**~~CRINGELINER~~ DEADLINER API**

Работа со списком

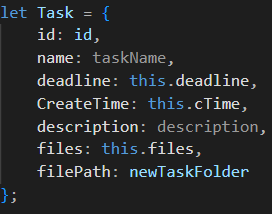
Сам по себе задачник расположен в src/todo.js. В нём как раз таки и есть его логика. При импорте этого файла возвращается объект класса TodoList

*#пометка. эта хуита не работает по абсолютно тупой причине*

Но для взаимодействия со списком в смотри в src/preloade.js он отвечает за взаимодействие фронта и бека

По подробнее:

Формат каждого таска:



Id – сам создаётся при добавлении таска

Name – название

Deadline – время окончания

createTime – время создания задания

description – описание

files – массив файлов

filePath – путь к папке с файлами задания

# **ADD**

Func add (task, deadline, description, files) => void

Добавляет таск в список. Ничего не возвращает

Task – название задания

Deadline – ну, собственно дедлайн задания

*# время лучше передавай в формате (dd.mmyyyyThh:mm), с этим тут беда, #ну не только с этим, но сейчас я про время говорю*

Description(опц.) – это описание

Files(опц.) – закреплённые файлы. Передавать надо не сам файл, а путь к нему

# **getTask**

Func getTask (id) => [Task]

Принимает id задания. Возвращает массив содержащий задание

Id – id задания

# **Delete**

Func delete (id) => Boolean

Удаляет задание по id. Если удалено возвращает true, иначе false

# **Get**

Func get() => [Tasks]

Возвращает массив со всеми заданиями

# **Edit**

Func edit (id, name)=> void

Принимает id задания и новое название

#я что то забыл про эту функцию…..пока только имя меняет, потом доделаю

# **removeFile**

func removeFile (name) => void

удаляет файл указанный в name. сам файл ищет в папке для файлов задания

# **openFile**

openFile(name)=>void

открывает файл name. сам файл ищет в папке для файлов задания

# **ДИАЛОГОВЫЕ ОКНА**

Кстомный диалоговые окна находятся в src/CustomDialog/CustomDialog.js. импорируется класс, но для создания окон это не надо, это 2 статичных метода

## **showMessage**

showMessage(opt{titile, desc, button, interfacePath}, callback) => Number

создаёт диалоговое окно с переданым в interfacePath окном. В interfacePath надо передать путь к html. Title это название окна. Desc- содержимое. В callback возвращается номер нажатой кнопки. Номер в том порядке, в котором передаётся в массиве

## **showError**

showError (opt{title, desc, interfacePath }) => void

создаёт диалоговое окно, просто диалоговое окно. Дизаин так же через interfacePath