## 

## **Property Source External API Documentation**

This comprehensive documentation provides detailed information about all the endpoints available in the Property Source External API, including instructions on retrieving user tokens, setting them in cookies, and transferring users to the Property Source platform.

**API DOCS URL :** [**https://api.propertysource.w3saas.com/property-source/ui**](https://api.propertysource.w3saas.com/property-source/ui)

### **1. API Overview**

The Property Source External API allows you to manage users and their access to the Property Source platform. It provides the following functionalities:

* Get User Token: Retrieves a JWT (JSON Web Token) for a user, enabling secure authentication and authorization.
* Create Issuer: Promotes a user to an admin (issuer) within the Property Source platform, granting them special privileges.
* Update Issuer: Updates details of an existing issuer (admin user), such as their contact information or status.
* Delete Issuer: Removes issuer privileges or deletes the issuer record entirely, revoking their admin access.

### **2. Authentication**

This API uses API Key authentication for all endpoints, except for login, which utilizes Basic Authentication.

API Key Authentication:

* Obtaining an API Key:
  + Log in to the Property Source application using your provided username and password (e.g., username: **admin@blockchainlaboratories.com**, password: **Password2024@**).
  + Navigate to the "API Management" section, typically found in the navigation bar.
  + Follow the instructions to generate a new API key. You'll also find API documentation and other relevant information in this section.
* Using an API Key: Include the generated API key in the X-API-Key header for each request to the API endpoints (except for /login).

Login/Basic Authentication:

* /login Endpoint: For the /login endpoint, use Basic Authentication with the following placeholder credentials:
  + Username: **admin@blockchainlaboratories.com**
  + Password: **Password2024@**

### **3. API Endpoints**

#### **3.1 Get User Token**

Endpoint: /get\_user\_token

Method: POST

Summary: Retrieves a JWT for a user, allowing them to access the Property Source platform. If the user doesn't exist, it registers them first.

Request Body:

{

"userId": "user123",

"email": "user@example.com",

"userName": "John Doe"

}

Response (200 OK):

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

Error Responses:

* 400 Bad Request: Invalid parameters supplied.
* 405 Method Not Allowed: Validation exception.

#### **3.2 Create Issuer**

Endpoint: /create\_issuer

Method: POST

Summary: Promotes an existing user to an admin (issuer) within the Property Source platform.

Request Body:

{

"userId": "user123"

}

Response (200 OK):

{

"email": "user@example.com",

"userId": "user123",

"createdAt": 1678886400000,

"updatedAt": 1678886400000,

"isIssuer": true,

"numberOfOfferings": 2

}

Error Responses:

* 400 Bad Request: Invalid parameters supplied.
* 405 Method Not Allowed: Validation exception.

#### **3.3 Update Issuer**

Endpoint: /update\_issuer

Method: POST

Summary: Updates the details of an existing issuer (admin user).

Request Body:

{

"email": "updated\_user@example.com",

"userId": "user123",

"createdAt": 1678886400000,

"updatedAt": 1678886400000,

"isIssuer": true,

"numberOfOfferings": 3

}

Response (200 OK):

{

"message": "Issuer updated successfully",

"user": {

"email": "updated\_user@example.com",

"userId": "user123",

// ... other user details

}

}

Error Responses:

* 400 Bad Request: Invalid parameters supplied.
* 405 Method Not Allowed: Validation exception.

#### **3.4 Delete Issuer**

Endpoint: /delete\_issuer

Method: POST

Summary: Removes issuer privileges or deletes the issuer record entirely.

Request Body:

{

"userId": "user123"

}

Response (200 OK):

{

"message": "Issuer marked as deleted successfully",

"userId": "user123"

}

Error Responses:

* 400 Bad Request: Invalid parameters supplied.
* 404 Not Found: Issuer not found.
* 405 Method Not Allowed: Validation exception.

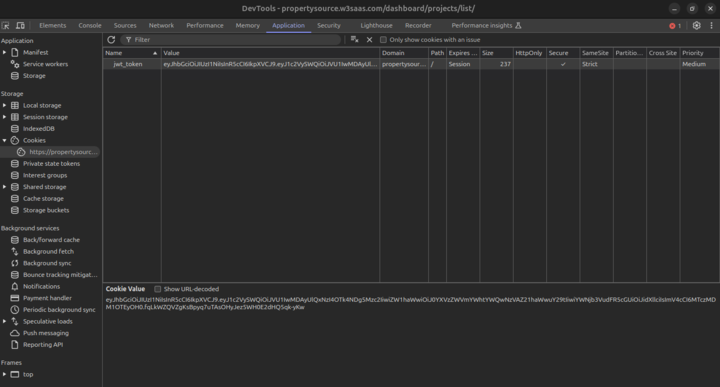
### **4. Setting the JWT in a Cookie**

After successfully retrieving a JWT from the /get\_user\_token endpoint, store it securely in a cookie to enable seamless user transfer to the Property Source platform:

document.cookie = `jwt\_token=${token}; path=/; domain=”propertysource.w3saas.com”; secure; HttpOnly`;

Additional Information:

Add jwt\_token in cookies and domain as “propertysource.w3saas.com”



Explanation:

* jwt\_token: The name of the cookie.
* ${token}: The JWT received from the API.
* path=/: The cookie is accessible from the entire domain.
* secure: The cookie is only sent over HTTPS.
* HttpOnly: The cookie is not accessible from JavaScript, enhancing security.
* Domain: “propertysource.w3saas.com”

### **5. Transferring the User**

Once the JWT cookie is set, redirect the user to the Property Source platform:

window.location.href = "https://propertysource.w3saas.com";

### **6. Important Notes**

* Security: Handle the JWT with care as it contains sensitive user information. Ensure that your application follows secure coding practices to prevent token theft or misuse.
* Cookie Settings: Adjust the cookie settings (path, domain, expiry) based on your specific needs and security requirements.
* Error Handling: Implement proper error handling in your application to gracefully manage API errors and inform the user accordingly.
* API Key Management: Store your API keys securely and follow best practices for API key management.

This comprehensive documentation provides detailed information about all the endpoints in the Property Source External API. It should enable you to successfully integrate with the API and implement user management and transfer functionalities in your application. If you have any further questions, please contact Property Source support.