

Console Component

Symfony2

2013-02-21

#phpse

@tobiasnyholm

<http://bit.ly/consolecomponent>

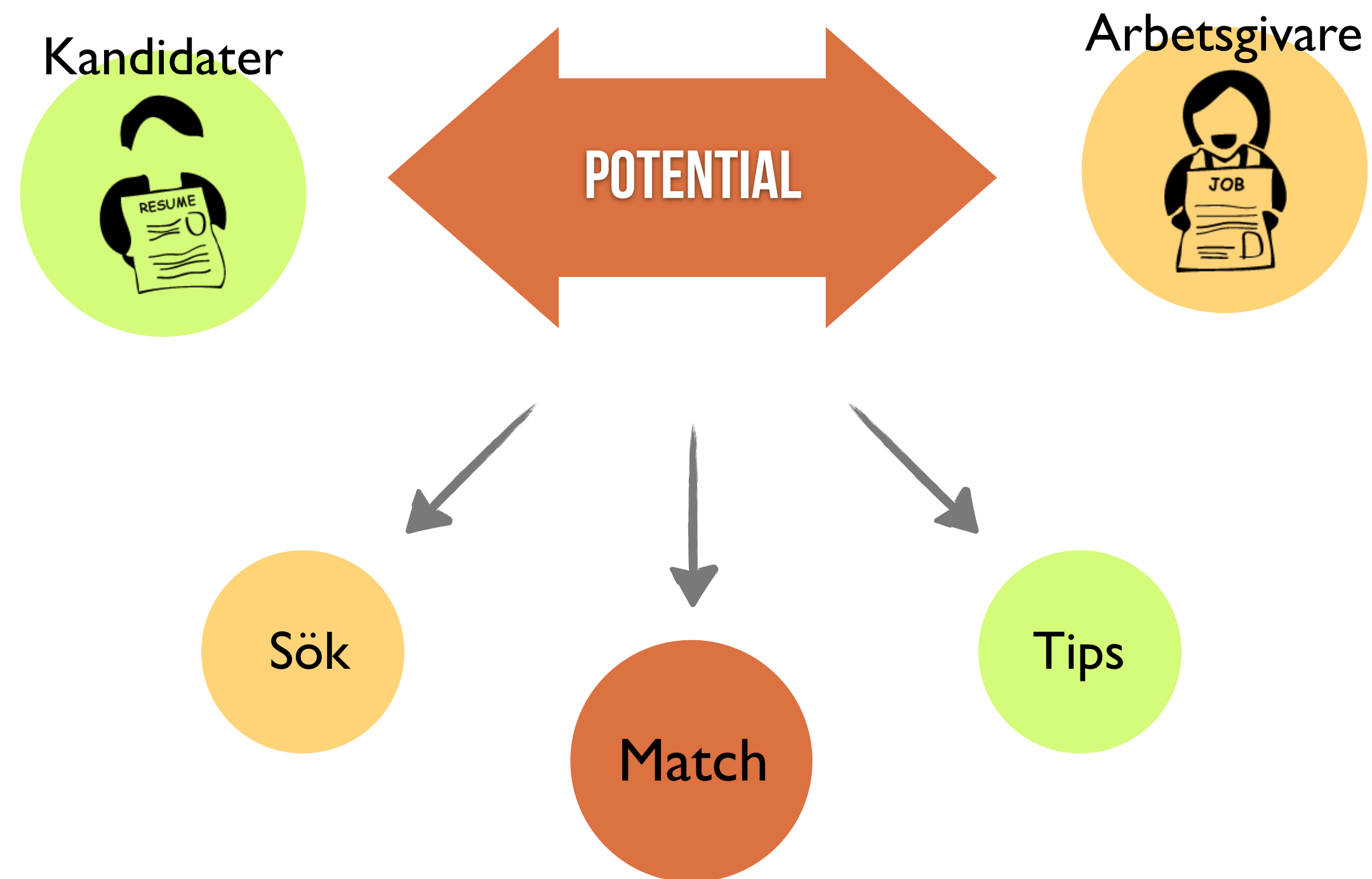
Innehåll

- **Om Tobias**
- Projektbeskrivning
- Demo
- Skriv egna kommandon
- Autocomplete
- Avslutning

Om Tobias

- Webbsidor sen 2004, eget CMS
- Wordpress & SEO
- Symfony 1.2
- WebFish.se, HappyRecruiting.se
- Continuous Integration

HappyR i ett nötskal



- ▶ molntjänst
- ▶ screenar varje kandidat
- ▶ kvalificerar och rangordnar
- ▶ sorterar cv-högen

HappyR ökar rekryteringsförmågan oavsett kompetensnivå.

Innehåll

- Om Tobias
- **Projektbeskrivning**
- Demo
- Skriv egna kommandon
- Autocomplete
- Avslutning

Projektbeskrivning - Hemnet

- House

- Street
- Area
- Price
- Agent
- Category

- Agent

- Name
- Company
- Age
- Salery

- Category

- Name

Innehåll

- Om Tobias
- Projektbeskrivning
- **Demo**
- Skriv egna kommandon
- Autocomplete
- Avslutning

Kommandon

- `php app / console generate:bundle`
- `php app / console generate:doctrine:entity`
- `php app / console generate:doctrine:entities`
- `php app / console generate:doctrine:crud`
- `php app / console doctrine:schema:update --dump-sql`
- `php app / console doctrine:schema:update --force`

Innehåll

- Om Tobias
- Projektbeskrivning
- Demo
- **Skriv egna kommandon**
- Autocomplete
- Avslutning

Skriv egna kommandon

```
use Symfony\Component\Console\Command\Command;
use Symfony\Component\Console\Input\InputArgument;
use Symfony\Component\Console\Input\InputInterface;
use Symfony\Component\Console\Input\InputOption;
use Symfony\Component\Console\Output\OutputInterface;

class HelloCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('happy:hello')
            ->setDescription('Say something');
    }

    protected function execute(InputInterface $input,
                               OutputInterface $output)
    {
        $output->writeln('Hello world');
    }
}
```

- App / AcmeBundle / Command / myCommand.php
- configure()
- execute(\$input, \$output)

Ett litet förtydligande

```
happy:command arg1 arg2 --opt2=value --opt3 --opt1=foobar
```

Skriv egna kommandon

```
protected function configure()
{
    $this
        ->setName('happy:like')
        ->setDescription('Like something')
        ->addArgument(
            'object',
            InputArgument::VALUE_REQUIRED,
            'What do you like?'
        )
        ->addOption(
            'iterations',
            'i',
            InputOption::VALUE_REQUIRED,
            'How many times should the message be printed?',
            1
        )
        ->addOption(
            'caps',
            'c',
            InputOption::VALUE_NONE,
            'Use all caps'
        );
}
```

- Argument - ordning är viktigt
- Options är alltid valfria
- Booleans eller värden

Skriv egna kommandon

```
protected function execute(InputInterface $input,
                           OutputInterface $output)
{
    $object=$input->getArgument('object');
    $iterations=$input->getOption('iterations');

    $text='I really like '.$object;
    if($input->getOption('caps')){
        $text=strtoupper($text);
    }

    for ($i = 0; $i < $iterations; $i++) {
        $output->writeln($text);
    }
}
```

- happy:like apples
- happy:like apples --iterations=3
- happy:like apples -i=3
- happy:like apples -c
- happy:like apples -i=3 -c
- happy:like apples -c -i=3

Låt kommandon köra kommandon

- Utföra en serie av beräkningar
- Ladda om databasen

Ladda om databasen

```
protected function configure() {  
    $this->setName('database:fullreload')  
        ->addOption('force',  
            null,  
            InputOption::VALUE_NONE,  
            'Has to be set to run the operation'  
        )  
        ->setDescription('Drop, create, load fixtures.');
```

- Ett kommando som sköter hela processen
- --force är valfri option

Ladda om databasen

```
protected function execute(InputInterface $input,
                           OutputInterface $output) {
    if (!$input->getOption('force')) {
        $output->writeln('<info>Notice: --force not set. Aborting.</info>');
        return;
    }
    $app = new Application();

    $progress->advance();
    $this->reloadAcl($output);
    $this->createDatabase($output);

    $this->clearUploadsFolders($output);

    //without this it will crash...
    $connection = $this->getDoctrineConnection('default');
    $connection->close();

    $this->createSchema($output);|
    $this->loadFixtures($output);

    $this->updateStatuses($output);

    $output->writeln('<info>Operation completed.</info>');
}
```

- Avbryt om --force inte är satt
- Skapa en ny ArrayInput

```
private function dropDatabase($output) {
    $command = $this->getApplication()->find('doctrine:database:drop');
    $arguments = array(
        'command' => 'doctrine:database:drop',
        '--force' => true,
    );
    $input = new ArrayInput($arguments);
    return $command->run($input, $output);
}
```


Progressbar

```
$progress = $app->getHelperSet()->get('progress');  
  
$progress->start($output, 50);  
$i = 0;  
while ($i++ < 50) {  
    // do some work  
  
    // advance the progress bar 1 unit  
    $progress->advance();  
}  
  
$progress->finish();
```

```
41/50 [=====>-----] 80%
```

- För att ge lite snygg feedback

Progressbar - Exempel

```
protected function execute(InputInterface $input,
                           OutputInterface $output) {
    if (!$input->getOption('force')) {
        $output->writeln('<info>Notice: --force not set. Aborting.</info>');
        return;
    }
    $app = new Application();
    $progress = $app->getHelperSet()->get('progress');
    $progress->start($output, 7);

    $this->dropDatabase($output);
    $progress->advance();
    $this->reloadAcl($output);
    $progress->advance();
    $this->createDatabase($output);
    $progress->advance();

    $this->clearUploadsFolders($output);
    $progress->advance();

    //without this it will crash...
    $connection = $this->getDoctrineConnection('default');
    $connection->close();
    $progress->advance();

    $this->createSchema($output);
    $progress->advance();
    $this->loadFixtures($output);
    $progress->advance();

    $this->updateStatuses($output);

    $progress->finish();
    $output->writeln('<info>Operation completed.</info>');
}
```

- Nån kanske tycker att det blev snyggare...
- ... annars skiter man i det

Innehåll

- Om Tobias
- Projektbeskrivning
- Demo
- Skriv egna kommandon
- **Autocomplete**
- Avslutning

Autocomplete

- Nytt i Symfony2.2
- Ingen riktig autocomplete
- Men mer användbart än man tror

Symfony2forum.org

- Lista med kommandon
- Symfony2 och Ajax
- Allmänna frågor

Tack för mig

Tobias Nyholm

tobias@happyrecruiting.se

www.tnyholm.se, www.webfish.se, www.happyrecruiting.se

Projektet finns på: <http://github.com/Nyholm/SymfonyTechTalk>

Presentationen finns på: <http://www.slideshare.net/TobiasNyholm>