

ЛІСТИНГ КОДУ КОМП'ЮТЕРНОЇ СИСТЕМИ

DetectFaceCommand.php

```

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

/**
 * Command line utility to detect which language some text is
 * written in.
 */
class DetectFaceCommand extends Command
{
    private $imageCreateFunc = [
        'png' => 'imagecreatefrompng',
        'gd' => 'imagecreatefromgd',
        'gif' => 'imagecreatefromgif',
        'jpg' => 'imagecreatefromjpeg',
        'jpeg' => 'imagecreatefromjpeg',
    ];

    private $imageWriteFunc = [
        'png' => 'imagepng',
        'gd' => 'imagegd',
        'gif' => 'imagegif',
        'jpg' => 'imagejpeg',
        'jpeg' => 'imagejpeg',
    ];

    protected function configure()
    {
        $this
            ->setName('face')
            ->setDescription('Detect faces in an image using '
                . 'Google Cloud Vision API')
            ->setHelp(<<<EOF
The <info>%command.name%</info> command finds faces in an image
using
the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF
                )
            ->addArgument(

```

```

        'path',
        InputArgument::REQUIRED,
        'The image to examine.'
    )
    ->addArgument(
        'output',
        InputArgument::OPTIONAL,
        'The output image with bounding boxes.'
    )
    ->addOption(
        'project',
        'p',
        InputOption::VALUE_REQUIRED,
        'Your Project ID.'
    )
    ;
}

protected function execute(InputInterface $input,
OutputInterface $output)
{
    $projectId = $input->getOption('project');
    $path = $input->getArgument('path');
    if (preg_match('/^gs:\\/\\/([a-z0-9\\._\\-]+)\\/(\S+)$/ ',
    $path, $matches)) {
        list($bucketName, $objectName) = array_slice($matches,
1);
        $result = require __DIR__ .
'/snippets/detect_face_gcs.php';
    } else {
        $result = require __DIR__ .
'/snippets/detect_face.php';
    }
    if (
        isset($result->info()['faceAnnotations'])
&& $outFile = $input->getArgument('output')
    ) {
        copy($path, $outFile);
        $ext = strtolower(pathinfo($path,
PATHINFO_EXTENSION));
        if (!in_array($ext, array_keys($this-
>imageCreateFunc))) {
            throw new \Exception('Unsupported image
extension');
        }
        $outputImage = call_user_func($this-
>imageCreateFunc[$ext], $outFile);
        # [START highlight_image]
        foreach ($result->info()['faceAnnotations'] as
$annotation) {
            if (isset($annotation['boundingPoly'])) {
                $vertices =
$annotation['boundingPoly']['vertices'];

```

```

        $x1 = isset($verticies[0]['x']) ?
$verticies[0]['x'] : 0;
        $y1 = isset($verticies[0]['y']) ?
$verticies[0]['y'] : 0;
        $x2 = isset($verticies[2]['x']) ?
$verticies[2]['x'] : 0;
        $y2 = isset($verticies[2]['y']) ?
$verticies[2]['y'] : 0;
        imagerectangle($outputImage, $x1, $y1, $x2,
$y2, 0x00ff00);
    }
}
# [END highlight_image]
call_user_func($this->imageWriteFunc[$ext],
$outputImage, $outFile);
printf('Output image written to %s' . PHP_EOL,
$outputFile);
}
}
}

```

DetectImagePropertyCommand.php

```

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

/**
 * Command line utility to detect which language some text is
 * written in.
 */
class DetectImagePropertyCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('property')
            ->setDescription('Detect image proerties in an image
using '
                . 'Google Cloud Vision API')
            ->setHelp(<<<EOF
The <info>%command.name%</info> command detects image properties
in an image
using the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF

```

```

    )
    ->addArgument(
        'path',
        InputArgument::REQUIRED,
        'The image to examine.'
    )
    ->addOption(
        'project',
        'p',
        InputOption::VALUE_REQUIRED,
        'Your Project ID.'
    )
    ;
}

protected function execute(InputInterface $input,
OutputInterface $output)
{
    $projectId = $input->getOption('project');
    $path = $input->getArgument('path');
    if (preg_match('/^gs:\\\\\/([a-z0-9\\._\\-]+)\\\\(\\S+)$/ ',
$path, $matches)) {
        list($bucketName, $objectName) = array_slice($matches,
1);
        $result = require __DIR__ .
'/snippets/detect_image_property_gcs.php';
    } else {
        $result = require __DIR__ .
'/snippets/detect_image_property.php';
    }
}
}
}

```

DetectLabelCommand.php

```

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

/**
 * Command line utility to detect which language some text is
 * written in.
 */
class DetectLabelCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('label')
    }
}

```

```

        ->setDescription('Detect labels in an image using '
            . 'Google Cloud Vision API')
        ->setHelp(<<<EOF
The <info>%command.name%</info> command labels objects seen in an
image using
the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF
        )
        ->addArgument(
            'path',
            InputArgument::REQUIRED,
            'The image to examine.'
        )
        ->addOption(
            'project',
            'p',
            InputOption::VALUE_REQUIRED,
            'Your Project ID.'
        );
    }

    protected function execute(InputInterface $input,
OutputInterface $output)
    {
        $projectId = $input->getOption('project');
        $path = $input->getArgument('path');
        if (preg_match('/^gs:\/\/\/([a-z0-9\._\-\-]+)\/(\S+)\/',
$path, $matches)) {
            list($bucketName, $objectName) = array_slice($matches,
1);
            $result = require __DIR__ .
'/snippets/detect_label_gcs.php';
        } else {
            $result = require __DIR__ .
'/snippets/detect_label.php';
        }
    }
}

```

DetectLandmarkCommand.php

```

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

```

```

/**
 * Command line utility to detect which language some text is
 * written in.
 */
class DetectLandmarkCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('landmark')
            ->setDescription('Detect landmarks in an image using '
                . 'Google Cloud Vision API')
            ->setHelp(<<<EOF
The <info>%command.name%</info> command landmarks objects seen in
an image using
the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF
            )
            ->addArgument(
                'path',
                InputArgument::REQUIRED,
                'The image to examine.'
            )
            ->addOption(
                'project',
                'p',
                InputOption::VALUE_REQUIRED,
                'Your Project ID.'
            )
        ;
    }

    protected function execute(InputInterface $input,
OutputInterface $output)
    {
        $projectId = $input->getOption('project');
        $path = $input->getArgument('path');
        if (preg_match('/^gs:\/\/\/([a-z0-9\._-]+)\/(\S+)$/ ',
$path, $matches)) {
            list($bucketName, $objectName) = array_slice($matches,
1);

            $result = require __DIR__ .
'/snippets/detect_landmark_gcs.php';
        } else {
            $result = require __DIR__ .
'/snippets/detect_landmark.php';
        }
    }
}

```

DetectLogoCommand.php

```

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

/**
 * Command line utility to detect which language some text is
 * written in.
 */
class DetectLogoCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('logo')
            ->setDescription('Detect logos in an image using '
                . 'Google Cloud Vision API')
            ->setHelp(<<<EOF
The <info>%command.name%</info> command logos objects seen in an
image using
the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF
        )
        ->addArgument(
            'path',
            InputArgument::REQUIRED,
            'The image to examine.'
        )
        ->addOption(
            'project',
            'p',
            InputOption::VALUE_REQUIRED,
            'Your Project ID.'
        )
    }

    protected function execute(InputInterface $input,
OutputInterface $output)
    {
        $projectId = $input->getOption('project');
        $path = $input->getArgument('path');
        if (preg_match('/^gs:\:\/\/([a-z0-9\._-]+)\\/(\S+)\/$',
$path, $matches)) {

```

```

        list($bucketName, $objectName) = array_slice($matches,
1);
        $result = require __DIR__ .
'/snippets/detect_logo_gcs.php';
    } else {
        $result = require __DIR__ .
'/snippets/detect_logo.php';
    }
}
}
}

```

DetectSafeSearchCommand.php

```

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

/**
 * Command line utility to detect which language some text is
written in.
 */
class DetectSafeSearchCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('safe-search')
            ->setDescription('Detect adult content in an image
using '
                . 'Google Cloud Vision API')
            ->setHelp(<<<EOF
The <info>%command.name%</info> command detects adult content in
an image using
the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF
                )
            ->addArgument(
                'path',
                InputArgument::REQUIRED,
                'The image to examine.'
            )
            ->addOption(
                'project',
                'p',
                InputOption::VALUE_REQUIRED,

```



```

        'Your Project ID.'
    )
;
}

protected function execute(InputInterface $input,
OutputInterface $output)
{
    $projectId = $input->getOption('project');
    $path = $input->getArgument('path');
    if (preg_match('/^gs:\/\/\/([a-z0-9\._\-\-]+)\/(\S+)\$/',
$path, $matches)) {
        list($bucketName, $objectName) = array_slice($matches,
1);
        $result = require __DIR__ .
'/snippets/detect_safe_search_gcs.php';
    } else {
        $result = require __DIR__ .
'/snippets/detect_safe_search.php';
    }
}
}
}

```