

Exercice 3

a. le chemin admin/modules

L'item de menu correspondant au chemin `admin/modules` est à la ligne **668** du fichier `modules/system/system.module`.

```
$items['admin/modules'] = array(  
  'title' => 'Modules',  
  'description' => 'Extend site functionality.',  
  'page callback' => 'drupal_get_form',  
  'page arguments' => array('system_modules'),  
  'access arguments' => array('administer modules'),  
  'file' => 'system.admin.inc',  
  'weight' => -2,  
);
```

Cet item permet d'afficher un formulaire, car il fait appel à la fonction `drupal_get_form`. Cette fonction se trouve dans le fichier `system.admin.inc`, et il faut les permissions administrateur pour afficher ce formulaire.

b. le formulaire system_modules

Le nom de la fonction qui va récupérer les informations de tous les modules pour les mettre dans le tableau `$files` est `system_rebuild_module_data()`.

La boucle qui itère sur la liste des modules commence à la ligne **797**.

La ligne où commence le code qui crée le lien Help d'un module est la ligne **866**.

```
if ($help_arg && $module->status && in_array($filename, module_implements('help')))  
{  
  if (module_invoke($filename, 'help', "admin/help#$filename", $help_arg)) {  
    $extra['links']['help'] = array(  
      '#type' => 'link',  
      '#title' => t('Help'),  
      '#href' => "admin/help/$filename",  
      '#options' => array('attributes' => array('class' => array('module-link',  
        'module-link-help'), 'title' => t('Help'))),  
    );  
  }  
}
```

La ligne où commence le code qui crée le lien Permissions d'un module est la ligne **877**.

```
if ($module->status && user_access('administer permissions') && in_array($filename,  
module_implements('permission')) {  
  $extra['links']['permissions'] = array(  
    '#type' => 'link',  
    '#title' => t('Permissions'),
```

```

        '#href' => 'admin/people/permissions',
        '#options' => array('fragment' => 'module-' . $filename, 'attributes' =>
array('class' => array('module-link', 'module-link-permissions'), 'title' =>
t('Configure permissions'))),
    );
}

```

La ligne où commence le code qui crée le lien Configure d'un module est la ligne **887**.

```

if ($module->status && isset($module->info['configure'])) {
    $configure_link = menu_get_item($module->info['configure']);
    if ($configure_link['access']) {
        $extra['links']['configure'] = array(
            '#type' => 'link',
            '#title' => t('Configure'),
            '#href' => $configure_link['href'],
            '#options' => array('attributes' => array('class' => array('module-link',
'module-link-configure'), 'title' => $configure_link['description'])),
        );
    }
}

```

La fonction `_system_modules_build_row` est appelée à la ligne **913** pour ajouter la ligne au formulaire.

Exercice 4

En cliquant sur le bouton Save configuration, nous exécutons la fonction nommée **system_modules_submit** (fichier `modules/system/system.module`, ligne **1109**) qui appelle (ligne **1228**) la fonction `module_enable`.

Lors de la première activation d'un module (ou lorsqu'un module est réactivé après une désinstallation), la fonction `module_enable` (fichier `includes/module.inc` ligne **400**), déclenche l'exécution du `hook_schema` puis exécute le `hook_install` du module. Plus précisément, `module_enable`, appelle (ligne **483**) la fonction `drupal_install_schema`.

La fonction `drupal_install_schema` (fichier `includes/common.inc` ligne **7223**) appelle d'abord la fonction `drupal_get_schema_unprocessed` (ligne **7224**).

La fonction `drupal_get_schema_unprocessed` (fichier `includes/common.inc` ligne **7283**) récupère le schéma d'un module en exécutant : `$schema = module_invoke($module, 'schema');` (ligne **7286**)

La fonction `drupal_install_schema` appelle (ligne **7225**) la fonction `_drupal_schema_initialize` pour compléter le schéma

La fonction `drupal_install_schema` commande la création des tables du schéma en exécutant (ligne **7227**) :

```
foreach ($schema as $name => $table) { db_create_table($name, $table); }
```

La fonction `db_create_table` (fichier `includes/database/database.inc` ligne **2837**) exécute l'appel de méthode : `Database::getConnection()->schema()->createTable($name, $table);`

La méthode `createTable` de la classe `DatabaseSchema` (fichier `includes/database/schema.inc` ligne **733**) appelle la méthode `createTableSql`.

La méthode `createTableSql` de la classe `DatabaseSchema_mysql`, qui se trouve dans le fichier (fichier `includes/database/mysql/schema.inc`) ligne (ligne **83**), génère la requête SQL de création d'une table à partir de son schéma.

La méthode `createTable` lance ensuite (ligne **735**) l'exécution de la requête.

Le `hook_install` est, quant à lui, appelé ensuite par `module_enable` à la ligne (ligne **498**) par l'expression :
`module_invoke($module, 'install');`
