



Node Access for Common Folks With Uncommon Problems

Matt Kleve
(vordude)

Who is this guy?

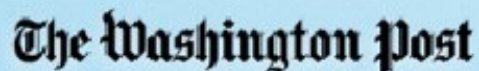
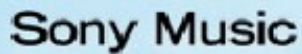


- 5 years of Drupal
- (bad) Module Maintainer
- Drupal Security Team
- Senior Developer, Lullabot



Lullabot™

Consulting | Development | Training



drupalize.me



Get instant access to an unrivaled library of Drupal training from top-tier experts streaming to your computer, tablet, smart phone, & tv.







a.



b.



c.

d.



f.



e.



g.



h.

You basic Node Access Needs



- **Who?** (The user on the site)
- **What are they doing?** (View? Edit? Delete? Create?)
- **To What?** (The Node They're doing it to)

WARNING



These first 2 tools work together, if you've never looked at this stuff before, it won't make sense until you hear about both of them.

Go with me for a moment.



hook_node_access_records()



“I see you're saving a node.

Based on that node,
tell me about the type
of user who should
have (view/update/delete)
permissions on this node.”

hook_node_access_records()



```
function image_hider_node_access_records($node) {
  $grants = array();
  if ($node->type == 'image') {
    $grants[] = array(
      'realm' => 'image_hider_author',
      'gid' => $node->uid,
      'grant_view' => 1,
      'grant_update' => 1,
      'grant_delete' => 1,
      'priority' => 0,
    );

    $grants[] = array(
      'realm' => 'image_hider_paid',
      'gid' => 42,
      'grant_view' => 1,
      'grant_update' => 1,
      'grant_delete' => 0,
      'priority' => 0,
    );

    return $grants;
  }
}
```


hook_node_access_records()



Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> select * from node_access;
```

nid	gid	realm	grant_view	grant_update	grant_delete
1	1	image_hider_author	1	1	1
1	42	image_hider_paid	1	1	0
134	0	all	1	0	0
135	1	image_hider_author	1	1	1
135	42	image_hider_paid	1	1	0
136	0	all	1	0	0
137	1	image_hider_author	1	1	1
137	42	image_hider_paid	1	1	0

```
8 rows in set (0.00 sec)
```

```
mysql>
```

hook_node_access_records_alter

7 node.api.php hook_node_access_records_alter(&\$grants, \$node)

```
8 node.api.php hook_node_access_records_alter(&$grants, Drupal node
Node $node)
```

Alter permissions for a node before it is written to the database.

Node access modules establish rules for user access to content. Node access records are stored in the {node_access} table and define which permissions are required to access a node. This hook is invoked after node access modules returned their requirements via [hook_node_access_records\(\)](#); doing so allows modules to modify the \$grants array by reference before it is stored, so custom or advanced business logic can be applied.

Upon viewing, editing or deleting a node, [hook_node_grants\(\)](#) builds a permissions array that is compared against the stored access records. The user must have one or more matching permissions in order to complete the requested operation.

A module may deny all access to a node by setting \$grants to an empty array.

Parameters

\$grants: The \$grants array returned by [hook_node_access_records\(\)](#).

\$node: The node for which the grants were acquired.

The preferred use of this hook is in a module that bridges multiple node access modules with a configurable behavior, as shown in the example with the 'is_preview' field.

Search 7

Function, file, or topic: *

API Navigation

[Drupal 7](#)

[Constants](#)

[Classes](#)

[Files](#)

[Functions](#)

[Globals](#)

[Topics](#)



hook_node_grants()



“So, a user wants to view/update/delete a node.

Based on that user,
I'll let node access system
if he's cool or not.”



hook_node_grants()

```
function image_hider_node_grants($account, $op) {
  $grants = array();

  $grants['image_hider_author'] = array($account->uid);

  if (($op == 'view' || $op == 'update') && user_access('view hidden images', $account)) {
    $grants['image_hider_paid'] = array(42);
  }

  return $grants;
}
```



hook_node_grants()

```
function image_hider_node_grants($account, $op) {  
    $grants = array();  
  
    $grants['image_hider_author'] = array($account->uid);  
  
    if (($op == 'view' || $op == 'update') && user_access('view hidden images', $account)) {  
        $grants['image_hider_paid'] = array(42);  
    }  
  
    return $grants;  
}
```

realm

grant_id

(To Match against hook_node_access_records().)

hook_node_grants_alter

```
7 node.api.php hook_node_grants_alter(&$grants, $account, $op)
```

```
8 node.api.php hook_node_grants_alter(&$grants, $account, $op)
```

Alter user access rules when trying to view, edit or delete a node.

Node access modules establish rules for user access to content. [hook_node_grants\(\)](#) defines permissions for a user to view, edit or delete nodes by building a `$grants` array that indicates the permissions assigned to the user by each node access module. This hook is called to allow modules to modify the `$grants` array by reference, so the interaction of multiple node access modules can be altered or advanced business logic can be applied.

The resulting grants are then checked against the records stored in the `{node_access}` table to determine if the operation may be completed.

A module may deny all access to a user by setting `$grants` to an empty array.

Parameters

\$grants: The `$grants` array returned by [hook_node_grants\(\)](#).

\$account: The user account requesting access to content.

\$op: The operation being performed, 'view', 'update' or 'delete'.

Developers may use this hook to either add additional grants to a user or to remove existing grants. These rules are typically based on either the permissions assigned to a user role, or specific attributes of a user account.

Search 7

Function, file, or topic: *

API Navigation

[Drupal 7](#)[Constants](#)[Classes](#)[Files](#)[Functions](#)[Globals](#)[Topics](#)

Node Access From a Query!



```
function node_page_default() {
    $select = db_select('node', 'n')
        ->fields('n', array('nid', 'sticky', 'created'))
        ->condition('promote', 1)
        ->condition('status', 1)
        ->orderBy('sticky', 'DESC')
        ->orderBy('created', 'DESC')
        ->extend('PagerDefault')
        ->limit(variable_get('default_nodes_main', 10))
        ->addTag('node_access'); ←
    $nids = $select->execute()->fetchCol();
}
```




hook_node_access()



```
function image_hider_node_access($node, $op, $account) {
    $type = is_string($node) ? $node : $node->type;

    if ($op == 'view' && date(w, REQUEST_TIME) == 5 && $type == 'image') {
        return NODE_ACCESS_DENY;
    }

    return NODE_ACCESS_IGNORE;
}
```


hook_node_access()



“Forget those guys,

That stuff is hard.

I'll do whatever you want.

Well,

almost.”





Enough
THEORY

CAUTION-HOPPERS A AND AC:
CLOSE HOPPER SLIDES ON OPPOSITE
COMPARTMENT BEFORE OPENING

VIBRATOR

STRA

CLEAN INSIDE GROOVES AND TOP OF SLIDES
BEFORE CLOSING

AC
1199 CU FT

SELF LOCK / UNLOCK
GEARS-GATE
OPEN