



Updated: 2024/10/17

Doofenshmirtz Data Transmogrifier API Documentation

COS214 Group Project



*Doofenshmirtz evil
incorporated*

FAQ:

What does this API do

The API is used to convert data provided by the C++ backend into visual changes to our HTML and CSS frontend.

What is the API written in

The API is written in JavaScript.

What libraries are used

The nlohmann JSON library is used to convert C++ JSON-like objects into string before being passed to the API via websockets. Furthermore, the boost beast library is used for creating and maintaining websockets.

What are the HTML element IDs that can be updated via the valueUpdate endpoint:

Any of the following may be used as the id parameter (note: case is NOT important):

money, citizens, satisfaction, residential, commercial, industrial, landmarks, wood, steel, concrete, bricks, power, water, waste, sewage, progress-public, progress-car, progress-train, progress-plane

How to use this API

Necessary Installations:

Using the WSL terminal install the following libraries:

- Boostbeast: `sudo apt-get install libboost-all-dev`
- Nlohmann JSON: `sudo apt-get install nlohmann-json3-dev`

Making Requests:

The “request” object should be created in the C++ backend using the nlohmann pseudo JSON object library. For example:

```
json UpdateRequest = {  
    {"type", "valueUpdate"},  
    {"data", {  
        {"id", "money"},  
        {"value", "30"}  
    }}  
};
```

The request object should then be passed to the websocket by calling the `dump()` method on it.

```
ws.write(net::buffer(UpdateRequest.dump()));
```

Note: In many instances the nlohmann JSON object type will simply be *json* this is because of:

```
using json = nlohmann::json;
```

API Endpoints:

valueUpdate:

Parameter Name	Required	Type	Description
type	*	String	Endpoint Name: valueUpdate
data	*	Nlohmann JSON i.e. {}	A key value pair of containing the id of the element to update and the new value.
id	*	String	The id of the HTML element to update
Value	*	String	The new value of the element: Either: A plain number i.e. "30" An increment i.e. "10++" A decrement i.e. "4--"

Example Request:

```
json UpdateRequest = {  
    {"type", "valueUpdate"},  
    {"data", {  
        {"id", "money"},  
        {"value", "30"}  
    }}  
};
```

News

Parameter Name	Required	Type	Description
type	*	String	Endpoint Name: news
data	*	String	The message that should be displayed.

Example Request:

```
json newsRequest = {  
  {"type", "news"},  
  {"data", "Testing out the news API endpoint. If you can see this it probably worked."}  
};
```

cityVisuals

type	*	String	Endpoint Name: cityVisuals
data	*	Nlohmann JSON	<p>The JSON object result of calling <i>getJSONrepresentation()</i> on the city composite structure.</p> <p>Note: This method is defined in the composite structure to assemble a JSON object representative of the city.</p>

Example Request:

```
json cityVisualsRequest = {  
    {"type", "cityVisuals"},  
    {"data", city->getJSONrepresentation()}  
};
```