

# OwnerLimitation

A limitation to specify that only owner of content gets a certain access right.

<b>Identifier</b>	Owner
<b>Value Class</b>	eZ\Publish\API\Repository\Values\User\Limitation\OwnerLimitation
<b>Type Class</b>	eZ\Publish\Core\Limitation\OwnerLimitationType
<b>Criterion used</b>	eZ\Publish\API\Repository\Values\Content\Query\Criterion\UserMetadata( UserMetadata::OWNER )
<b>Role Limitation</b>	no

## Possible values

Value	UI value	Description
1	"self"	Only user which is owner gets access (see " <a href="#">Legacy compatibility notes</a> " #2)
2	"session"	Same as "self" (see " <a href="#">Legacy compatibility notes</a> " #1)

## Legacy compatibility notes:

1. Owner("session") is deprecated and works exactly like "self" in Public API since it has no knowledge of user Sessions
2. User is no longer auto assumed to be owner of himself and get access to edit him/herself when Owner limitation is used in Public API\*

\* Workaround for Owner limitation on user object:

To make sure user gets access to himself when using Owner limitation across 4.x and 5.x, the solution is to change the user to be owner of himself.

This is accomplished using a privileged user to do the following API calls:

```
$user = $userService->loadUser( $userId );
$contentMetadataUpdateStruct = $contentService->newContentMetadataUpdateStruct();
$contentMetadataUpdateStruct->ownerId = $user->id;
$contentService->updateContentMetadata( $user->getVersionInfo()->getContentInfo(),
$contentMetadataUpdateStruct );
```