WI DSPS License API Documentation

API Overview

The DSPS API enables authorized third parties to securely retrieve and verify license information in real-time. This guide provides step-by-step instructions on how to connect to and utilize the API effectively.

Introduction to the DSPS API

The DSPS API is a critical tool for organizations and individuals requiring real-time access to licensing information. This API eliminates the need for manual processes by providing an automated, secure, and efficient way to verify credentials. With easy integration into existing systems, the DSPS API supports industries such as healthcare, education, and government compliance.

Key Benefits of the API

- *Efficiency:* Automate license verifications to save time and reduce errors.
- *Accuracy*: Access the most up-to-date licensing information directly from the source.
- *Security*: Maintain confidentiality with secure API authentication protocols.
- *Scalability*: Support large-scale operations with real-time, on-demand data retrieval.
- *Compliance Support*: Meet regulatory requirements with verified and comprehensive licensing details.

Steps to Connect

To make API requests, you will need the base URL, request method, and the required parameters. The following sections explain each component in detail.

Base URL

The base URL serves as the entry point for accessing the API. All API requests should begin with the following endpoint:

https://prod-exp-wi-license–search-v1.us-e2.cloudhub.io/api/blp/{LicenseNumber}

This endpoint must be appended with the appropriate license number to retrieve specific license details. Replace {LicenseNumber} with the relevant value to construct the full URL.

HTTP Method

The API uses the **GET** method. This method retrieves data from the server without modifying any information. It's a straightforward and reliable way to query license data.

Required Parameters

To successfully make a request, you must include the following parameter:

- *LicenseNumber:* The identifier for the license you want to query. The format should follow this structure: <License Number> <space> - <space> <Credential Type Id>

Example: 1000 – 21

Headers

Authentication is required to access the API. Ensure that your request includes the following headers:

- **client_id**: Your assigned API client ID.
- **client_secret**: Your assigned API client secret.

Sample Request

This example demonstrates how to construct and send an API request.

Example Request:

GET https://prod-exp-wi-license–search-v1.us-e2.cloudhub.io/api/blp/1000 - 21

Headers:

```
 client_id: your-client-id
 client_secret: your-client-secret
```

This request retrieves information for the license "1000 - 21," assuming your credentials are valid.

Response Details

The API provides detailed responses that include license information or indicate issues. Understanding these responses is crucial for effective integration.

Successful Response (200 OK)

A successful request returns a JSON object with license details. Below is an example of such a response:

Example Response:

```
{
  "Name": "John Doe",
  "Status": "Active",
  "License Name": "1000 - 21",
  "Period End": "2025-12-31",
  "License Number": "1000 - 21",
  "Jurisdiction Type": "STATE",
  "License Type": "Regular",
  "Multi-State Indicator": "true",
  "Regulatory Authorization Type": "Medical License",
  "Issuing Authority": "DSPS",
  "Multi-State": "Illinois;Wisconsin",
  "Disciplinary Action Indicator": "false",
  "Compact Indicator": "true",
  "Granted Date": "2022-06-15",
  "Application Method": "Licensure by Examination",
  "Speciality": "true",
  "Specialities Selected": "General Medicine",
  "Sub - Speciality Selected": "Pediatrics",
  "Organization Name": "HealthCare Inc."
}
```

No Data Found (204 No Content)

If the specified license number does not exist, the API returns the following

Response Example:

```
{
  "statusCode": 204,
  "message": "Data Not Found for provided LicenseNumber",
  "appName": " prod-exp-wi-license-search-v1",
  "uuid": "710bde5e-6600-4033-8f92-420d8738e986"
}
```

Error Response (401 Unauthorized)

Unauthorized access or invalid credentials result in the following response:

Response Example:

```
{
  "statusCode": 401,
  "errorType": "Unauthorized",
  "message": "Invalid client_id or client_secret",
  "appName": "prod-exp-wi-license-search-v1",
  "uuid": "cd1bd12d-2aa1-4021-af0c-211652f52f7d"
 }
```

Contact Information

If you have any questions or need assistance during the setup, our technical support team is here to help. Please contact us to obtain your client ID and client secret to connect to the API.

- **Email:** Naveen.aindla@wisconsin.gov, randalj.norton@wisconsin.gov
- **Phone:** 608.261.5050, 608 261-5891
- **Support Hours:** Monday to Friday, 8 AM - 5 PM (CST)

We are committed to ensuring a seamless integration experience. Reach out to us anytime for further assistance.

Implementation Examples

The following example demonstrates how to implement the API in various programming languages. These examples are intended for technical teams but are also included to illustrate the flexibility and ease of API integration.

1. Using C#

```
using System;
using System.Net.Http;
using System.Threading.Tasks;

class Program
{
    static async Task Main(string[] args)
    {
        string baseUrl = "https://prod-exp-wi-license–search-v1.us-e2.cloudhub.io/api/blp/";
        string licenseNumber = "1000 - 21"; // Replace with your license number
        string clientId = "your-client-id"; // Replace with your client ID
        string clientSecret = "your-client-secret"; // Replace with your client secret

        using (HttpClient client = new HttpClient())
        {
            client.DefaultRequestHeaders.Add("client_id", clientId);
            client.DefaultRequestHeaders.Add("client_secret", clientSecret);

            HttpResponseMessage response = await client.GetAsync(baseUrl + Uri.EscapeUriString(licenseNumber));
            if (response.IsSuccessStatusCode)
            {
                string responseBody = await response.Content.ReadAsStringAsync();
                Console.WriteLine("Response: " + responseBody);
            }
            else
            {
                Console.WriteLine($"Error: {response.StatusCode} - {response.ReasonPhrase}");
            }
        }
    }
}
```