

```

<?php
/**
 * Make a page edit.
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
along
 * with this program; if not, write to the Free Software Foundation,
Inc.,
 * 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
 * http://www.gnu.org/copyleft/gpl.html
 *
 * @file
 */

```

```

class GetData {
// Initialize cURL session
$ch = curl_init();
$searchParameter = $this->getOption ("SearchParameter")

// Set cURL options
curl_setopt($ch, CURLOPT_URL, "https://worldwide.espacenet.com/
searchResults?
ST=singleline&locale=de_EP&submitted=true&DB=&query=" .
$searchParameter . "&Submit=SUCHEN");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_USERAGENT, $get_browser );
curl_setopt($ch, CURLOPT_COOKIEFILE, "cookies.txt");
curl_setopt($ch, CURLOPT_COOKIEJAR, "cookies.txt");

// Execute cURL request
$response = curl_exec($ch);

// Close cURL session
curl_close($ch);

// Extract desired information from HTML response
}

require_once __DIR__ . '/Maintenance.php';

/**
 * Maintenance script to make a page edit.

```

```

*
* @ingroup Maintenance
*/
class EditCLI {
    public function __construct() {
        parent::__construct();
        $this->mDescription = "Edit an article from the
command line, text is from stdin";
        $this->addOption( 'user', 'Username', false, true,
'u' );
        $this->addOption( 'summary', 'Edit summary', false,
true, 's' );
        $this->addOption( 'minor', 'Minor edit', false,
false, 'm' );
        $this->addOption( 'bot', 'Bot edit', false, false,
'b' );
        $this->addOption( 'autosummary', 'Enable
autosummary', false, false, 'a' );
        $this->addOption( 'no-rc', 'Do not show the change
in recent changes', false, false, 'r' );
        $this->addOption( 'nocreate', 'Don\'t create new
pages', false, false );
        $this->addOption( 'createonly', 'Only create new
pages', false, false );
        $this->addArg( 'title', 'Title of article to
edit' );
    }

    public function execute() {
        global $wgUser;

        $userName = $this->getOption( 'user', 'Maintenance
script' );

        $summary = $this->getOption( 'summary', '' );
        $minor = $this->hasOption( 'minor' );
        $bot = $this->hasOption( 'bot' );
        $autoSummary = $this->hasOption( 'autosummary' );
        $noRC = $this->hasOption( 'no-rc' );

        $wgUser = User::newFromName( $userName );
        if ( !$wgUser ) {
            $this->error( "Invalid username", true );
        }
        if ( $wgUser->isAnon() ) {
            $wgUser->addToDatabase();
        }

        $title = Title::newFromText( $this->getArg() );
        if ( !$title ) {
            $this->error( "Invalid title", true );
        }

        if ( $this->hasOption( 'nocreate' ) && !$title-
>exists() ) {

```

```

        $this->error( "Page does not exist", true );
    } elseif ( $this->hasOption( 'createonly' ) &&
$title->exists() ) {
        $this->error( "Page already exists", true );
    }

    $page = WikiPage::factory( $title );

    # Read the text
    $text = $this->getStdin( Maintenance::STDIN_ALL );
    $content = ContentHandler::makeContent( $text,
$title );

    # Do the edit
    $this->output( "Saving... " );
    $status = $page->doEditContent( $content, $summary,
        ( $minor ? EDIT_MINOR : 0 ) |
        ( $bot ? EDIT_FORCE_BOT : 0 ) |
        ( $autoSummary ? EDIT_AUTOSUMMARY : 0 ) |
        ( $noRC ? EDIT_SUPPRESS_RC : 0 ) );
    if ( $status->isOK() ) {
        $this->output( "done\n" );
        $exit = 0;
    } else {
        $this->output( "failed\n" );
        $exit = 1;
    }
    if ( !$status->isGood() ) {
        $this->output( $status->getWikiText() .
"\n" );
    }
    exit( $exit );
}

}

$maintClass = "EditCLI";
require_once RUN_MAINTENANCE_IF_MAIN;

```