1 TouchSwitch AppData specification table

Object	Mandatory/ Optional/ App set/ Choose	Composition/Options	JSON	Comments

AppData
 	
Main data object for app
	M
	A
	A
	O
	O
		public ArrayList<GridControllerData> controllers;
		public String hashed_password;
		public String salt;
		public Long screen_timeout;
		public boolean debug_mode = false;
			{
			"controllers": [{}, {}]
			"hashed_password":
			"some_hashed_password"
			"salt": "some_salt"
			"screen_timeout": 5000
			"debug_mode": false
			}
				Hashed password and salt will be stored by the app, so not needed when initially creating the app.

GridControllerData
 	
Information for each controller (view)
	A
	M
	C.1
	C.1
	
	O
	
	O
	O
	O
	O
	
	O
	O
	O
	
	O
		public static final String *KEY* = "GridControllerData";
		public String controller_id;
		public GridControllerDataVariant portrait;
		public GridControllerDataVariant landscape;
		
		public ButtonType layout_click_type = ButtonType.*NON_CLICKABLE*;
		//layout click data
		public CommandData command_data;
		public ArrayList<MultipleButtonData> sub_buttons;
		public SubButtonPopupData sub_button_layout_info;
		public String destination_controller_id;
		//init command data
		public Boolean send_command_on_init;
		public CommandData layout_init_command;
		public Integer footer_logo_width_pixels;
		//for querying information from the control system on a timer
		public ArrayList<Event> events;
			{
			controller_id": "command_controller",
			"events": [{},{}],
			"footer_logo_width_pixels": 80,
			"layout_click_type": "NON_CLICKABLE",
			"portrait":{},
			"landscape":{},
			"send_command_on_init":true,
			"layout_init_command":{},
			}
				Ensure one of the **portrait** or **landscape** GridControllerDataVariant objects is defined (depending on what orientation you plan to use the app in).
				
				For layout_click_type:
				• **NON_CLICKABLE** – All subsequent fields Optional
				• **COMMAND** – command_data now Mandatory
				• **SUB_BUTTON** – sub_buttons and sub_button_layout_info now Mandatory
				• **VIEW_TRANSACTION** – destination_controller_id now Mandatory
				• **DISMISS_VIEW** – All subsequent fields Optional.
				• Other ButtonType values not valid.

GridControllerDataVariant
 	
Information for each orientation of the view
	M
	M
	M
	O
	O
	O
	O
	O
	
	O
	O
	O
	O
		public ArrayList<GridButtonData> buttons = new ArrayList<>();
		public Integer row_count;
		public Integer column_count;
		public Boolean plan_mode; *//will have plan view background*
		public Boolean sub_button_footer;
		public Boolean sub_button_relative; *//show at touch point*
		public String background_file_name;
		public Integer background_colour;
		
		public Boolean plan_touch_mode; *//will have touch listener*
		public String plan_control_file_name;
		public ArrayList<PlanColourCode> plan_colour_codes;
		public String plan_highlight_colour = **"#D041B19C"**;
			{
			"buttons": [{}, {}],
			"column_count": 2,
			"row_count": 5
			}
				If **plan_mode** is set background_file_name is Mandatory
				
				sub_button_footer locks any popup windows to the bottom of the view.
				
				sub_button_relative shows popup windows to the point touched in plan_touch_mode.
				
				background_colour is a signed integer representation of the hex value *#AARRGGBB*
				
				If **plan_touch_mode** is set **plan_control_file_name** is Mandatory
				
				background_file_name and **plan_control_filename** should be the filename of the image including the tile type, eg “background_image.png”
				
				plan_colour_codes are the list of colours on the control image which link to actions.
				
				plan_hightlight_colour is an AndroidHexString in the format *#AARRGGBB*

PlanColourCode
information for each colour section in the plan_control_file_name image
	M
	
	M
	O
	O
	O
	O
		public String colour_code_hex;
		//button click data
		public ButtonType type = ButtonType.*NON_CLICKABLE*;
		public CommandData command_data;
		public ArrayList<MultipleButtonData> sub_buttons;
		public SubButtonPopupData sub_button_layout;
		public String destination_controller_id;
			{
			"colour_code_hex": "#ffaec9",
			"type": "SUB_BUTTON",
			"sub_button_layout": {},
			"sub_buttons": []
			}
				colour_code_hex is an AndroidHexString in the format *#AARRGGBB*
				Depending on the value stored in ButtonType the required data changes, these are also the only supported types for this mode:
				• **VIEW_TRANSACTION** – destination_controller_id required.
				• **SUB_BUTTON**, ***SUB_BUTTON_DIMMER***– sub_button_layout and sub_buttons required, either 2, 3, or 4 entries.
				• **COMMAND** – *command_data required.*
				• ***DISMISS_VIEW*** *– all other fields optional*

GridButtonData
Information for each button on screen
	O
	O
	M
	O
	M
	O
	O
	O
	O
	O
	O
		public CommandData command_data;
		public String label;
		public GridButtonProperties properties;
		public ArrayList<State> states;
		public ButtonType type;
		public CommandData init_command_data;
		public ArrayList<MultipleButtonData> sub_buttons;
		public SubButtonPopupData sub_button_layout;
		public String destination_controller_id;
		public Integer slider_max;
		public Integer slider_min;
			{
			"command_data": {},
			"label": "Scene 1",
			"properties": {},
			"states": [],
			"type": "STATE_BUTTON"
			}
				Depending on the value stored in ButtonType the required data changes:
				• **SLIDER** – slider_min and slider_max are mandatory
				• **DIMMER_COMMAND** or **IMAGE_DIMMER_COMMAND -**
				○ init_command_data should be used for dimming buttons as it will send an optional command first and then every 10 repeats of the dimming command. This is useful for creating dim light up and down to send STEP_DOWN_AND_OFF or STEP_UP_AND_ON.
				• **VIEW_TRANSACTION** – destination_controller_id required.
				• **SUB_BUTTON** – sub_button_layout and sub_buttons required, either 2, 3, or 4 entries.
				• **STATE_BUTTON, STATE_BUTTON_TOGGLE** or **STATE_VIEW** states required, at least 1 entry.

GridButtonProperties
Button placement information
	M
	M
	M
	M
	M
		public Integer column_size;
		public Integer column_start;
		public Integer row_size;
		public Integer row_start;
		public ButtonViewParameters view_parameters;
			{
			"column_size": "1",
			"column_start": "1",
			"row_size": "1",
			"row_start": "4",
			"view_parameters": {}
			}
				column_start and row_start define the starting position of the button in the grid defined in **GridControllerDataVariant row_count** and **column_count** where the top left corner is (0,0).
				column_size and row_size are the span in each dimension. Ensure that the start and size element in both dimensions don’t exceed the counts defined in **GridControllerDataVariant**.

ButtonViewParameters
Button styling information
	O
	C.1
	C.1
	C.2
	C.2
	O
	O
	C.3
	C.3
	C.4
	C.4
	O
	M
	O
	C.5
	C.5
	C.6
	C.6
		public Float alpha;
		public int button_colour;
		public String button_colour_hex;
		public int button_pressed_colour;
		public String button_pressed_colour_hex;
		public String image_path;
		public boolean skeleton_mode;
		public int text_colour;
		public String text_colour_hex;
		public int text_pressed_colour;
		public String text_pressed_colour_hex;
		public float text_size;
		public boolean tint_image;
		public Integer text_gravity;
		public int button_state_unknown_colour = -12471908;
		public int button_state_unknown_colour_hex;
			{
			"alpha": 1,
			"button_colour": -12009044,
			"button_pressed_colour": -14272955,
			"image_path": "arrow-down.svg",
			"skeleton_mode": false,
			"text_colour": -1,
			"text_pressed_colour": -12009044,
			"text_size": 18,
			"tint_image": true
			}
				All colour values (**button_colour**, **button_pressed_colour**, **text_colour**, **text_pressed_colour**, **button_state_unknown_colour**) are a signed integer representation of the hex value *#AARRGGBB*
				All colour hex values (**button_colour_hex**, **button_pressed_colour_hex**, **text_colour_hex**, **text_pressed_colour_hex**, **button_state_unknown_colour_hex**) are a String with the format “#AARRGGBB”. *Colour hex values will override the corresponding colour value.*
				alpha is a decimal value between 0 and 1 where 0 is invisible and 1 is visible.
				image_path should be the file name stored in the assets folder.
				gravity see android reference https://developer.android.com/reference/android/view/Gravity
				button_state_unknown_colour is the default colour for the state indicator
				tint_image if set image from **image_path** will match **text_colour** and **text_pressed_colour**

CommandData
Third Party Interface (TPI) data.
	M
	M
	M
	M
	M
	M
	M
	M
		public String ip;
		public byte control;
		public byte data_0;
		public byte data_1;
		public byte data_2;
		public byte address;
		public byte command;
		public int port;
			{
			"address": 129,
			"command": "17",
			"control": 3,
			"data_0": 0,
			"data_1": 0,
			"data_2": 0,
			"ip": "192.168.0.200",
			"port": 5108
			},
				See the TPI document:
				https://support.zencontrol.com/hc/en-us/articles/360000337175

MultipleButtonData
Data for buttons in a popup
	M
	O
	M
	O
	M
		public CommandData command_data;
		public CommandData init_command_data;
		public String label;
		public AltCommandMode alt_mode = AltCommandMode.*BOTH*;
		public ButtonViewParameters view_parameters;
			{
			"command_data": {},
			"label": "Max",
			"view_parameters": {}
			}
				command_data command to be sent.
				init_command_data alternate command sent based on the **alt_mode** setting:
				• **BOTH - init_command_data** will run at start and end of hold action.
				• **START - init_command_data** will run only at start of hold action.
				• **END - init_command_data** will run only at end of hold action.

SubButtonPopupData
styling information for popup window
	M
	M
	M
	M
	M
	M
	M
		public String label;
		public int label_text_colour;
		public float label_text_size;
		public int popup_width;
		public int popup_height;
		public int start_x;
		public int start_y;
			{
			"label": "Zone 3",
			"label_text_colour": -14272955,
			"label_text_size": 16.0,
			"popup_height": 80,
			"popup_width": 480,
			"start_x": 0,
			"start_y": 800
			},
				label_text_colour is a signed integer representation of the hex value *#AARRGGBB*
				start_x and **start_y** are the row and column start positions of the popup in pixels.
				popup_width and **popup_height** define the size of the popup in pixels.

State
state for STATE_BUTTON, STATE_BUTTON_TOGGLE or STATE_VIEW ButtonType
	M
	M
	M
		public Action action;
		public Condition condition;
		public String event_label;
			{
			"action": {},
			"condition": {},
			"event_label": "button5_variable"
			}	

Condition
condition to match State to Event
	M
	M
		public Answer answer;
		public ConditionType type;
			{
			"answer": {},
			"type": "EQUALS"
			},	

Answer
Expected answer from TPI
	M
	M
		private final byte answer_byte;
		private final AnswerType type;
			{
			"answer_byte": "0",
			"type": "ANSWER"
			}
				See TPI document for expected answers.

Action
Action to perform when Condition matches
	C.1
	C.1
	M
		public int status_colour;
		public String status_colour_hex;
		public ActionType type;
			{
			"status_colour": -14272955,
			"type": "VIEW"
			},
				Colour hex values will override the corresponding colour value.
				status_colour is a signed integer representation of the hex value *#AARRGGBB*
				status_colour_hex is a String with the format ”#AARRGGBB“.

Event
Events to query controller for.
TPI mode 3 only.
	M
	M
	M
		public CommandData command_data;
		public Integer interval_ms;
		public String label;
			{
			"command_data": {},
			"interval_ms": "5000",
			"label": "button0_variable"
			}
				command_data should always have the control byte set to 3 (Quick Query Commands), see TPI document for supported commands.

2 ENUMS

Enum name	Options
ActionType	*VIEW*
AnswerType	*OK*, *ANSWER*, *NO_ANSWER*, *ERROR*, *UNRECOGNIZED*,
ButtonType	*COMMAND*(0), *SUB_BUTTON*(1), *VIEW_TRANSACTION*(2), *NON_CLICKABLE*(3), *IMAGE*(4), *DISMISS_VIEW*(5), *SUB_BUTTON_DIMMER*(6), *SLIDER*(7), *STATE_BUTTON*(8), *STATE_VIEW*(9), *IMAGE_COMMAND*(10), *DIMMER_COMMAND*(11), *IMAGE_DIMMER_COMMAND*(12), *COLOUR_TC_COMMAND*(13), *COLOUR_XY_COMMAND*(14), *COLOUR_RGBWAF_COMMAND*(15), *SUB_BUTTON_COLOUR_TC*(16), *SUB_BUTTON_COLOUR_XY*(17), *SUB_BUTTON_COLOUR_RGBWAF*(18), *COLOUR_RGB_COMMAND*(19), *SUB_BUTTON_COLOUR_RGB*(20), *STATE_BUTTON_TOGGLE*(21),
ConditionType	*EQUALS*(0), *NOT_EQUALS*(1)