

```
<?php
/**
 * Interface for ILS Drivers
 *
 * PHP version 5
 *
 * Copyright (C) Villanova University 2007.
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2,
 * as published by the Free Software Foundation.
 *
 * 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., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 *
 * @category VuFind
 * @package ILS_Drivers
 * @author Andrew S. Nagy <vufind-tech@lists.sourceforge.net>
 * @license http://opensource.org/licenses/gpl-2.0.php GNU General Public License
 * @link http://vufind.org/wiki/building_an_ils_driver Wiki
 */

/**
 * Catalog Specific Driver Class
 *
 * This interface class is the definition of the required methods for
 * interacting with the local catalog.
 *
 * The parameters are of no major concern as you can define the purpose of the
 * parameters for each method for whatever purpose your driver needs.
 * The most important element here is what the method will return. In all cases
 * the method can return a PEAR_Error object if an error occurs.
 *
 * @category VuFind
 * @package ILS_Drivers
 * @author Andrew S. Nagy <vufind-tech@lists.sourceforge.net>
 * @license http://opensource.org/licenses/gpl-2.0.php GNU General Public License
 * @link http://vufind.org/wiki/building_an_ils_driver Wiki
 */
interface DriverInterface
{
    /**
     * Get Status
     *
     * This is responsible for retrieving the status information of a certain
     * record.
     *
     * @param string $id The record id to retrieve the holdings for
     *
     * @return mixed On success, an associative array with the following keys:
     * id, availability (boolean), status, location, reserve, callnumber; on
```

```
* failure, a PEAR_Error.
* @access public
*/
public function getStatus($id);

/**
 * Get Statuses
 *
 * This is responsible for retrieving the status information for a
 * collection of records.
 *
 * @param array $ids The array of record ids to retrieve the status for
 *
 * @return mixed An array of getStatus() return values on success,
 * a PEAR_Error object otherwise.
 * @access public
 */
public function getStatuses($ids);

/**
 * Get Holding
 *
 * This is responsible for retrieving the holding information of a certain
 * record.
 *
 * @param string $id The record id to retrieve the holdings for
 * @param array $patron Patron data
 *
 * @return mixed On success, an associative array with the following keys:
 * id, availability (boolean), status, location, reserve, callnumber, duedate,
 * number, barcode; on failure, a PEAR_Error.
 * @access public
 */
public function getHolding($id, $patron = false);

/**
 * Get Purchase History
 *
 * This is responsible for retrieving the acquisitions history data for the
 * specific record (usually recently received issues of a serial).
 *
 * @param string $id The record id to retrieve the info for
 *
 * @return mixed An array with the acquisitions data on success, PEAR_Error
 * on failure
 * @access public
 */
public function getPurchaseHistory($id);
}
```

?>