



## Why I'm here

- Author of *Developing Extensions for Joomla! 5*
- I love CLI!!
- Some projects I have made with CLI:
  - Create email queues
  - Check renewal subscriptions
  - Update 26000 products in a HikaShop site

**Carlos  
Cámara**

# The Grand Tour of Today's Adventures

1. Why CLI?
2. Working with the console in Joomla!
3. Writing our first console plugin in Joomla!
4. Adding options to our command

You may download the most recent version of this presentation at:

<https://developingextensionsforjoomla5.com/jdayusa2025/>

## Why CLI?

Source: <https://workchronicles.substack.com/p/comic-automation>

## A bit of Joomla! Console

```
php cli/joomla.php --list
```

## References

### Chapter 9

# A bit of Joomla! Console

## References

### Chapter 9

## Some usage tips

1. All the commands are run using the command:

```
php cli/joomla.php COMMAND
```

2. You can run the command with the option `--help` to see the available options
3. Common practice is to include the name of the related component to the command name

## References

### Chapter 9

# Adding CLI commands to Joomla!

1. Core Commands are hard-coded into Joomla!
2. To add more commands we can develop a `console` plugin



## All plugins are created equal

1. Console plugins have the same structure as other plugins.
2. Console plugins are located in the folder `plugins/console`.
3. For output, we need to be aware we are not dealing with HTML anymore.

## References

### Chapter 9

## Last year Workshop

1. Clear To-Do list component ([Download it from GitHub](#))
2. We are going to create a plugin to check the deadlines for our tasks

### References

[2024 workshop](#)

## Let's start writing our plugin

In your Joomla! installation, create our folder structure:

Create the folder structure:

1. `plugins/console`
2. `plugins/console/services`
3. `plugins/console/src`
4. `plugins/console/src/Extension`
5. `plugins/console/src/CliCommand`

## References

### Chapter 9

## plugins/console/ctl/ctl.xml

```
<extension type="plugin" folder="console" method="upgrade">
...
  <namespace path="src">AwCo\Plugin\Console\ClearTodoList</namespace>
  <files>
    <folder plugin="ctl">services</folder>
    <folder>src</folder>
  </files>
</extension>
```

[https://developingextensionsforjoomla5.com/  
jdayusa2025/live/basic-plugin](https://developingextensionsforjoomla5.com/jdayusa2025/live/basic-plugin)

## References

Chapters 8 and 9

## src/CliCommands/TaskDeadlineCommand.php

```
<?php
namespace AwCo\Plugin\Console\Task\CliCommand;

class TaskDeadlineCommand extends AbstractCommand
{
    protected static $defaultName = 'ctl:deadline';

    protected function configure(): void
    {
        $this->setDescription('Check the deadlines for your tasks.');
        $this->setHelp('Execute ctl:deadline to check the deadlines for your tasks.');
    }

    protected function doExecute(InputInterface $input, OutputInterface $output): int{}
}
```

<https://developingextensionsforjoomla5.com/jdayusa2025/live/basic-plugin>

## References

Chapters 8 and 9

## src/Extension/TaskConsolePlugin.php

```
<?php

namespace AwCo\Plugin\Console\Task\Extension;

use AwCo\Plugin\Console\Task\CliCommand\TaskDeadlineCommand;

class TaskConsolePlugin extends CMSPlugin implements SubscriberInterface
{
    protected $autoloadLanguage = true;

    public static function getSubscribedEvents(): array
    {
        return [
            ApplicationEvents::BEFORE_EXECUTE => 'registerCLICommands'
        ];
    }

    public function registerCLICommands(ApplicationEvent $event): void
    {
        $app = $event->getApplication();

        $app->addCommand(new TaskDeadlineCommand());
    }
}
```

## References

Chapters 8 and 9

## services/provider.php

```
<?php
...
use AwCo\Plugin\Console\Task\Extension\TaskConsolePlugin;

return new class implements ServiceProviderInterface
{
    public function register(Container $container)
    {
        $container->set(
            PluginInterface::class,
            function (Container $container) {
                $dispatcher = $container->get(DispatcherInterface::class);
                $plugin = new TaskConsolePlugin( $dispatcher,
                    (array) PluginHelper::getPlugin('console', 'task')
                );
                return $plugin;
            }
        );
    }
};
```

## References

Chapters 8 and 9

[https://developingextensionsforjoomla5.com/  
basic-plugin](https://developingextensionsforjoomla5.com/basic-plugin)

## First install

1. We use the nice *Discover* function at Joomla! Backend
2. We enable the plugin in our backend
3. After that, we can run the command:

```
php cli/joomla.php ctl:deadline
```

## References

### Chapter 9



# Adding functionality to our plugin

File: `src/CliCommand/TaskDeadlineCommand.php`

```
class SpmTaskDeadlineCommand extends AbstractCommand
{
    ...

    protected function doExecute(InputInterface $input, OutputInterface $output): int
    {
        $outputStyle = new SymfonyStyle($input, $output);
        $outputStyle->title('Upcoming deadlines for your tasks');

        $deadlines = $this->getDeadlines();

        $this->showDeadlines($outputStyle, $deadlines);

        return Command::SUCCESS;
    }
}
```

## References

## Chapter 9

# Adding functionality to our plugin

File: `src/CliCommand/TaskDeadlineCommand.php`

```
protected function getDeadlines(): array
{
    $days = 42;

    $db = Factory::getContainer()->get(DatabaseInterface::class);
    $query = $db->createQuery();
    $query->select('*')
        ->from('#__spm_tasks');

    $query->where('deadline BETWEEN NOW() AND DATE_ADD(NOW(), INTERVAL ' . $days . ' DAY)');
    $db->setQuery($query);
    $deadlines = $db->loadAssocList('id');

    return $deadlines;
}

protected function showDeadlines($outputStyle, $deadlines)
{
    if (empty($deadlines)) {
        $outputStyle->note('There are no upcoming deadlines for your tasks.');
```

## References

## Chapter 9

# Adding options to our command

File: `src/CliCommand/TaskDeadlineCommand.php`

```
class SpmTaskDeadlineCommand extends AbstractCommand
{
    protected function configure(): void
    {
        ...

        $this->addOptions();
    }

    protected function addOptions()
    {
        $description = 'Days in the future to check for deadlines.';
        $this->addOption('days', 'd', InputOption::VALUE_OPTIONAL, $description, 7);

        return;
    }
}
```

## References

### Chapter 9

# Passing the options to our command

```
...
class SpmTaskDeadlineCommand extends AbstractCommand
{
    ...

    protected function doExecute(InputInterface $input, OutputInterface $output): int
    {
        $outputStyle = new SymfonyStyle($input, $output);
        $outputStyle->title('Upcoming deadlines for your tasks');

        $options = $this->getOptions($input);

        $deadlines = $this->getDeadlines($options);

        $this->showDeadlines($outputStyle, $deadlines);

        return Command::SUCCESS;
    }

    protected function getOptions($input): array
    {
        $options = [];

        $options = $input->getOptions();

        return $options;
    }
}
```

## References

## Chapter 9

# Passing the options to our command

```
...
class SpmTaskDeadlineCommand extends AbstractCommand
{
    ...

    protected function getDeadlines($options) : array
    {
        $deadlines = [];
        $days = (int)$options['days'];
        if ($days <= 0) {
            $days = 7;
        }
        $db = $this->getDatabase();
        $query = $db->getQuery(true);
        $query->select('id, project, title AS task')
            ->from('#__awco_tasks');
        $query->where('deadline BETWEEN NOW() AND DATE_ADD(NOW(), INTERVAL ' . $days . ' DAY)');
        $query->setQuery($query);
        $deadlines = $db->loadAssocList('id');
        return $deadlines;
    }
}
```

## References

## Chapter 9

## Ideas for commands

### Basic checks for our plugin

- Domain expiration
- SSL expiration
- Disk Usage
- Article published this week

### References

Chapter 1

## On the shoulder of giants

- Joomla Extension Development by Nicholas Dionysopoulos
  - <https://www.dionysopoulos.me/book.html>
- Joomla 4 – Developing Extensions: Step by step to an working Joomla extension
  - <https://a.co/d/1BlVa8j>
  - <https://web.archive.org/web/20230518080457/https://blog.astrid-guenther.de/en/der-weg-zu-joomla4-erweiterungen/>
- Joomla! Documentation
  - <https://manual.joomla.org>

# Thank you!

Get all the code and more articles about Joomla! extension development at:  
[DevelopingExtensionsForJoomla5.com](http://DevelopingExtensionsForJoomla5.com)