Persistence Done

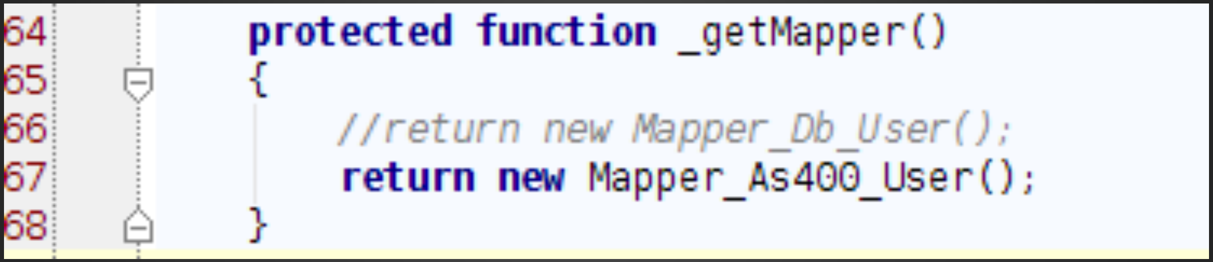
Yay

Now - let's introduce the value!

Users must be sync'd to AS400

- Previously was from mysql db
- Need to change as little code as possible
- The only thing changing is how we map data
  - to / from the data source
  - "map"... mapper
- Lets substitute in a different mapper

Modify User Service

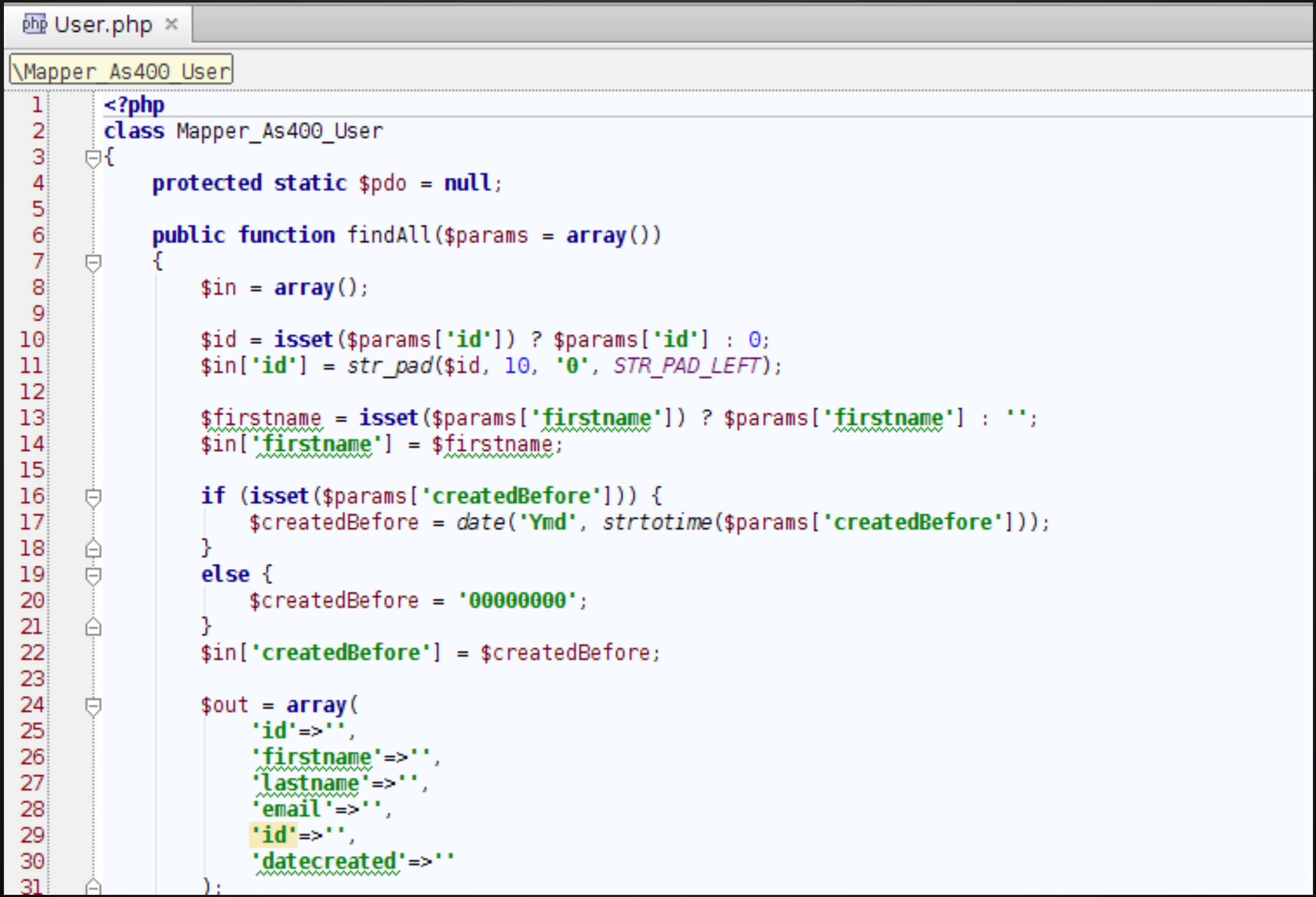


```
    protected function _getMapper()
    {
        //return new Mapper_Db_User();
        return new Mapper_As400_User();
    }
```

This is the ONLY change to your code.

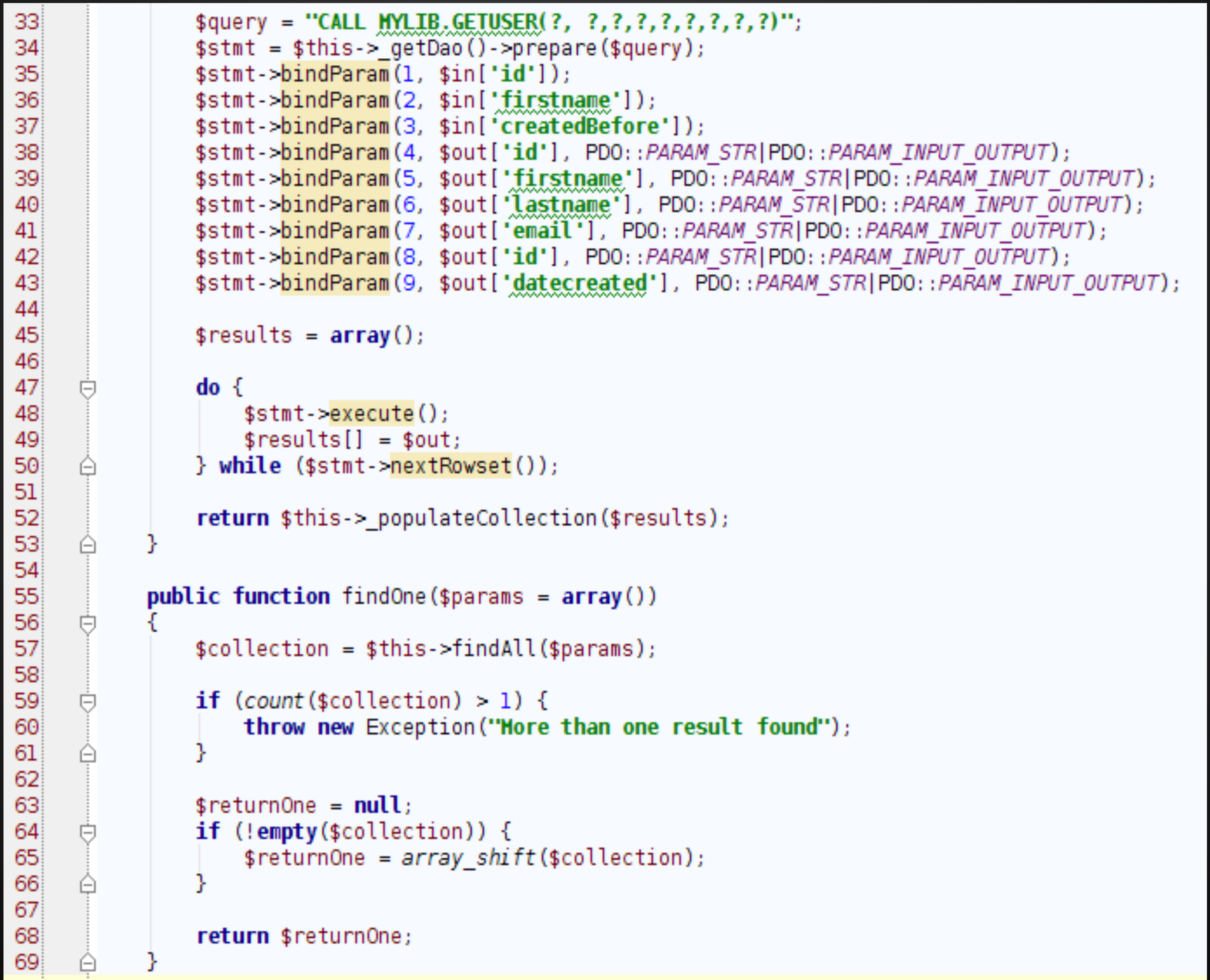
All else is NEW code.

What does this mapper look like?



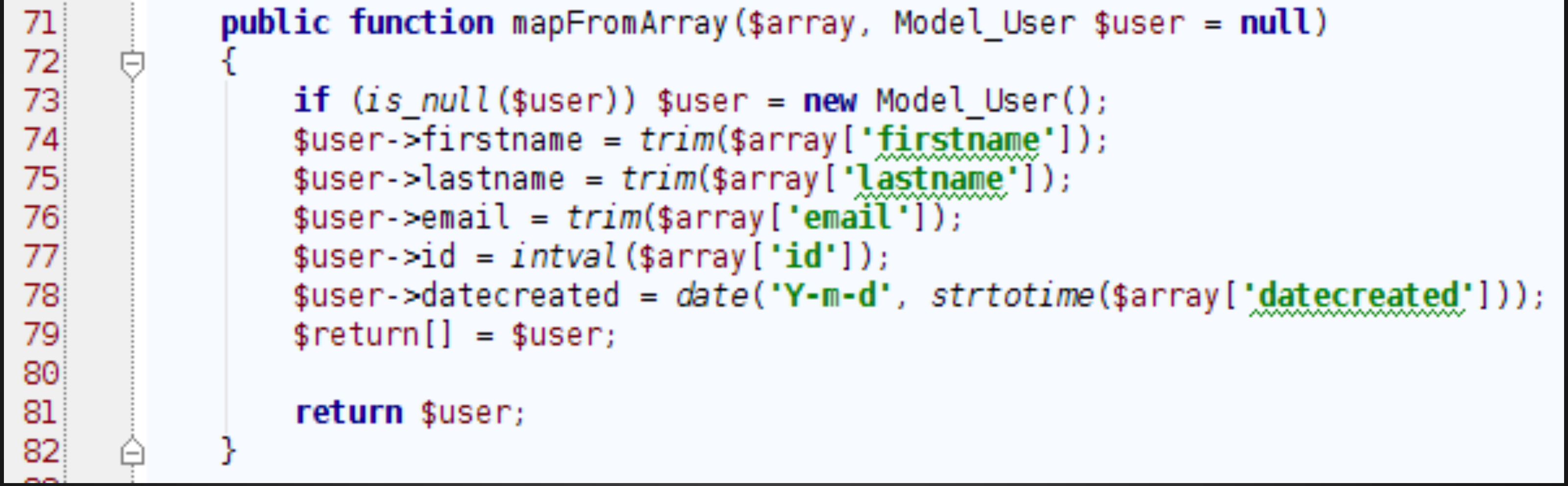
```php
<?php
class Mapper_As400_User
{
    protected static $pdo = null;

    public function findAll($params = array())
    {
        $in = array();

        $id = isset($params['id']) ? $params['id'] : 0;
        $in['id'] = str_pad($id, 10, '0', STR_PAD_LEFT);

        $firstname = isset($params['firstname']) ? $params['firstname'] : '';
        $in['firstname'] = $firstname;

        if (isset($params['createdBefore'])) {
            $createdBefore = date('Ymd', strtotime($params['createdBefore']));
        }
        else {
            $createdBefore = '00000000';
        }
        $in['createdBefore'] = $createdBefore;

        $out = array(
            'id'=>'',
            'firstname'=>'',
            'lastname'=>'',
            'email'=>'',
            'id'=>'',
            'datecreated'=>''
        );
```

AS400 Mapper Continued



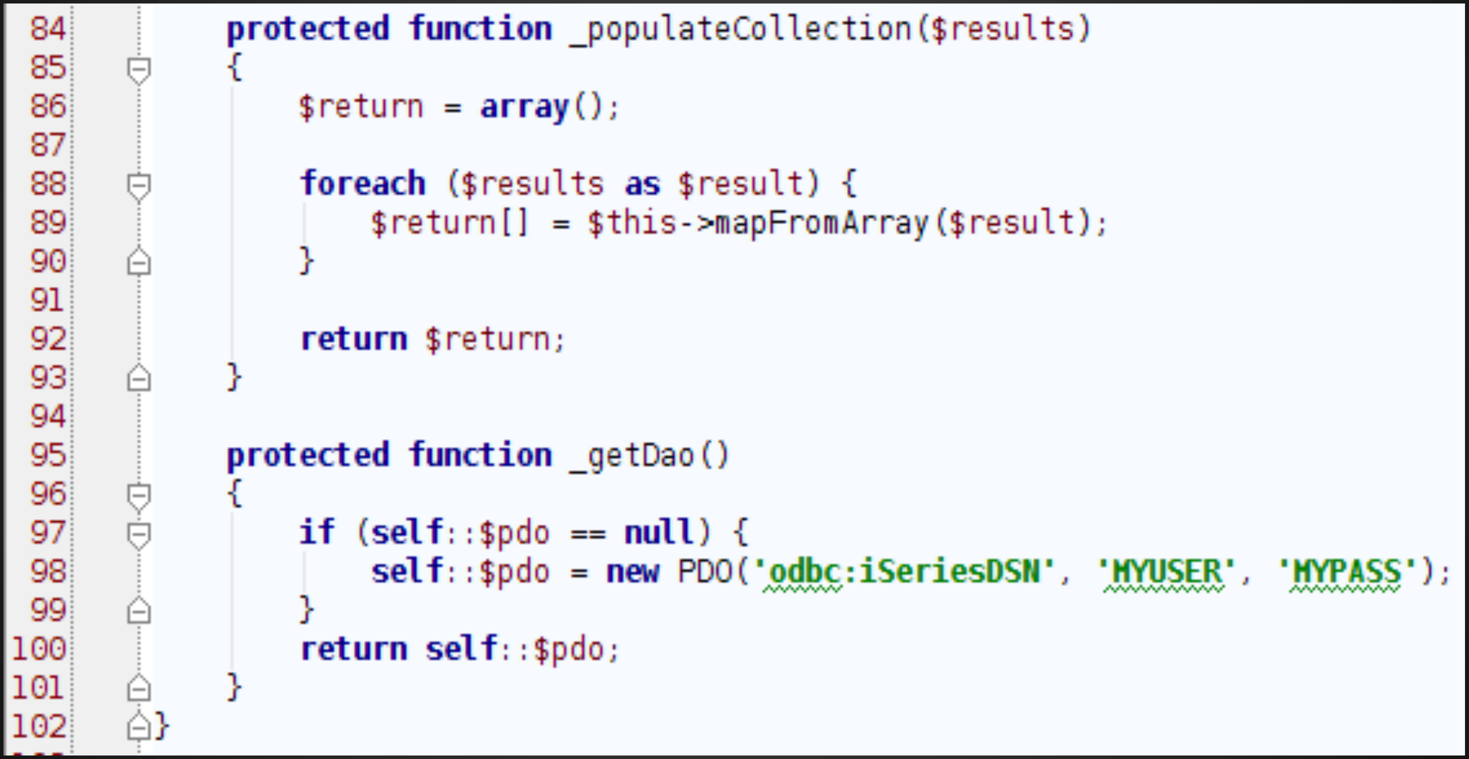
```php
        $query = "CALL MYLIB.GETUSER(?, ?,?,?,?,?,?,?,?)";
        $stmt = $this->_getDao()->prepare($query);
        $stmt->bindParam(1, $in['id']);
        $stmt->bindParam(2, $in['firstname']);
        $stmt->bindParam(3, $in['createdBefore']);
        $stmt->bindParam(4, $out['id'], PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT);
        $stmt->bindParam(5, $out['firstname'], PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT);
        $stmt->bindParam(6, $out['lastname'], PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT);
        $stmt->bindParam(7, $out['email'], PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT);
        $stmt->bindParam(8, $out['id'], PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT);
        $stmt->bindParam(9, $out['datecreated'], PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT);

        $results = array();

        do {
            $stmt->execute();
            $results[] = $out;
        } while ($stmt->nextRowset());

        return $this->_populateCollection($results);
    }

    public function findOne($params = array())
    {
        $collection = $this->findAll($params);

        if (count($collection) > 1) {
            throw new Exception("More than one result found");
        }

        $returnOne = null;
        if (!empty($collection)) {
            $returnOne = array_shift($collection);
        }

        return $returnOne;
    }
```

AS400 Mapper Continued



```php
    public function mapFromArray($array, Model_User $user = null)
    {
        if (is_null($user)) $user = new Model_User();
        $user->firstname = trim($array['firstname']);
        $user->lastname = trim($array['lastname']);
        $user->email = trim($array['email']);
        $user->id = intval($array['id']);
        $user->datecreated = date('Y-m-d', strtotime($array['datecreated']));
        $return[] = $user;

        return $user;
    }
```

AS400 Mapper Continued



```php
    protected function _populateCollection($results)
    {
        $return = array();

        foreach ($results as $result) {
            $return[] = $this->mapFromArray($result);
        }

        return $return;
    }

    protected function _getDao()
    {
        if (self::$pdo == null) {
            self::$pdo = new PDO('odbc:iSeriesDSN', 'MYUSER', 'MYPASS');
        }
        return self::$pdo;
    }
}
```

Whew! Done!

Check it out at

http://aaronsaray.com/midwestphp

Tweet at @aaronsaray