

ADP Gateway API Documentation

John Machado

March 2024

Contents

1	Introduction	4
2	Read Screens	5
2.1	Endpoint	5
2.2	Method	5
2.3	Payload	5
2.3.1	Valid Values	5
2.4	Returned Data	6
3	Update Screens	7
3.1	Endpoint	7
3.2	Method	7
3.3	Payload	8
3.3.1	Valid Values	9
3.4	Returned Data	10
4	Remove Screens	11
4.1	Endpoint	11
4.2	Method	11
4.3	Payload	11
4.3.1	Valid Values	11
4.4	Returned Data	11
5	Remove All Screens	12
5.1	Endpoint	12
5.2	Method	12
5.3	Payload	12
5.3.1	Valid Values	12
5.4	Returned Data	12
6	Read Playlist	13
6.1	Endpoint	13
6.2	Method	13
6.3	Payload	13
6.3.1	Valid Values	13
6.4	Returned Data	14
7	Update Playlist	15
7.1	Endpoint	15
7.2	Method	15
7.3	Payload	15
7.3.1	Valid Values	15
7.4	Returned Data	15

8	Remove from Playlist	16
8.1	Endpoint	16
8.2	Method	16
8.3	Payload	16
8.3.1	Valid Values	16
8.4	Returned Data	16
9	Remove All from Playlist	17
9.1	Endpoint	17
9.2	Method	17
9.3	Payload	17
9.3.1	Valid Values	17
9.4	Returned Data	17

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 Read Screens

Returns all playlist and screen data.

2.1 Endpoint

`.../screens/read`

2.2 Method

GET

2.3 Payload

NOT REQUIRED

2.3.1 Valid Values

NOT APPLICABLE

2.4 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": ""
                  }
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

3 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.
- The playlist will become the list of screens added by this operation.

3.1 Endpoint

`.../screens/update`

3.2 Method

`GET`

3.3 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": ""
                  }
                }
              ]
            }
          ]
        ]
      }
    }
  ]
}
```


3.3.1 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

3.4 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": ""
                  }
                }
              ]
            }
          ]
        ]
      }
    }
  ]
}
```

4 Remove Screens

Delete screens from the signs database and playlist

4.1 Endpoint

.../screens/delete

4.2 Method

GET

4.3 Payload

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

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

4.3.1 Valid Values

- screen_ids_to_delete: array where elements must be valid screen ids

4.4 Returned Data

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

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

5 Remove All Screens

Removes all screens from both the database and the playlist.

5.1 Endpoint

`.../screens/delete_all`

5.2 Method

GET

5.3 Payload

NOT REQUIRED

5.3.1 Valid Values

NOT APPLICABLE

5.4 Returned Data

Returned data taken from database and provided for user validation.

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

6 Read Playlist

Returns all playlist data.

6.1 Endpoint

.../playlist/read

6.2 Method

GET

6.3 Payload

NOT REQUIRED

6.3.1 Valid Values

NOT APPLICABLE

6.4 Returned Data

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

7 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.

7.1 Endpoint

`.../playlist/update`

7.2 Method

`GET`

7.3 Payload

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

7.3.1 Valid Values

- playlist: array where elements must be valid screen ids

7.4 Returned Data

Returned data taken from database and provided for user validation.

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

8 Remove from Playlist

Removes entries from the playlist.

8.1 Endpoint

`.../playlist/delete`

8.2 Method

GET

8.3 Payload

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

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

8.3.1 Valid Values

- `screen_ids_to_delete`: array where elements must be valid screen ids

8.4 Returned Data

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

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


9 Remove All from Playlist

Removes all playlist entries.

9.1 Endpoint

.../playlist/delete_all

9.2 Method

GET

9.3 Payload

NOT REQUIRED

9.3.1 Valid Values

NOT APPLICABLE

9.4 Returned Data

Returned data taken from database and provided for user validation.

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