

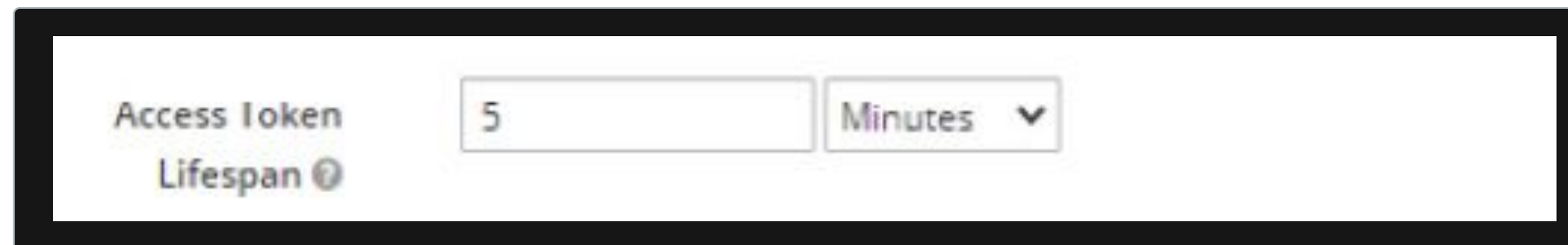
Gestionando tokens

Podemos configurar el ciclo de vida de los tokens de forma similar a las sesiones. Para esto tenemos que seleccionar la **pestaña de “Tokens”** dentro de la página de configuración de nuestro reino. Allí, podemos establecer el tiempo de vida de cada uno de los tres tokens mencionados anteriormente.



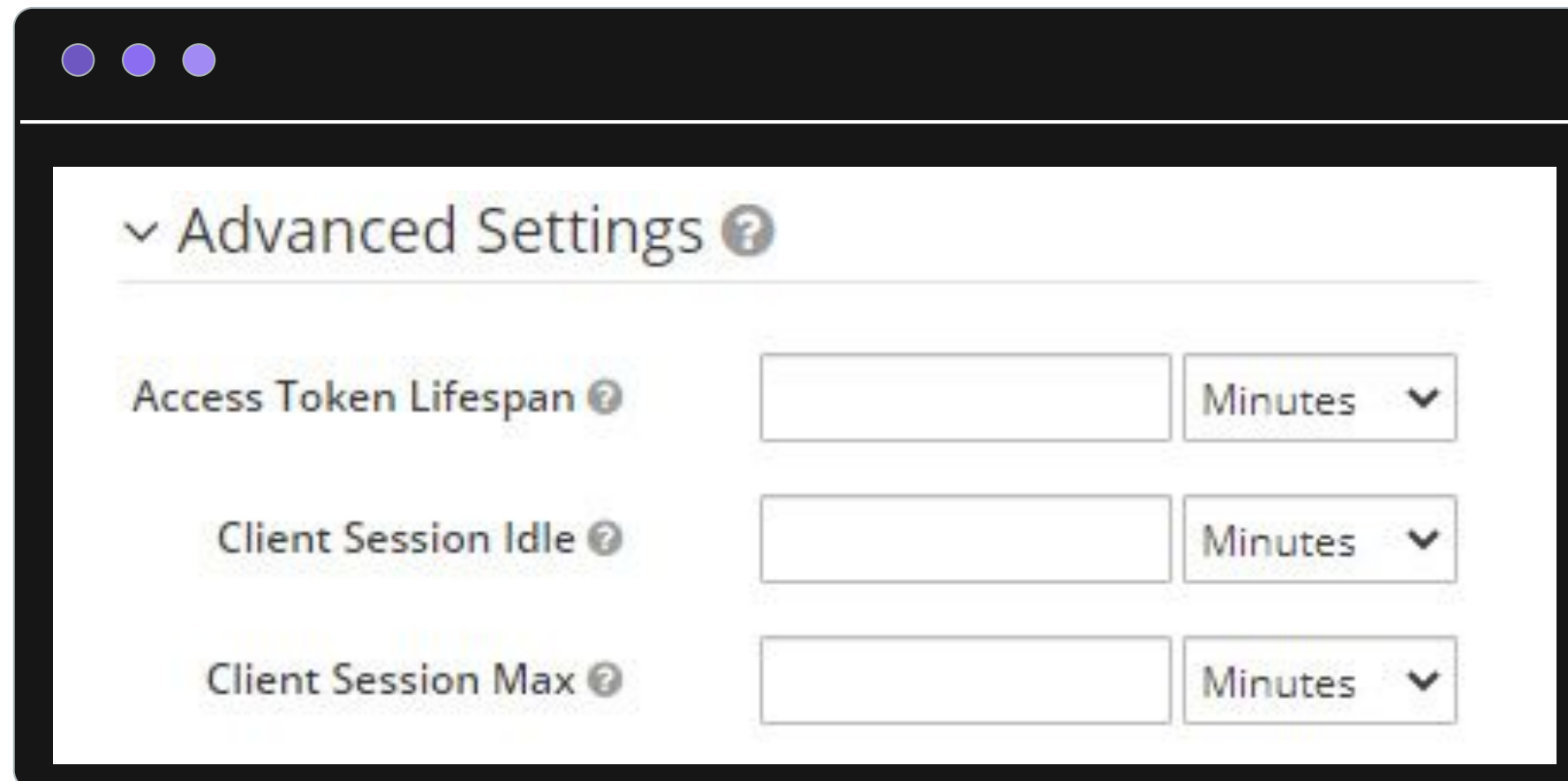
Gestionando el ciclo de vida de ID token y access token

Para **ID token** y **access token** tenemos la opción general:



A screenshot of a configuration field for 'Access Token Lifespan'. It consists of a text input box containing the number '5' and a dropdown menu currently set to 'Minutes'. The entire field is enclosed in a black rectangular border.

Estos tokens también se pueden configurar por cliente dentro de la página de detalle en la pestaña de “**Settings**”, en el apartado “**Advance Settings**”.



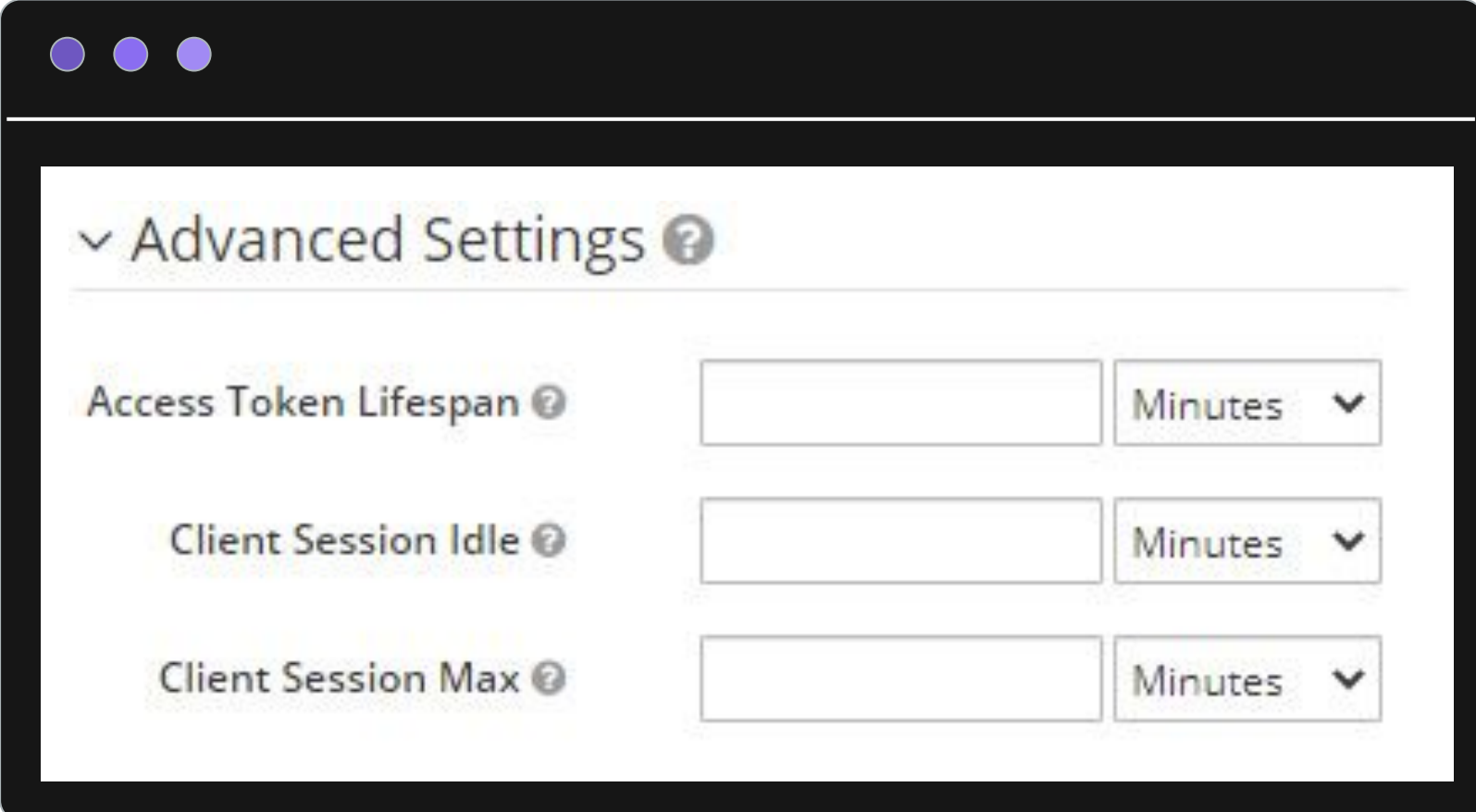
A screenshot of a web application window showing the 'Advanced Settings' section. The section is titled 'Advanced Settings' with a dropdown arrow and a help icon. It contains three rows of configuration fields, each with a label, a text input box, and a dropdown menu set to 'Minutes':
1. 'Access Token Lifespan' with an empty input box.
2. 'Client Session Idle' with an empty input box.
3. 'Client Session Max' with an empty input box.
The window has a dark header with three colored circles (purple, blue, green) on the left.

El tiempo de estos tokens tiene que ser reducido al máximo para evitar vulnerabilidades de aplicación ante la pérdida de dichos tokens, dado que tendremos la posibilidad de actualizarlos por el refresh token.

Gestionando el ciclo de vida de refresh token

Este token debe tener la misma vida útil que el máximo de sesión de SSO y cliente definido. En su defecto, tomará el máximo valor de sesión definido para el cliente. En caso de que este valor tampoco exista, tomará el valor de sesión SSO general.

Para configurar este valor, debemos ir a la página de “**Settings**” de cliente y seleccionar la opción “**Advanced Settings**”, donde debemos actualizar el valor de “**Client Session Idle**” y “**Client Session Max**” en el caso de que no queramos heredar el valor definido en sesión de SSO.



The screenshot shows a web interface with a dark header and a light content area. The content area has a title "Advanced Settings" with a question mark icon. Below the title, there are three rows of configuration options, each with a label, an input field, and a unit dropdown menu.

Setting	Value	Unit
Access Token Lifespan		Minutes
Client Session Idle		Minutes
Client Session Max		Minutes

Antes de actualizar este valor, debemos considerar que los refresh tokens:

- Siempre están vinculados a una sesión de cliente después de autenticar a los usuarios en Keycloak.
- Se consideran válidos si las sesiones de usuario y cliente a las que están vinculados no han expirado.
- Los clientes deberían poder usar tokens de actualización para obtener nuevos tokens solo si sus respectivas sesiones de cliente todavía están activas.

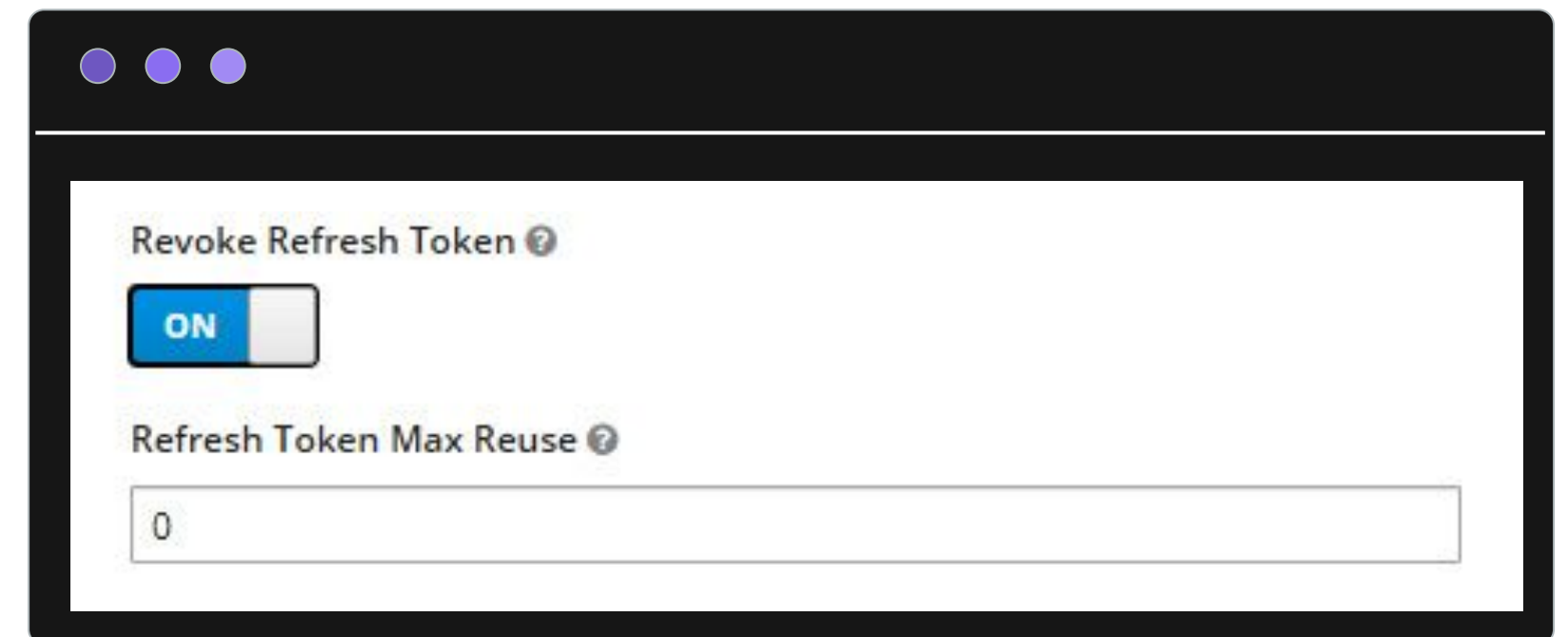
La vida útil de este tipo de token se puede ajustar según la cantidad de clientes que puedan mantener sus tokens seguros. Por ejemplo, un cliente confidencial podría tener tokens de actualización que duren más, mientras que para los clientes públicos buscamos una ventana de tiempo más pequeña. Sin embargo, debemos tener en cuenta que tan pronto como expire un refresh token, los usuarios se verán obligados a volver a autenticarse en el cliente, lo que afectará la experiencia del usuario si utiliza un navegador.

Habilitando la rotación del refresh token

Keycloak nos ofrece la capacidad de utilizar la rotación de **refresh tokens** para reducir el riesgo de seguridad de tener que perder este tipo de tokens.

Es una estrategia que reemplaza el refresh token actual con uno nuevo cada vez que se invoca a la URL de refresh, lo cual obliga a un atacante a quedar expuesto o sin acceso cuando el usuario original actualiza su token.

Para esto debemos activar la característica de “**Refresh Token Rotation**” en la pestaña de “**Tokens**” dentro de la página de configuración del reino en cuestión.



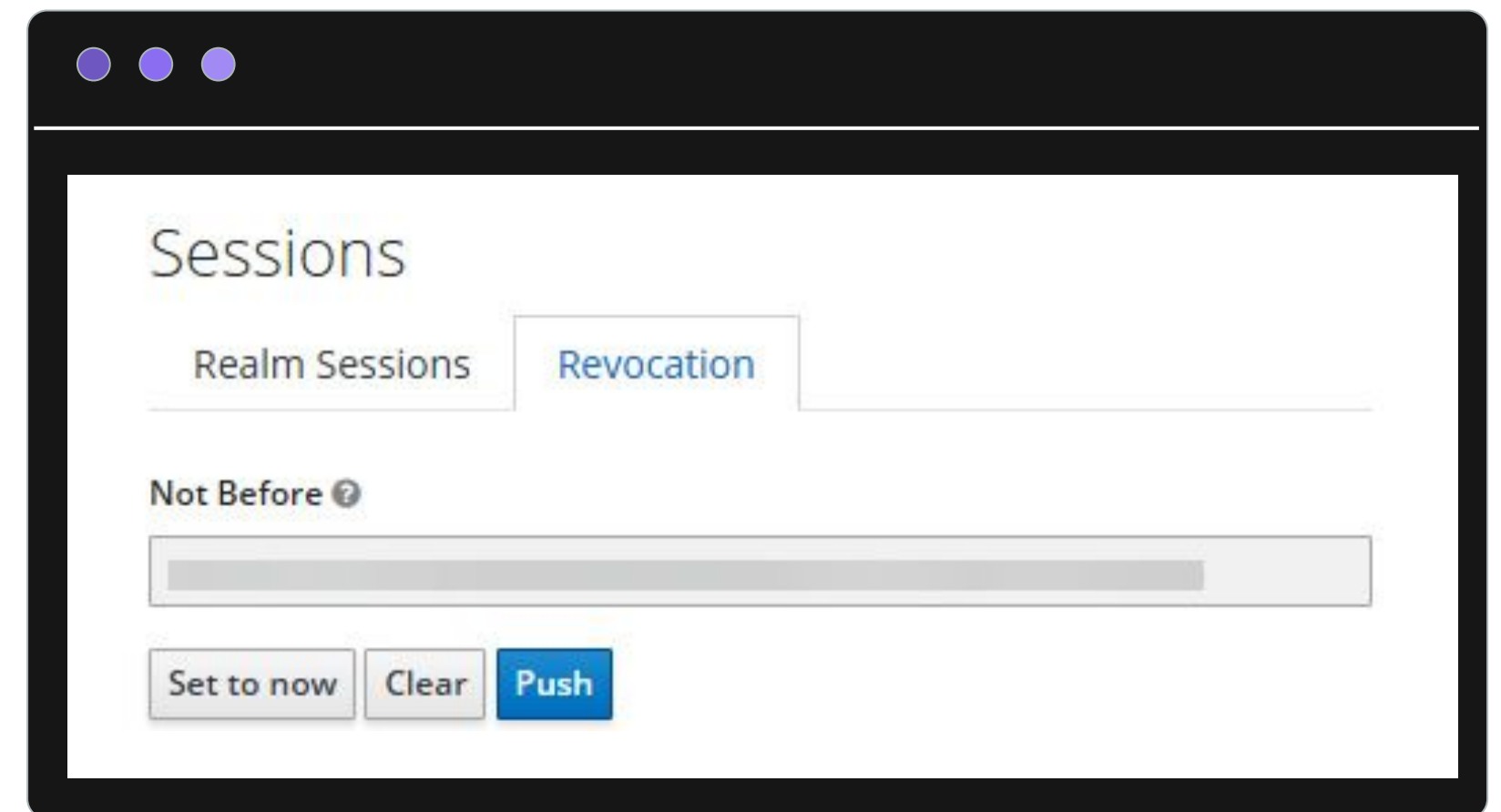
Al habilitar esta característica, tenemos la opción de “**Refresh Token Max Reuse**” que establece la cantidad de llamadas de renovación de token que se puede hacer con el token de refresh. Por defecto, es cero para mayor seguridad.

Revocación de tokens

Keycloak nos permite revocar tokens utilizando diferentes métodos.

Una de las formas más sencillas de invalidar tokens globalmente —independientemente del usuario y el cliente— es usar una política **not-before-revocation** para obligar a los tokens a caducar en función de un parámetro de tiempo, evitando que se anule la sesión de cliente y usuario.

Dentro de la opción de sesiones de Keycloak, en la pestaña de “**Revocation**”, tendremos la opción que podemos observar en la imagen.



Al hacer clic en el botón “**Set to now**”, Keycloak llenará automáticamente el campo “**Not Before**” con la hora actual y actualizará la configuración del reino para fallar la validación del token siempre que se haya creado un token antes del tiempo establecido.

¡Muchas gracias!