Authenticating Sendbird Clients with JWTs

Date: 2020-08-11
Prepared By: Solutions Engineering

What are JWTs

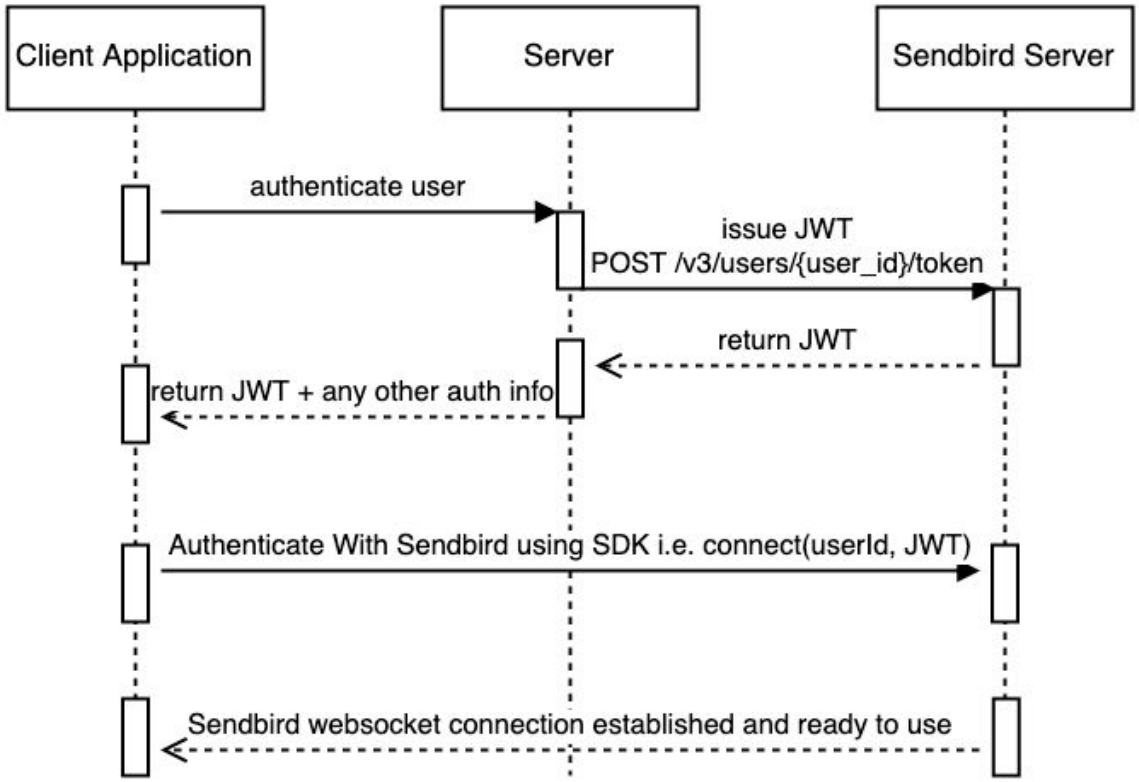
JWTs or JSON Web Tokens are an alternative method of authenticating a Sendbird client. These tokens are another type of stateful session token, but differ from the current Sendbird access and session tokens in that they are not stored in the Sendbird database, but instead the relevant information or claims are part of the token itself. The token is signed with a private secret by the Sendbird server when the token is issued, and then when a Sendbird client uses the token to authenticate, its claims are validated by the Sendbird server. jwt.io explains this in more detail.

JWT Flow

Sendbird tokens can only be retrieved using the platform API, they cannot be retrieved using the Sendbird SDK. The most common flow would be something like this. If you are already using Sendbird with access or session tokens this is likely identical to the flow you are already using.

Issuing a token:

1. Authenticate with your own server from a client application
2. Your server requests a JWT from the Sendbird server
3. Your server returns the JWT to the client application which it can cache if desired
4. Authenticate the client app using the Sendbird SDK and the JWT
5. Your client now has an active session with the Sendbird server



Refreshing a token:
If the token cached by the client is expired authentication will fail. There is not a unique refresh flow and you can simply issue a token if it is expired. The expiration timestamp is included with the token response so you can check this prior to attempting to authenticate.

Differences Between JWTs and Other Sendbird Authentication Methods

- JWTs are not stored in the Sendbird database. This makes it more efficient to issue the token and authenticate a user.
- All existing JWTs can be revoked in bulk with a single API call
- It does not require a user update call to issue a token so this API call has a more focused responsibility
- There is no limit on the number of JWTs that are valid at any one time.

API Endpoints

Issuing a token for a user:

```
POST https://api-{application_id}.sendbird.com/v3/users/{user_id}/token
```

Response:

```
{
```

```
    "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlIjoxNTk4NjUxODE1LCJ1IjoxMjI0NzQ5O
DEsInYiOjF9.CFbZ4iOsSt-1tUGQVjSY001K8_a6KJGZWO2VEeA4Edk",
    "expires_at": 1598651815000
}
```


Properties:

Name	Type	Required	Default Value	Description
expires_at	int	false	Current datetime + 7 days	Expiration time in milliseconds from epoch

Revoking all tokens for a user:

```
DELETE https://api-{application_id}.sendbird.com/v3/users/user_id/token
```

Response:

```
{}
```

Enabling JWT Authentication

For new or existing applications, if you contact us, Sendbird can enable your application to use JWTs for authentication.

Migrating From Legacy Session Tokens

For those already using legacy session tokens to authenticate with Sendbird you can make this transition easier with JWT compatibility mode. This must be approved and enabled by Sendbird and allows you to issue JWTs without any changes to your existing code. This should be considered temporary and does not have some of the performance improvements because tokens are still issued by making a user update request. In compatibility mode a token is still issued by calling:

This is only available in compatibility mode for those in the process of updating their implementation. If JWTs are enabled, but compatibility mode is disabled this call will not issue a new JWT or legacy session token.

PUT https://api-{application_id}.sendbird.com/v3/users/{user_id}

Properties:

Name	Type	Required	Default Value	Description
issue_session_token	boolean	false	false	Issues session token
expires_at	int	false	Current datetime + 7 days	Expiration time in milliseconds from epoch

Response:

```
{
	...
	"session_tokens": [
	{
		"expires_at": 1598652190000,
		"session_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlIjoxNTk4NjUyMTkwLCJ1IjoxMjI0NzQ5ODEsInYiOjF9.WOlKxggerocChFKa1Nw9Uq8sWz7ppn0gPJ2NEiblZzQ"
	}
	],
	...
}
```

When migrating from legacy session tokens there are some concerns to be aware of:

- JWTs are larger than legacy session tokens. They are 128 bytes. If you are caching tokens on the client you will want to ensure that you support caching the larger token rather than having a limit on its length
- For users with existing legacy session tokens, their legacy tokens will remain valid until they expire or that user has been issued a JWT. Once a JWT is issued all existing legacy session tokens will be revoked.