playing a music track that it is capable of controlling.		
"TimeStamp"	**finite integer**	optional	no default
	This is the time at which the client believed it was starting the request to the server, in Unix time (i.e. seconds since midnight January 1, 1970 UTC not counting leap seconds).		
"TimeZone"	**string**	optional	no default
	If the client believes it knows what time zone it is in, it should send that information in this field, in the form of an Olson name.		
"ConversationState"	**object (see below)**	optional	no default
	If the client believes there is a reasonable liklihood that the current request to the server is a continuation of a conversation, and the last response from the server in that conversation had a "ConversationState" field, the client should send back exactly the value of that "ConversationState" field in this field.		
	Type details: This field uses JSON objects, with any JSON object at all allowed.		
"ConversationStateTime"	**finite integer**	optional	no default
	This is a time stamp associated with the "ConversationState" field, in		

	Unix time (i.e. seconds since midnight January 1, 1970 UTC not counting leap seconds). Please note that the clients should not set this field based on its own understanding of time. If the client sets the "ConversationState" field, it should also set "ConversationStateTime" to the value from the "ConversationStateTime" that the server sent in the same object as the "ConversationState" that the client is echoing back. If the "ConversationState" isn't set by the client, neither should the "ConversationStateTime" field be.		
"ClientState"	**ClientState (ClientState.html)**	optional	no default
	This field is used to communicate information that is dependent on the current state of the client.		
"PreferredImageSize"	**array (see below)**	optional	no default
	This field provides a way for the client to specify its preferred image size. If present, it should be an array of two positive integers. The first is taken as the width and the second as the heigh. Both are in pixels.		
	Type details: This field uses only JSON arrays. The array may have from 2 to 2 elements (inclusive). Each element of the array uses only JSON integers. Any integer greater than or equal to 1 is legal here.		
"Language"	**string**	optional	no default
	This field allows the client to inform the server of the user's preferred language. Please note that for most purposes the server currently ignores this value and assumes the query comes in English and it sends its results as English. It is possible that in the future the server will use this field more, and become less English-centric.		
"UnitPreference"	**string enumeration (see below)**	optional	no default
	Type details:		

	This field uses only a fixed, finite number of JSON strings to encode an enumeration. The legal values are: • "US" -- The United States (also known as Imperial) measurement system (miles, pounds, etc.). • "METRIC" -- The metric system (kilometers, kilograms, etc.).		
"ClientID"	**string**	optional	no default
	This string should be set to a value that distinguishes one kind of client from another. A developer creating a new client should choose a name for that client that is reasonably descriptive and which is not likely to be the same as that used by another client. If there are multiple versions of the client with the same "ClientID", or even multiple copies of the same version of the client, that's OK. This field, alone or in conjunction withthe "ClientVersion" field, can be used by the server to help debug problems that occur only with specific clients, to track usage of different clients, and to provide client-specific services such as work-arounds for known bugs or limitations in particular clients. This is analagous to the product name in the User-Agent field of an HTTP request.		
"ClientVersion"	**see below**	optional	no default
	This string should be set to a value that specifies which version of the client made the request. If this field is set, the "ClientID" field also should be set, and the version is relative to the name from the "ClientID" field. For example, "ClientID" might be "AndroidHound" and the "ClientVersion" might be "3.12.4beta". See the description of the "ClientID" field for how the information in this field might be used. This is analagous to the product version in the User-Agent field of an HTTP request.		

<table>
<tr><td></td><td colspan="3">Type details:
This field uses one of the following formats:

• It uses only JSON strings. Any JSON string is legal here.
• It uses only JSON integers. Any integer greater than or equal to 0 is legal here.</td></tr>
<tr><td rowspan="2">"DeviceID"</td><td>string</td><td>optional</td><td>no default</td></tr>
<tr><td colspan="3">If the client has a device-specific ID, it should send it in this field. It is intended for keeping track of which requests are coming from the same client. This can be used for logging and debugging problems, as well as to improve the server performance by learning about particular client devices.</td></tr>
<tr><td rowspan="2">"FirstPersonSelf"</td><td>string</td><td>optional</td><td>"Hound"</td></tr>
<tr><td colspan="3">The client can optionally set this field to specify how the system should refer to itself in written responses. This will also be used for spoken responses if the "FirstPersonSelfSpoken" field is not set.</td></tr>
<tr><td rowspan="2">"FirstPersonSelfSpoken"</td><td>string</td><td>optional</td><td>no default</td></tr>
<tr><td colspan="3">The client may use this field to specify a variant of what's in "FirstPersonSelf" that is to be used in spoken responses. This is in case the written version doesn't work well for text-to-speech. This should not be a totally different name from "FirstPersonSelf", just a variant to help the system properly pronounce that name, if necessary.</td></tr>
<tr><td rowspan="3">"SecondPersonSelf"</td><td>array (see below)</td><td>optional</td><td>["Hound"]</td></tr>
<tr><td colspan="3">The client can optionally set this field to specify names the user may use to address the system.</td></tr>
<tr><td colspan="3">Type details:
This field uses only JSON arrays. The array may have any number of elements.

Each element of the array uses only JSON strings. Any JSON string is legal here.</td></tr>
<tr><td rowspan="2">"SecondPersonSelfSpoken"</td><td>array (see below)</td><td>optional</td><td>no default</td></tr>
<tr><td colspan="3"></td></tr>
</table>

	The client may optionally set this field to specify spoken forms of the names in the "SecondPersonSelf" field. This field should not be set uless "SecondPersonSelf" is also set, and it should have the same number of element. Each element of this array should specify the pronunciation of the corresponding element of the "SecondPersonSelf" array. This field should only be used if one or more of the names in the "SecondPersonSelf" array has an unusual pronunciation that cannot be determined by the system from the text form.		
	Type details: This field uses only JSON arrays. The array may have any number of elements. Each element of the array uses only JSON strings. Any JSON string is legal here.		
"UserID"	**string**	optional	no default
	This field should be used by the client to identify the user making the request. The server can keep track of information about specific users, such as their contact lists, to do a better job in some cases. The server can also use this information to help in debugging problems.		
"RequestID"	**string**	optional	no default
	This field should be filled in with a different unique string for every request made to the server. It is strongly recommended that every client fill in this field for each request. It can be used by the server for logging and debugging problems. If you have a problem with the server and report a bug or other issue, it will be much easier to track down what happened if a unique RequestID was provided in the request and that RequestID is given in the bug report. On many platforms, a good way to generate a request ID string is to use a library that implements the UUID (http://en.wikipedia.org/wiki/Universally_unique_identifier) specification.		
"SessionID"	**string**	optional	no default

	This field should be filled in by the client with a unique string that lasts potentially across multiple requests that are all considered by the client to be one session. A session is a sequence of requests that come without too big a break in between and across which conversation state, if any, is preserved. The user closing the app and restarting it, or not interacting with it for half an hour, might be considered gaps by the client that constitute the start of a new session. When a new session ID is used, no conversation state should be sent. On many platforms, a good way to generate a session ID string is to use a library that implements the UUID (http://en.wikipedia.org/wiki/Universally_unique_identifier) specification.		
"ResultUpdateAllowed"	**boolean**	optional	false
	This field specifies whether the client can accept updating of result information. If true, then the server might in some cases send an initial result but keep the connection open and later send an updated version of the result to replace the original result. The idea here is to improve the user experience by allowing the client to immediately show a partial result, or at least that the query was understood and the proper information is being found, in the case that it takes a few seconds to fetch the requested information or to take some other action on the server side. If this field is not present or is set to false, the server will wait until it has fetched all the relevant data and then sends it in a single result JSON object of type HoundServer (HoundServer.html). If this field is present and set to true, the server may still send all the data in a single HoundServer (HoundServer.html) object. But in some cases it may also send an initial HoundServer (HoundServer.html) object and keep the connection open and send updates in HoundUpdate (HoundUpdate.html) JSON objects. It uses the HTTP chunking protocol to send the different objects in this case. The HoundServer (HoundServer.html) and HoundUpdate (HoundUpdate.html) objects contain information specifying whether there are additional updates coming.		

"PartialTranscriptsDesired"	**boolean**	optional	false
	This field specifies whether the client wants to get partial transcripts for an audio query as the query is still going on. If this field is present and set to true, then the server may send partial transcripts. These partial transcripts will be in HoundPartialTranscript (HoundPartialTranscript.html) JSON objects and will come before the HoundServer (HoundServer.html) object. There can be any number of HoundPartialTranscript (HoundPartialTranscript.html) objects before the HoundServer (HoundServer.html) object. The server will use HTTP chunking and keep the connection open to send these multiple JSON objects.		
"MinResults"	**integer >= 1**	optional	1
	This field specifies that the client would like to be given at least the specified number of results. Those different results are for different interpretations of the query. For text queries, it's for different parses of the text. For audio queries, its for a combination of different parses, some of which may be different parses of the same transcription and some of which may be based on different transcriptions. These different results are put in the "AllResults" field of the HoundServer result object. For example, a 3 for "MinResults" means that the server should try to fill in at least three elements of the "AllResults" field in the response. The server is free to send fewer results if it can't find that many different interpretations of the query. The intention here is to allow multiple results to be returned to the client to let the client have the option of letting the user choose which result is for the query as the user meant it.		
"MaxResults"	**integer >= 1**	optional	1
	This field specifies that the client would like to be given at most the specified number of results. Those different results are for different interpretations of the query. For text queries, it's for different parses of the text. For audio queries, its for a combination of different parses, some of which may be different parses of the same transcription and some of which may be based on different transcriptions. These different results are put in the "AllResults" field of the HoundServer result object. For example, a 5 for "MaxResults"		

	means that the server should never return an "AllResults" field in the response with more than five elements. The intention of sending multiple results back is to let the client have the option of letting the user choose which result is for the query as the user meant it. Each client will have some limit on how many choices it will show the user, so there's no point in having the server send back more choices than that; this field lets the client communicate that limit to the server. Note that the value of the "MaxResults" field should always be greater than or equal to the value of the "MinResults" field. If they are different, it means the server should use the "MinResults" number of results if it's fairly confident that the answer is one of those, but can use up to "MaxResults" if the server has less confidence and thinks there are that many strong possibilities.		
"ObjectByteCountPrefix"	**boolean**	optional	false
	If this flag is set to true, it specifies that the server should put a byte count prefix before each top-level JSON object in the response. There is always one HoundServer top-level object and there might also be HoundUpdate (HoundUpdate.html) and HoundPartialTranscript (HoundPartialTranscript.html) objects. If this flag is set to true, the byte counts will be in the same format as the byte counts that prefix chunks in the HTTP protocol. Note that this means there are byte counts layered on top of byte counts. The server already uses the HTTP chunking format to send back the JSON objects, with one object per HTTP chunk. The additional layer of byte counts is redundant. It's optionally provided to help clients that are built on top of an HTTP layer that abstracts away the chunking in HTTP itself.		
"ClientMatches"	**array (see below)**	optional	no default
	If present, this field specifies patterns that the server should try to match in the query. This allows a client to extend what the server understands to arbitrary additional language patterns. An example of how the client might use this would be for client-specific voice controls. The client might let the user say "options menu" or "show me the options menu" to get the same effect as		

	clicking "Options" from a pull-down menu. This lets an app give the user full voice control over every available feature without having to modify the server to know about the details of the app. If the query matches a pattern specified here, and there was no higher-weight match of another sort, then the server will return a result of type ClientMatchCommand (ClientMatchCommand.html).		
	~~Type details:~~ This field uses only JSON arrays. The array must have at least 1 element but may have any number of additional elements. Each element of the array uses values of type ClientMatch (ClientMatch.html).		
"ClientMatchesOnly"	**boolean**	optional	false
	If this flag is set to true, it specifies that the server only match patterns specified in the "ClientMatches" field, not any of the built-in patterns the server understands.		
"UseContactData"	**boolean**	optional	true
	If this flag is set to false, this request is handled as if there was no contact data uploaded for this user, regardless of whether there actually was such contact data uploaded.		
"UseClientTime"	**boolean**	optional	false
	If this flag is set to true, it specifies that the server should do any time calculations necessary based on the time specified by the client in the "TimeStamp" and "TimeZone" fields of this request info. If this flag is set to false, the server will do time calculations based on the "TimeZone" field of this request info, but using the UTC time as the server understands it. This field is intended primarily to ease testing and debugging, so the client can get back repeatable results.		
"ForceConversationStateTime"	**finite integer**	optional	no default
	If present, this field specifies that the server should use the specified value in the "ConversationStateTime" fields for all conversation		

	states, including those in dynamic responses, returned by the server. When this field isn't present, the server sets those "ConversationStateTime" fields to its own idea of seconds UTC Unix Time. This field is intended primarily to ease testing and debugging, so the client can get back repeatable results.

Additional fields with other names are also allowed.