Interface Specification

This document contains descriptions for all front-facing interface functions

# Exceptions

Exceptions that can be thrown by the backend.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Usage** |
| ToastError | Error message shown as toast popup next to mouse pointer | Non-critical errors (e.g. “nothing to undo”) |
| ProjectReadError | Thrown when a project is in an invalid format |  |
|  |  |  |

# Types

Data types used by the back-end.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Usage** |
| MidiLength | Length of a MIDI object (in ticks) | Lengths of notes, phrases |
| MidiPos | Position of a MIDI object (in ticks) relative to the parent body | Positions of MIDI notes relative to the start of a phrase, positions of phrases relative to the start of the project |
|  |  |  |

# Objects

## Projects

Methods related to multiple project management, contained inside the static class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Description** | **Arguments** | **Returns** | **Exceptions** |
| addProject | Create a new project | (none) | (int) projID: identifier that can be used to refer to this project | (none) |
| Open an existing project | (str) projDir: directory of the project to open | IOError: the directory doesn’t exist  ProjectReadError: the project folder doesn’t contain a valid file |
| Open an existing compressed project | (str) projFile: file location of the file to load | IOError: error reading file (file doesn’t exist or access denied)  ProjectReadError: the project file (when decompressed) doesn’t contain a valid file |
| closeProject | Close an open project (without saving) | (int) projID: project’s identifier | (none) | (none) |
| saveAllProjects | Saves every project that requires it | (none) | (none) | IOError: error saving project file(s) |

## Project

Methods related to a single project, contained inside the class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Description** | **Arguments** | **Returns** | **Exceptions** |
| saveProject | Saves a project (if it has been changed) | (none) | (none) | IOError: error saving project file |
| exportProject | Compiles a project to a MIDI file | (str) fileName | (none) | IOError: error saving MIDI file |

## Details

Methods related to a project’s details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Description** | **Arguments** | **Returns** | **Exceptions** |
|  |  |  |  |  |

## History

Methods related to a project’s history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Description** | **Arguments** | **Returns** | **Exceptions** |
| undo |  |  |  |  |
| redo |  |  |  |  |
| setUndoPos | **Set position in undo history** |  |  |  |

## Phrases

Methods relating to phrase management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Description** | **Arguments** | **Returns** | **Exceptions** |
| newPhrase | **Create a new phrase** | **(none)** | **(int)** phraseID |  |
| importPhrase | **Import MIDI file as phrase** | **(str)** fileName | **(int)** phraseID | IOError: error reading MIDI file (file doesn’t exist or access denied) |
| removePhrase | **Remove a phrase** | **(int)** phraseID | **(none)** |  |
| duplicatePhrase | **Make a copy of a phrase** | **(int)** phraseID | **(int)** newPhraseID |  |

## Phrase

Methods relating to a single phrase object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function | Description | Arguments | Returns | Exceptions |
| renamePhrase | **Renames a phrase** | (str) newName | **(none)** | ToastError: newName is empty |
| getLength | Returns the length of a phrase | (none) | (MidiLength) length |  |
| getHeight | Returns the number of tracks in a phrase | (none) | (int) number of tracks |  |