# Text Description automatically generatedSJSON Reference

Version: 1.01 – 2023/06/16

# SJSON Basics

SJSON (Song Javascript Object Notation) is the format used by Band BoomBox to store information related to songs, such as their title, artist, BPM, charts and note sequences. One SJSON file exists for every playable song in the game. When Band BoomBox starts up, it will scan for and load .sjson files in the configured song folders (see the below section).

SJSON files are valid JSON text, and as such they can be loaded and manipulated by any tool designed to work with JSON files. Any SJSON file that contains invalid JSON text will be ignored by the game.

# Where SJSON files are located

SJSON files are typically loaded from one or more of Band BoomBox’s song folders. The game uses the following folders as song folders by default:

* [Save Data Folder]\Songs
  + By default, any songs created in the editor will be saved here.
* [Installed Folder]\Band BoomBox\_Data\StreamingAssets\Songs

**[Save Data Folder]** refers to the folder where the game’s save data is stored. This folder includes settings, profiles, high scores as well as songs. On Windows, this folder is located at C:\Users\[Your Username]\Documents\My Games\Band BoomBox . It can also be accessed by selecting “Open Save Data” from the Options menu, in the Advanced tab.

**[Installed Folder]** refers to the folder where the game is installed – the one that contains **Band BoomBox.exe.**

All subfolders in the configured song folders will be scanned as well. The songs contained in these folders can be organized however you like, but the official standard is to have one subfolder per artist, and to place all songs for that artist in the same folder.

**Note:** The game can be configured to load from additional song folders by editing the **Settings.json** file manually, located in [Save Data Folder].

# SJSON Fields

## Root

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ID | String (Guid) | A unique identifier for the song. Note that every song should have its own unique ID. |
| Title | String | The title of the song, as it should appear in game. |
| Subtitle | String | (Optional) a second line for the song’s title. Typical usages include things like “(feat. Guest Artist)” or “(Awesome Remix)” |
| Artist | String | The name of the musician or band that created the song’s audio (not the chart). |
| ChartAuthor | String | The name of the player that created this chart. |
| AudioFile | String | The name of the file that contains the song’s audio. It must be located in the same folder as the .sjson file. The game supports both .ogg and .mp3 audio formats. |
| Bpm | Float | The number of beats per minute (BPM) in the song. Zero or negative values are considered invalid.  **Note: Only songs with a constant BPM are currently supported. Do not attempt to use a song with a variable BPM.** |
| Offset | Float | The position of the first playable beat of the song, in seconds, relative to the start of the audio. A value of 1.0 means that the first playable beat occurs 1 second after the audio starts playing.  **Note: A correct and accurate offset is essential!** Since it controls the position of every note in the song, if it is off by even a slight amount (0.02 seconds or more), the notes of the song will be noticeably harder to hit accurately.  **An offset of 0 is considered invalid.**  In general, the offset should be positioned precisely on a “down” beat, at least 2 seconds into the song (to give the player time to get ready for the first note). The offset can be much later than this though, if desired. Consider setting an AudioStart point if this is the case. |
| AudioStart | Float | (Optional) The position, in seconds where audio playback should begin. When this song is played, the audio is skipped forward to this point. This is useful songs with high Offset values. Otherwise, use zero to start audio playback from the beginning of the song.  A good value for this is typically 4 measures (or 16 beats) earlier than the Offset point. The editor in Band BoomBox automatically sets the offset to this value whenever the “Auto AudioStart” option is enabled. If this would result in a negative value, it is instead clamped to zero.  A zero AudioStart value is valid. **Negative values are not.** |
| Length | Float | The position of the end of the playable portion of the song, relative to the start of the audio. This controls the position of the Finish Line. One option is to set this position to one beat after the last playable note, though this position does not need to be precise.  **Note: Do not place notes after this point! Any**  **notes after the finish line are considered invalid.** |
| BeatsPerMeasure | Int | The number of beats per measure. The default value is 4, which indicates 4 beats per measure, also known as a “4/4 signature”. Most songs use this signature. |
| Issues | String | (Optional) Used to indicate if this song currently has any known issues, such as “WIP” or “Hard Difficulty Only”. Leave blank to indicate no issues. |
| Version | Int | The version of this SJSON file. If any significant changes are made to the SJSON format, this version should be incremented. Doing this will ensure that any existing high scores for this song are cleared. |
| Sections | Dictionary <Double, String>  **Note: This is represented as an Object in JSON.** | The named sections of this song (verses, choruses, drops etc).  The Key refers to the position of the section’s start, in beats, with 0 being the first playable beat.  The Value is the name of this section, which lasts until the next section, or the end of the song. |
| SongCharts | Array <SongChart> | The collection of charts for this song. See the “SongChart” section. |

## SongChart

A SongChart represents a specific sequence of notes in the song. Each song typically has multiple SongCharts defined for it, but there can only be one SongChart for a specific song, group, and difficulty. Individual high scores are saved per SongChart.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Group | String | Currently unused. Default is “Main”. |
| Difficulty | Int (Enum) | The difficulty class for this chart:   * 0 = Beginner * 1 = Medium * 2 = Hard * 3 = Expert * 4 = ??? * 10 = Extra   Other values are considered invalid.  **Note: There can only be one chart per Group and Difficulty. Duplicates are considered invalid.** |
| DifficultyLevel | Int | The difficulty level of this song, from 1 to 99. |
| Notes | Array <String> | The notes for this chart. If omitted or blank, notes will be generated randomly when this song is played. See the “Song Notes” section for how this field is structured. |

## Song Notes

The Notes field is optional and defines the sequence of notes for its parent SongChart. If omitted or blank, notes will be generated randomly when this song is played. Otherwise, this field will contain an array of Strings.

Example:

"Notes": [

"0500 0000",

"00A0",

"0040",

"0040",

"00A0",

"0GB0 0000",

"0000 0000",

"04G0 00A0",

"0400 0000",

"0080 0000",

"0000 0000",

"0DG0 0070",

"0D00 0000",

"00E0 0000 0000 0000",

"0000 0000 0000 0000",

"07G0 0000 0000 0700",

"0000 0000 0700 0000"]

The notes field is divided into **Blocks, Lines and Lanes.**

* Every item in the array is considered a **Block.** In the above example, each block is on a separate line, but this is optional. A song chart can have any number of blocks in its notes collection.
* Blocks can have one or more **Lines.** A Line consists of four digits. Each line in a block is separated by a space.
* Each line has four **Lanes,** with each digit in a line representing one lane.

**"07G0 0000 0000 0700" Block**

**"07G0 0000 0000 0700" Line**

**"07G0 0000 0000 0700" Lane**

## Blocks

Each block defines one beat of the song’s note sequence. In other words, all notes that appear from one beat of the song until the next are contained in a single block. The first block represents the first beat, the second block the second beat, and so on. A song chart can have any number of blocks in its Notes collection. The timing of each note in the block is determined by its **line,** explained below. The type of each note (whether it is “Left”, Right”, “A” etc) is determined by its **lane.**

## Lines

Every block in the chart is divided into lines. A line consists of four digits, and each line is separated by a space. A block can have any number of lines, though most songs would not typically need more than four lines per block. Each line can have one or more notes, or it can be left empty.

Whereas blocks control which beat notes are located in, each line controls the location of notes **inside** the current beat. The first line represents the beginning of the current beat. If notes only need to appear at this point, then only one line is required. For more precision, add more lines. Whenever a block contains more than one line, the beat is split equally among each line, as shown in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Precision** | **Beat Fractions** | **Lines** | **Example** |
| 4th Notes | 1 beat | 1 | 0700 |
| 8th Notes | ½ beats | 2 | 0700 0070 |
| 16th Notes | ¼ beats | 4 | 0700 0000 0070 0A00 |

Note that the number of lines can vary between blocks. Its is typical for some blocks to have only one line, and others to have more, depending on how much precision is needed for each block.

Any line can be left empty by filling it with four zeros. Otherwise, to place one or more notes in a line, see the Lanes section below.

## Lanes

Each line consists of digits, with each digit representing one lane in the game.

* The first digit controls the top lane.
* The second digit controls the middle lane.
* The third digit controls the bottom lane.
* The fourth digit is currently unused.

The digit in each lane controls which note (if any) should appear on this lane, at this point in the song. A 0 indicates a blank.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0000 | Empty (no notes) |  |  |  |  |
|  | **Top Lane** |  | **Middle Lane** |  | **Bottom Lane** |
| 1000 |  | 0100 |  | 0010 |  |
| 2000 | (Hold) | 0200 | (Hold) | 0020 | (Hold) |
| 4000 |  | 0400 |  | 0040 |  |
| 5000 | (Hold) | 0500 | (Hold) | 0050 | (Hold) |
| 7000 | ??? | 0700 |  | 0070 |  |
| 8000 | ??? | 0800 | (Hold) | 0080 | (Hold) |
| A000 |  | 0A00 |  | 00A0 |  |
| B000 | (Hold) | 0B00 | (Hold) | 00B0 | (Hold) |
| D000 | ??? | 0D00 |  | 00D0 |  |
| E000 | ??? | 0E00 | (Hold) | 00E0 | (Hold) |
| G000 | Hold Release | 0G00 | Hold Release | 00G0 | Hold Release |

“Hold Release” notes indicate the end of a Hold note on the same lane. Their colour will change to match the Hold that appears before it. **Every Hold note must be followed by a Release note**. Avoid placing any notes between a Hold note and its associated Release note on the same lane. However, placing notes in other lanes during a hold note is acceptable, and encouraged on higher difficulties.

**Note:** Multiple notes can appear on the same line if they are on separate lanes. For example: “A700” indicates “RB” and “Down” notes at the same time. Multiple notes on the same line and lane are not allowed.

Typically, the first lane (representing the top lane in gameplay) is only used in Expert difficulty charts. On the other hand, Beginner difficulty charts should only use the 0100, 0200, 0010 and 0020 notes. For more information on which notes should be used for each difficulty, consult the How To Play section inside the game.