

Vuex Store to Backend Interfaces and Data Transfer API

Use Case Requirement	Front-End (Client)	API (Routes-Interface)	DataStore-Interface
<div>Theme: Primary Color Requirement:<ul style="list-style-type: none">Allow User SelectionStore Current and New Color ChoiceUpdate all affected elements in all components</div>	<div>Element: Input Type: Color Data Members: Class: ColorSelector ID: dh-theme-primary NewColor: hexColor PreviousColor: hexColor</div>	<div>Params: NewColor Route: prefix: /api/theming route: /colors/primary</div>	<div>Method: Write Function: AddPrimColor DataIn: hexColor(string) DataStruct: Array File: Ira-themes.json Path: themes.colors.primary</div>
<div>Display: Swatch Blocks of Colors Previously Chose Requirement:<ul style="list-style-type: none">Swatches should be 'clickable'Update all affected elements in all components upon click</div>	<div>Element: Column Type: Div Square of color Data: PreviousHexColors Type: Array Event: Click SendToBackend: Index HexColor</div>	<div>Params: HexValue Route: prefix: /api/theming route: /colors/primary</div>	<div>Method: Update Function: UpdatePrimColors DataIn: hexColor(string) DataStruct: Array File: Ira-themes.json Path: themes.colors.primary Functionality: Shift currColor to index 1 Shift inputColor to index 0 Search & remove duplicates</div>