ADP Gateway API Documentation

John Machado

February 2024

Contents

1 Introduction

A REST API has been implemented to facilitate automated access to the sign's database. REST APIs are widely supported in all programming languages. For more information visit POSTMAN's REST API primer.

REST APIs facilitate sign automation.

They are analogous to user interfaces for humans.

2 Update Screens

A screen is a collection of rows, which in turn are a collection of segments.

- Screen creation/update database operations are inferred; this is the only endpoint needed.
- All screens added to the sign will become part of a playlist.

2.1 Endpoint

.../screens/update

2.2 Method

GET

2.3 Payload


```
{
  "playlist": [],
  "screens": [
    {
      "screen": {
        "id": "",
        "vertical_alignment": "",
        "rows": [
          {
            "row": {
              "font_size": 0,
              "font_weight": "",
              "hold_time": 0,
              "horizontal_alignment": "",
              "in_mode": "",
              "scroll_speed": "",
              "segments": [
                {
                  "segment": {
                    "foreground_color": "",
                    "background_color": "",
                    "flash": false,
                    "text": ""
                  }
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

2.3.1 Valid Values

- playlist: array where elements must be valid screen ids
- screen.id: four character length, can be any ascii character in range 32,125 inclusive
- screen.vertical_alignment: left, center, right
- row.font_size: font size in pixels, accepts all values but only certain sizes are loaded on the sign
- row.font_weight: bold, normal
- row.hold_time: range 1, 99 inclusive
- row.horizontal_alignment: top, middle, fill, bottom
- row.in_mode: hold, scroll
- row.scroll_speed: slowest, slow, normal, fast, fastest
- segment.foreground_color: black, red, green, blue, yellow, white, or 6 digit hex value preceded by #
- segment.background_color: black, red, green, blue, yellow, white, or 6 digit hex value preceded by #
- flash: true, false
- text: ascii characters range 32, 125

2.4 Returned Data

Returned data taken from database and provided for user validation.


```
{
  "playlist": [],
  "screens": [
    {
      "screen": {
        "id": "",
        "vertical_alignment": "",
        "rows": [
          {
            "row": {
              "font_size": 0,
              "font_weight": "",
              "hold_time": 0,
              "horizontal_alignment": "",
              "in_mode": "",
              "scroll_speed": "",
              "segments": [
                {
                  "segment": {
                    "foreground_color": "",
                    "background_color": "",
                    "flash": false,
                    "text": ""
                  }
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

3 Update Playlist

The playlist is the list of screens that will be shown sequentially. The interval between screens is determined by the HOLD TIME option and any overflow condition.

- Updating the playlist will replace the old playlist with the new one.
- Removing screen ids from the playlist will not remove the screen data from the database.

3.1 Endpoint

.../playlist/update

3.2 Method

GET

3.3 Payload

```
{
  "playlist": []
}
```

3.3.1 Valid Values

- playlist: array where elements must be valid screen ids

3.4 Returned Data

Returned data taken from database and provided for user validation.

```
{
  "playlist": []
}
```

4 Read Screens from Sign

Returns all playlist and screen data.

4.1 Endpoint

.../screens/read

4.2 Method

GET

4.3 Payload

NOT REQUIRED

4.3.1 Valid Values

NOT APPLICABLE

4.4 Returned Data


```
{
  "playlist": [],
  "screens": [
    {
      "screen": {
        "id": "",
        "vertical_alignment": "",
        "rows": [
          {
            "row": {
              "font_size": 0,
              "font_weight": "",
              "hold_time": 0,
              "horizontal_alignment": "",
              "in_mode": "",
              "scroll_speed": "",
              "segments": [
                {
                  "segment": {
                    "foreground_color": "",
                    "background_color": "",
                    "flash": false,
                    "text": ""
                  }
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

5 Remove Screens from Sign

Delete screens from the signs database and playlist

5.1 Endpoint

.../playlist/delete

5.2 Method

GET

5.3 Payload

The screen ids of screens that should be removed from the signs database.

```
{
  "screen_ids_to_delete": []
}
```

5.3.1 Valid Values

- screen_ids_to_delete: array where elements must be valid screen ids

5.4 Returned Data

Post operation screen ids of screens saved in the database and playlist.

```
{
  "saved_screen_ids": [],
  "playlist_screen_ids": []
}
```

6 Remove All Screens from Sign

Removes all screens from both the database and the playlist.

6.1 Endpoint

.../playlist/delete_all

6.2 Method

GET

6.3 Payload

NOT REQUIRED

6.3.1 Valid Values

NOT APPLICABLE

6.4 Returned Data

Returned data taken from database and provided for user validation.

```
{
  "saved_screen_ids": [],
  "playlist_screen_ids": []
}
```