DevTools Documentation

INDEX

INTRODUCTION

DevTools, Graphical DevTools, aims to be a simple software to generate code for Blockly and other graphical programming alternatives. Specifically, DevTools aims to allow for user created add-ons (plugins) for PatetLex applications. DevTools generates inerrant JSON (.json) code necessary for graphical programming used by other PatetLex apps. Stand alone, DevTools generated code can be used for other Blockly implementations, however, the mappings (code, dependencies) may differ.

GETTING STARTED

On launch, the user is granted with *image A*. The Blockly workspace button solely opens a new window with Blockly functions; represented by *image B.* The user can manipulate his new block to his ability. Upon finishing, the user can then save the code given in the top left. This code can be used for Blockly plugin implementation, and soon PatetLex style plugins. For details on manipulating Blockly blocks, see *INSIDE BLOCKLY WORKSPACE*.

INSIDE BLOCKLY WORKSPACE

The user can manipulate their Blockly block graphically in many ways.

- Name - name of block, often formatted as id (“Hello World!” = “hello_world!”).

- Category - category of block (mapped as PatetLex), often formatted as id.
- Dependencies - dependencies of block (mapped as PatetLex) *(Note: While this feature isn't present in DevTools, when block dependencies are used inside valid apps the block will only be able to be used if the provision is present; see below table for examples).*

Dependencies	*Provisions*	*Satisfied (will the block be able to be used)*
User		*No*
User	*User*	*Yes*
User	*User | Message*	*Yes*
User	*Message*	*No*

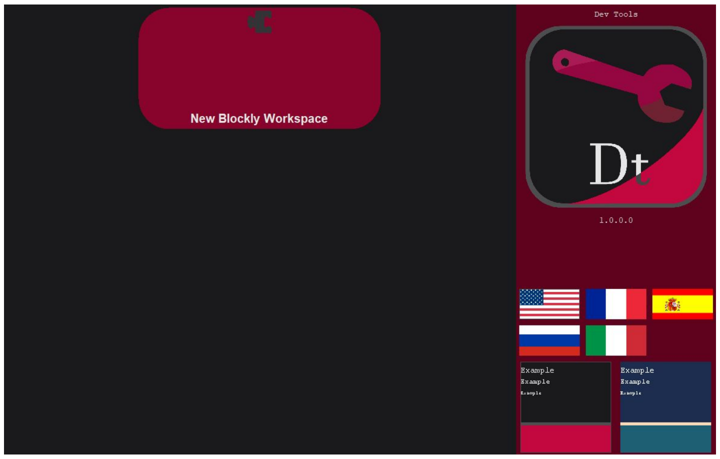
- Arguments - arguments of block
    - Type;
        - DUMMY - has no visible effect on block, raw text.
        - VALUE - value input *(Note: To get the value in code use **$[value%NAME]**).*
        - STATEMENT - statement input *(Note: To get the statement in code use **$[statement%NAME]**).*
    - Name - name of block, often formatted as id; used to identify the value/statement.
    - Text - text of block, can be empty *(Note: This option has no use in PatetLex applications).*
    - Check - check for blocks *(Note: Used in value and statement blocks)*, used in conjunction with other blocks output. I.e a block with the output of "Number" can't be placed as a value in the field with the check of "String".
- Code - code of block
    - Module - specifies code format, in other words the code language. I.e "JAVA" or "JAVASCRIPT"

    - Code - the template of code *(Note: To access values use $[value%NAME]; to access statements use $[statement%NAME]).*
- Field mode;
    - INLINE - Values appear inline to each other.
    - EXTERNAL - Values appear externally.
- Block type;
    - OUTPUT - Block outputs values *(Note: This type is often used with block output).*
    - PROCEDURAL - Block goes from top to bottom, procedural.
- Output - output type of block, such as “Number” or “String” *(Note: One can create [use] any text and all features will work).*
- Color bar - color of block.

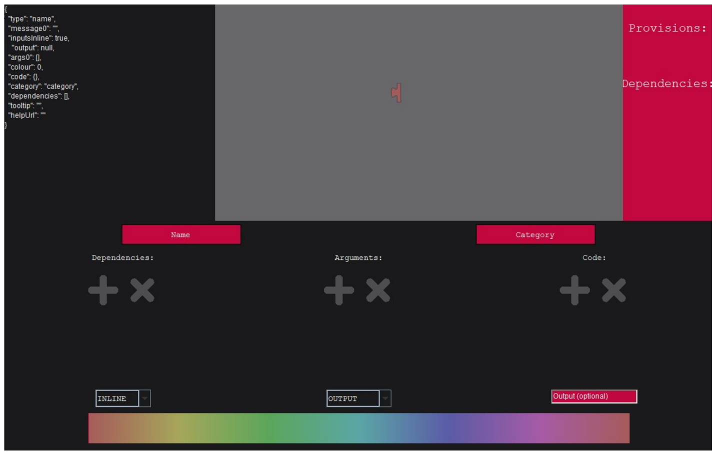
IMAGES

(Note: Images may not be exact and updated, however if the app looks abnormal, contact us)

A:



B:



```
{
 "type": "name",
 "message0": "",
 "inputsInline": true,
   "output": null,
 "args0": [],
 "colour": 0,
 "code": {},
 "category": "category",
 "dependencies": [],
 "tooltip": "",
 "helpUrl": ""
}
```

EXAMPLE VIDEO

An example of DevTools being used can be found at:

https://www.patetlex.com/res/vid/devtools.mp4

Contact: support@patetlex.com

Written by: Tye Phoenix

Date: 22 January 2021

Updated to DevTools version: 1.0.0.0

In regards: DevTools Documentation

From: PatetLex#19117