

# How to create a LORIS module?

Cécile Madjar  
Montreal brainhack school 2018



# LORIS

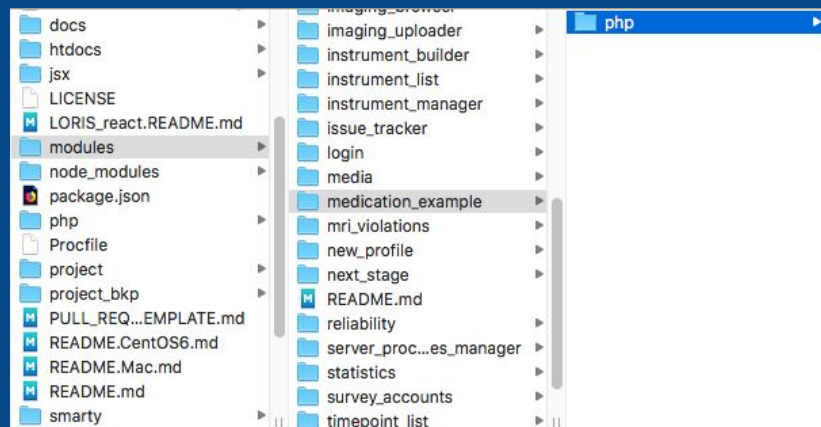
- Source code available via Github at <https://github.com/aces/Loris>
- LORIS demo website: <https://demo.loris.ca/main.php>
- LORIS is designed around the concept of modules. This allows:
  - Isolation of the code for each module (CSS, JS, etc...)
  - Easy creation of new modules

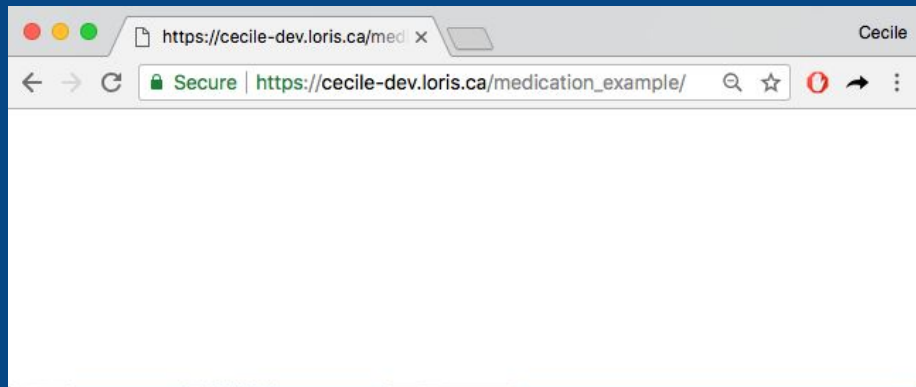
Create a LORIS page in the module

# Create the module's basic directory structure

- For this example, we are going to create a 'medication\_example' module
  - First, create the 'medication\_example' folder under the 'modules' directory of LORIS
  - Then, within the 'medication\_example' folder, create a 'php' directory

```
loris$ mkdir modules/medication_example
loris$ mkdir modules/medication_example/php
loris$
```





Apache log in `/var/log/apache2/loris-error.log`

```
[Wed May 16 08:16:48.742000 2018] [:error] [pid 31212] [client 132.216.237.19:4954  
8] Module medication_example is missing Module class descriptor. Modules missing a  
Module file will stop working in a future version of LORIS.  
[Wed May 16 08:16:48.742947 2018] [:error] [pid 31212] [client 132.216.237.19:4954  
8] Could not load Loris module medication_example.
```



Need to create a `module.class.inc` PHP file

# Creation of module.class.inc

- PHP file to be created in medication\_example/php
- Present in all modules' php directory

```
<?php
namespace LORIS\medication_example;

/**
 * Class module implements the basic LORIS module functionality
 *
 * @category Behavioural
 * @package Main
 * @author Cécile Madjar <cecile.madjar2@mcgill.ca>
 * @license http://www.gnu.org/licenses/gpl-3.0.txt GPLv3
 * @link https://www.github.com/aces/Loris/
 */
class Module extends \Module
{
}
```



This class will inherit functionalities from the main module found in \$LORIS\_repo\$/php/libraries/Module.class.inc



Need to create the module's actual PHP file  
and the template file (with HTML code)

# Creation of php/medication\_module.class.inc

- File to be created in the module's 'php' directory
- Name of the file should be the exact name of the module  
=> medication\_example.class.inc
- Name of the class within the file should be the name of the module with first letter of each word capitalized  
=> Medication\_Example

```
<?php

namespace LORIS\medication_example;

class Medication_Example extends \NDB_Form {

    function setup() { // setup function
        parent::setup();

        $this->addBasicText(
            field: 'helloWorld',
            label: 'This is a simple text form'
        );
    }
}
```

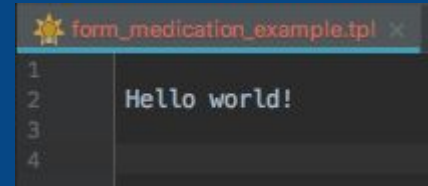


# Creation of templates/form\_medication\_example.tpl

```
loris$ mkdir modules/medication_example/templates
loris$
```

- File created in the 'templates' directory of 'medication\_example'
- Name of the file should be the exact name of the module preceded by 'form' in this example since using NDB\_Form in the PHP class file.  
=> form\_medication\_example.tpl
- Will store the HTML code of the page

*For now, let's just write the famous  
'Hello world!' into the template file*





Note how the header, footer and breadcrumb are automatically added to any new module page.

Create a LORIS form

# Creation of a simple form (medication\_example.class.inc)

- In the php directory of the module
- Creates the placeholders for the HTML fields
  - addBasicText()
  - addSelect()
  - addBasicDate()

```
<?php
namespace LORIS\medication_example;

class Medication_Example extends \NDB_Form {

    function setup () { // setup function
        parent::setup();

        $this->addBasicText( // creates a simple text input field
            field: "id",
            label: "Candidate's identifier"
        );
        $this->addSelect( // creates a drop down with array options
            name: "medication",
            label: "Medication",
            array(
                "" => "",
                "ASA" => "acetylsalicylic acid (ASA)",
                "NSAID" => "nonsteroidal anti-inflammatory drug (NSAID)",
                "APAP" => "acetaminophen (APAP)"
            )
        );
        $this->addBasicDate( // creates a date field
            field: "startDate",
            label: "Medication start date"
        );
        $this->addBasicDate( // creates a date field
            field: "endDate",
            label: "Medication end date"
        );
    }
}
```

# Creation of a simple form (form\_medication\_example.tpl)

- In the templates directory of the module
- Sets the HTML layout of the page:
  - Candidate ID input text
  - Medication drop down
  - Start date
  - End date
  - Submit button

```
<!-- create the HTML form -->
<form method="post" name="medication_example">

  <!-- display PSCID input text box -->
  {$form.id.label}
  {$form.id.html}
  <br>

  <!-- display medication drop down -->
  {$form.medication.label}
  {$form.medication.html}
  <br>

  <!-- display medication start date -->
  {$form.startDate.label}
  {$form.startDate.html}
  <br>

  <!-- display medication end date -->
  {$form.endDate.label}
  {$form.endDate.html}
  <br>

  <!-- display the submit button -->
  <input type="submit" name="fire_away">

</form>
```

# Creation of a simple form

medication\_example.class.inc

```
<?php
namespace LORIS\medication_example;

class Medication_Example extends \NDB_Form {

    function setup () { // setup function
        parent::setup();

        $this->addBasicText( // creates a simple text input field
            field: "id",
            label: "Candidate's identifier"
        );
        $this->addSelect( // creates a drop down with array options
            name: "medication",
            label: "Medication",
            array(
                "" => "",
                "ASA" => "acetylsalicylic acid (ASA)",
                "NSAID" => "nonsteroidal anti-inflammatory drug (NSAID)",
                "APAP" => "acetaminophen (APAP)"
            )
        );
        $this->addBasicDate( // creates a date field
            field: "startDate",
            label: "Medication start date"
        );
        $this->addBasicDate( // creates a date field
            field: "endDate",
            label: "Medication end date"
        );
    }
}
```

form\_medication\_example.tpl

```
<!-- create the HTML form -->
<form method="post" name="medication_example">

    <!-- display PSCID input text box -->
    {$form.id.label}
    {$form.id.html}
    <br>

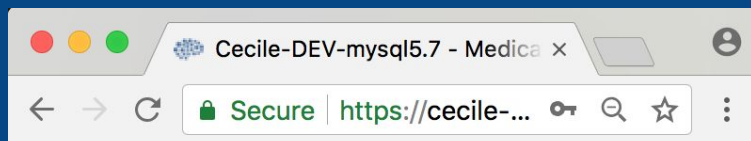
    <!-- display medication drop down -->
    {$form.medication.label}
    {$form.medication.html}
    <br>

    <!-- display medication start date -->
    {$form.startDate.label}
    {$form.startDate.html}
    <br>

    <!-- display medication end date -->
    {$form.endDate.label}
    {$form.endDate.html}
    <br>

    <!-- display the submit button -->
    <input type="submit" name="fire_away">

</form>
```



LORIS: DEV



Medication Example

Candidate's identifier

Medication

Medication start date

Medication end date

Submit

Medication

☒

acetylsalicylic acid (ASA)  
nonsteroidal anti-inflammatory drug (NSAID)  
acetaminophen (APAP)  
yyyy-mm-dd

Medication start date

yyyy-mm-dd

May 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Mantis |  
rights reserved.

# Map the submit button to print in the console

- Practical for debugging purposes
- Functions that print in the console:
  - `print`
  - `print_r`
  - `var_dump`



# Map the submit button to print in the console

```
<?php

namespace LORIS\medication_example;

class Medication_Example extends \WDB_Form {

    function setup () {
        parent::setup();

        $this->addBasicText( // creates a simple text input field...);
        $this->addSelect( // creates a drop down with array options...);
        $this->addBasicDate( // creates a date field...);
        $this->addBasicDate( // creates a date field...);

        // call _validate function to print in the console
        $this->form->addFormRule( array(&$this, '_validate') );

    }

    function _validate ($values) {
        var_dump($values);
    }

}
```

## Console Output

```
array(5) {
    ["id"]=>
    string(7) "MTL0001"
    ["medication"]=>
    string(3) "ASA"
    ["startDate"]=>
    string(10) "2018-05-10"
    ["endDate"]=>
    string(10) "2018-05-16"
    ["fire_away"]=>
    string(6) "Submit"
}
```

## Medication Example

Candidate's identifier

Medication

Medication start date

Medication end date

# Data entry validation and error printing

medication\_example.class.inc

```
<?php
namespace LORIS\medication_example;

class Medication_Example extends \NDB_Form {

    function setup () {...}

    function _validate($values) {
        //check that medication is not set to ""
        if ( $values['medication'] == "" ) {
            return array( 'medication' => "Medication cannot be empty");
        }
    }
}
```

form\_medication\_example.tpl

```
<!-- create the HTML form -->
<form method="post" name="medication_example">

    <!-- display PSCID input text box -->
    {$form.id.label}
    {$form.id.html}
    <br>

    <!-- display medication drop down -->
    {if $form.errors.medication}
        <div class="form-error">{$form.errors.medication}</div>
    {/if}
    {$form.medication.label}
    {$form.medication.html}
    <br>

    <!-- display medication start date -->
    {$form.startDate.label}
    {$form.startDate.html}
    <br>

    <!-- display medication end date -->
    {$form.endDate.label}
    {$form.endDate.html}
    <br>

    <!-- display the submit button -->
    <input type="submit" name="fire_away">

</form>
```



## Medication Example

Candidate's identifier

MTL0001

Medication cannot be empty

Medication

Medication start date

2018-05-10

Medication end date

2018-05-16

Submit

# Creation of the database table

- In LORIS MySQL database, create the new table that will host the data sent by the form using the following query

```
CREATE TABLE medication_example (  
  `MedicationExampleID` INT(6) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `ID` VARCHAR(255),  
  `Medication` VARCHAR(255),  
  `StartDate` DATE,  
  `EndDate` DATE,  
  PRIMARY KEY (MedicationExampleID)  
);
```

# Data insertion into the database upon submission

- Add a `_process()` function to `medication_example.class.inc`
- This function calls the Database class in order to insert in the `medication_example` table
- Note that at this point, the data has already been validated by the `_validate()` function

```
<?php
namespace LORIS\medication_example;

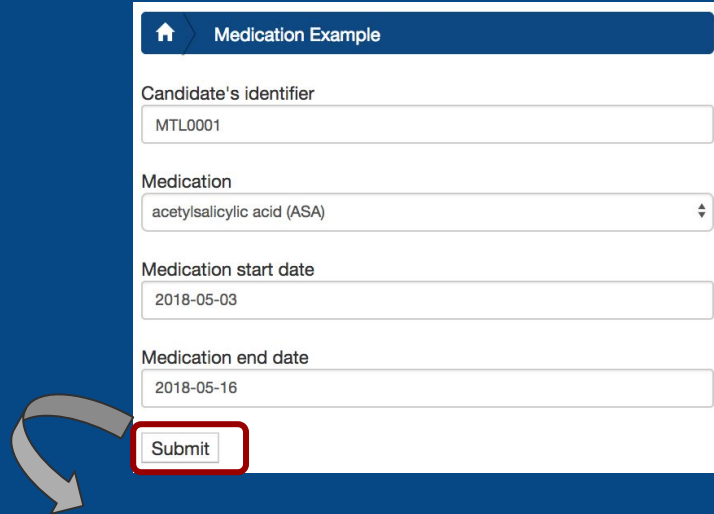
class Medication_Example extends \NDB_Form {

    function setup () {...}

    function _validate($values) {...}

    function _process($values) {
        // at this point, the data has already been validated
        $db = \Database::Singleton();
        $db->insert(
            'medication_example', // name of the table to insert to
            [ // array with values to insert (field => value)
                'ID' => $values['id'],
                'Medication' => $values['medication'],
                'StartDate' => $values['startDate'],
                'EndDate' => $values['endDate']
            ]
        );
    }
}
```

# Data insertion into the database upon submission



Medication Example

Candidate's identifier  
MTL0001

Medication  
acetylsalicylic acid (ASA)

Medication start date  
2018-05-03

Medication end date  
2018-05-16

Submit

```
mysql> select * from medication_example;
```

MedicationExampleID	PSCID	Medication	StartDate	EndDate
1	MTL0001	ASA	2018-05-03	2018-05-16

1 row in set (0.01 sec)

Create a LORIS data table

# Creation of a new page that will host the data table

- Creation of two new files that will create the data table page
  - `medication_data_table.class.inc` (in `modules/medication_example/php` folder)
  - `menu_medication_data_table.tpl` (in `modules/medication_example/templates` folder)
- Note how the template file has the same name as the PHP class file with the 'menu\_' string added to the beginning of the filename



## medication\_data\_table.class.inc

```
<?php
namespace LORIS\medication_example;

class Medication_Data_Table extends \NDB_Menu_Filter {

    var $AjaxModule = true; // stops from running the query twice

    function _setupVariables () { // setup the table structure
        // set the field names of the MySQL table
        $this->columns = array(
            'ID', 'Medication', 'StartDate', 'EndDate'
        );
        // set the FROM part of the query
        $this->query = " FROM medication_example";
    }

}
```

## menu\_medication\_data\_table.tpl

```
<div id="datatable" />

<script>
    var table = RDynamicDataTable(
    {
        "DataURL" : "{$baseurl}/medication_example/medication_data_table/?format=json"
    }
    );
    ReactDOM.render(table, document.getElementById("datatable"));
</script>
```

Medication Example Medication Data Table

4 rows displayed of 4. (Maximum rows per page: 20 ▾)

No.	ID	Medication	StartDate	EndDate
1	MTL0001	ASA	2018-05-03	2018-05-16
2	MTL0002	NSAID	2018-04-06	2018-05-01
3	MTL0003	APAP	2018-03-05	2018-03-06
4	MTL0010	ASA	2018-05-03	2018-05-05

4 rows displayed of 4. (Maximum rows per page: 20 ▾)

Download Table as CSV

# Important to remember when creating a module

- The module class (`module.class.inc`) itself is required
- Module name (`module_name`) must be identical to the module namespace (`LORIS\module_name`)
- Need to name the class in a specific way (`module_name.class.inc`)
- The class name (`Module_Name`) within the module needs to map to the page name (`module_name.class.inc`)
- Need a database table to query or insert into
- By default the module will load the page with the same name as the module

# What else can be done within a module

- Add CSS dependencies using `getCSSDependencies()` in the PHP file
  - These CSS scripts would be stored in `modules/medication_example/css`
- Add JS dependencies using `getJSDependencies()` in the PHP file
  - These JS scripts would be stored in `modules/medication_example/js`
- Check permissions using `_hasAccess()` in the PHP file
- Add a filter to the data table
- Anything else that you want!

# Example of a getCSSDependencies() function

```
/**
 * Include additional CSS files:
 * imaging_uploader.css
 *
 * @return array of css to be inserted
 */
function getCSSDependencies()
{
    $factory = \NDB_Factory::singleton();
    $baseUrl = $factory->settings()->getBaseUrl();
    $deps    = parent::getCSSDependencies();
    return array_merge(
        $deps,
        [$baseUrl . "/medication_example/css/medication_example.css"]
    );
}
```

# Example of a getJSDependencies() function

```
/**
 * Include any JS script required for the page
 *
 * @return array of javascript to be inserted
 */
function getJSDependencies()
{
    $factory = \NDB_Factory::singleton();
    $baseUrl = $factory->settings()->getBaseUrl();
    $deps     = parent::getJSDependencies();
    return array_merge(
        $deps,
        array(
            $baseUrl . "/medication_example/js/index.js",
            $baseUrl . "/medication_example/js/medication_example.js",
        )
    );
}
```

# Example of a `_hasAccess()` function

```
/**
 * The _has_access returns true
 * if the user has the specific permission
 *
 * @return boolean
 */
function _hasAccess()
{
    // create user object
    $user = \User::singleton();
    return $user->hasPermission('medication_example');
}
```