Egészítsük ki a Drupal-t

Drupal modul fejlesztés

Drupal 6.0

2008. február 13.

Miért írjunk Drupal modult?

- Nincs az igényeinknek megfelelő modul
- Valamilyen közösségi igény kielégítése
- Valami nem úgy működik a „Core” modulokban, ahogy az nekünk kellene
- Egy másik rendszert szeretnénk integrálni a Drupal-ba
- Később is lesz kapacitásunk a modult karban tartani

Kódolási szabályok

- Bővebben: http://drupal.org/coding-standards
- Tab = 2 space
- Feltétel nélküli include = require_once()
- Feltételes include = include_once()
- Csak a <?php ?> tag az elfogadott
- Kötelezően a // Id kell a fejlécbe, ezt később majd a CVS fogja kitölteni
- A konstansok csupa nagybetűsek
- Isset() vagy empty()?

Az .info file-ok

- A modulok meta adatait tartalmazza
- .ini fájl formátomúak
- Kötelező attribútumok:
  - ; Id //CVS használja meta adatokra
  - name = A modul neve
  - description = A modul leírása
  - core = 6.x //melyik Drupal alaprendszerhez lett írva
- Opcionális attributomok:
  - version = "$Name$" //CVS irja a verziót
  - dependencies = ezektől a moduloktól függ

A kampók, hurkok

- Olyan függvények, melyek segítségével befolyásolhatjuk az alap rendszer működését. Azaz kiegészíthetjük, „megcsonkíthatjuk” vagy teljes egészében helyetesíthetjük az egyes rendszer funkciókat, mint pl.: az azonosító rendszert, a hozzáférési rendszert, stb.

Fordítható felület

- Nem összetévesztendő a tartalom fordítással
- t() függvény
- format_plural() // többesszám kezelés
  - !variable: egyszerűen behelyetesítődik a változó tartalma
  - @variable: egyszerű szövegé alakítja a változó tartalmát megegyezik a check_plain() függvénnyel
  - %variable: simnkelt egyszerű szöveggé alakítja a váltzó tartalmát (check_plain + theme_placeholder)
- .po fileok
- http://drupal.org/project/potx

A block API

- A blokkok készítésért felelős
- Fő elemei:
  - list : a blokk listában ez a szöveg jelenik
  - view: a blokk tartamát jeleníti meg
  - config: konfigurációs felületet nyújt a blokk számára
  - save: a blokk beállításainak tárolásáért felelős

Közös SQL

- Miért van rá szükség? API?
- db_query('SELECT * FROM {node}) // adatbázis lekérdező függvény
- db_query_range() // tartomány szerinti lekérdezés
- db_rewrite_sql()
- db_result() // egyetlen sorral tér vissza
- pager_query() // lapozható kimenettel tér vissza
- db_query_temporary() // ideglenes tábla készítés
- hook_db_rewrite_sql() // átírhatunk vele SQL lekérdezéseket
- Több adatbázis használata:
  - settings.php -ban:
  - $db_url['default'] = 'mysql://user:password@localhost/drupal';
  - $db_url['masik'] = 'mysql://user:password@localhost/masik';
  - Adatbázis váltás pl.: db_set_active('masik')
- Adatbázistól nem függő adatbázis séma

A kimenet és bemenet ellenőrzése

- form_id_validate() // a felhasználó bemenet elő szűrése
- check_plain() // egyszerű szöveggé alakít
- check_markup() // filterek segítségével megszüri a kimenetet
- filter_xss() // xss támadások ellen véd

A node API

- hook_access() // tartalom elérés szabályzás
- hook_nodeapi() //
- hook_node_info()
- hook_theme()
- hook_view()

Az .install fileok 1/3

```
<?php
 // $Id$
 function sajat_install() {
   drupal_set_message(t('A saját modul installációja megkezdődött.'));
   switch ($GLOBALS['db_type']) {
     case 'mysql':
     case 'mysqli':
       db_query("CREATE TABLE annotations (
         id int NOT NULL default 0,
         note longtext NOT NULL,
         timestamp int NOT NULL default 0,
         PRIMARY KEY (uid, nid)
         ) /*!40100 DEFAULT CHARACTER SET utf8 */;"
       );
       $success = TRUE;
       break;
```

Az .install fileok 2/3

```
    case 'pgsql':
      db_query("CREATE TABLE sajat (
        id int NOT NULL DEFAULT 0,
        note text NOT NULL,
        timestamp int NOT NULL DEFAULT 0,
        PRIMARY KEY (uid, nid)
        );"
      );
      $success = TRUE;
      break;
    default:
      drupal_set_message(t('Unsupported database.'));
  }
```

Az .install fileok 3/3

```
    if ($success) {
      drupal_set_message(t('The module installed tables successfully.'));
    }
    else {
      drupal_set_message(t('The installation of the annotate module
        was unsuccessful.'),'error');
    }
  }
```

A form API

- drupal_get_form('form_id') // form generálás
- *form_id*_validate($form, &$form_state) // form validálás
- form_id_submit($form, &$form_state) // a form adatainak feldolgozását végzi

Telepítsünk / Távolítsuk el

- install.php
  - hook_install() // az installláláskor lefutó szkriptek, általában adatbázis séma létrehozása
  - hook_uninstall() // a modul kikapcsolásakor lefutó szkriptek
  - hook_update_x() // ha újabb modult telepítünk ezeke a szriptek futnak le, az x az update számát jelenti

jQuery

- drupal_add_js() // javascript filokat includol az oldalba
- drupal_to_js() // php változót konvetál javascript változóvá
- drupal_get_js() // magával a javascript forrásával tér vissza html formátumban

Ne lőjünk Ágyúval verébre!

- hook_form_alter()

```
function sajat_form_alter(&$form, $form_state, $form_id) {
  if (isset($form['type']) && $form['type']['#value'] .'_node_settings' ==
$form_id) {
    $form['workflow']['upload_'. $form['type']['#value']] = array(
      '#type' => 'radios',
      '#title' => t('Attachments'),
      '#default_value' => variable_get('upload_'. $form['type']['#value'], 1),
      '#options' => array(t('Disabled'), t('Enabled')),
    );
  }
}
```

Fő a biztonság

- SQL-ben placeholder db_query("SELECT n.nid FROM {node} n WHERE n.nid > %d", $nid);
- GET változóval nem módosítunk adatot!
- A felhasználói bemenetet szűrjük!
- A rendszer kimenetét ugyancsak szűrjük!
- JS-sel nem validálunk!

Hiba keresés

- Coder modul: a coding standardok betartására jó
- Devel modul: hiba keresésre,
- Xdebug

Gyakorlatok