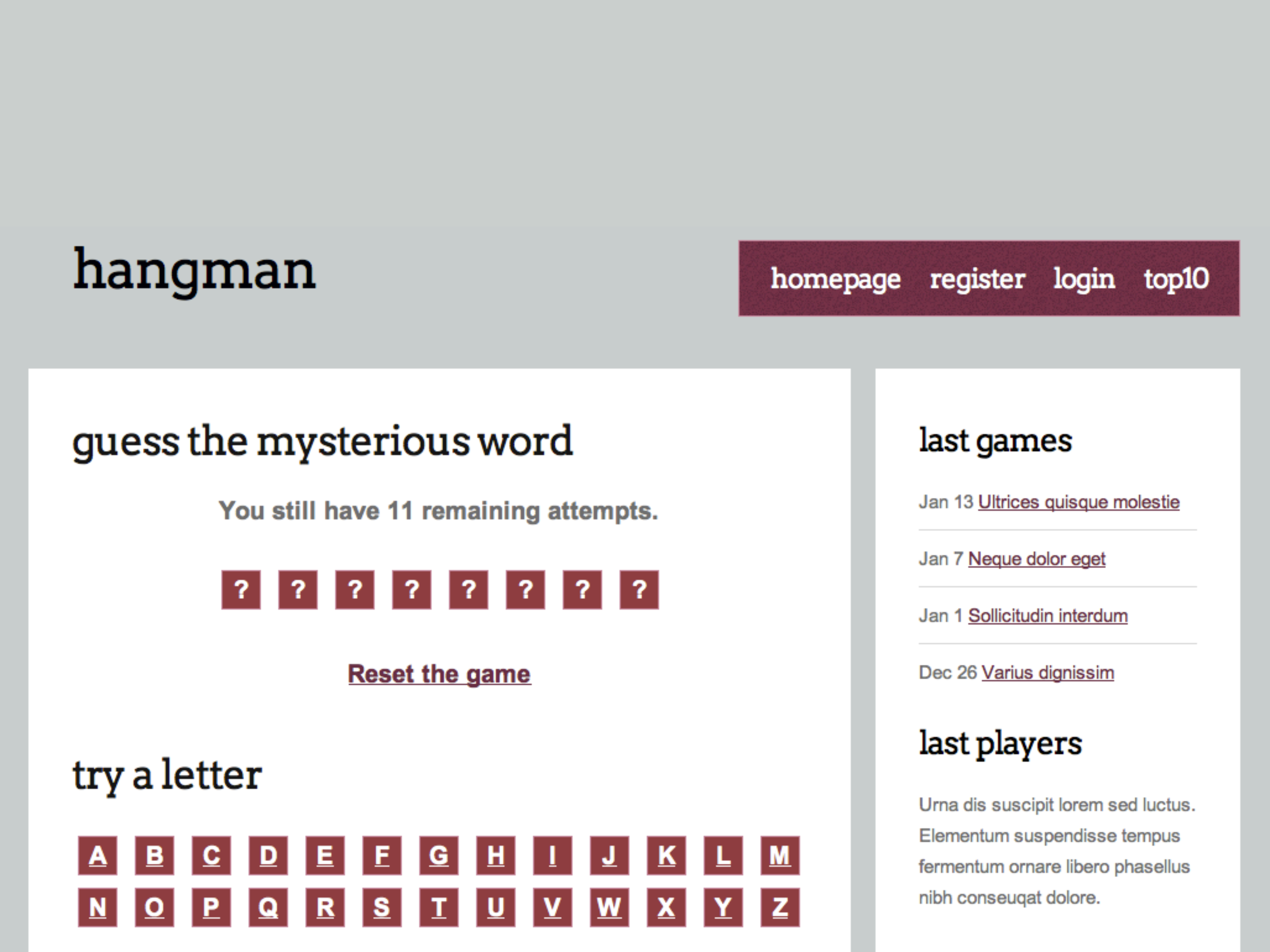
SensioLabs

Symfony

The Hangman Game

Installing the Project



hangman

guess the mysterious word

You still have 11 remaining attempts.

Reset the game

try a letter

last games

Jan 13 Ultrices quisque molestie

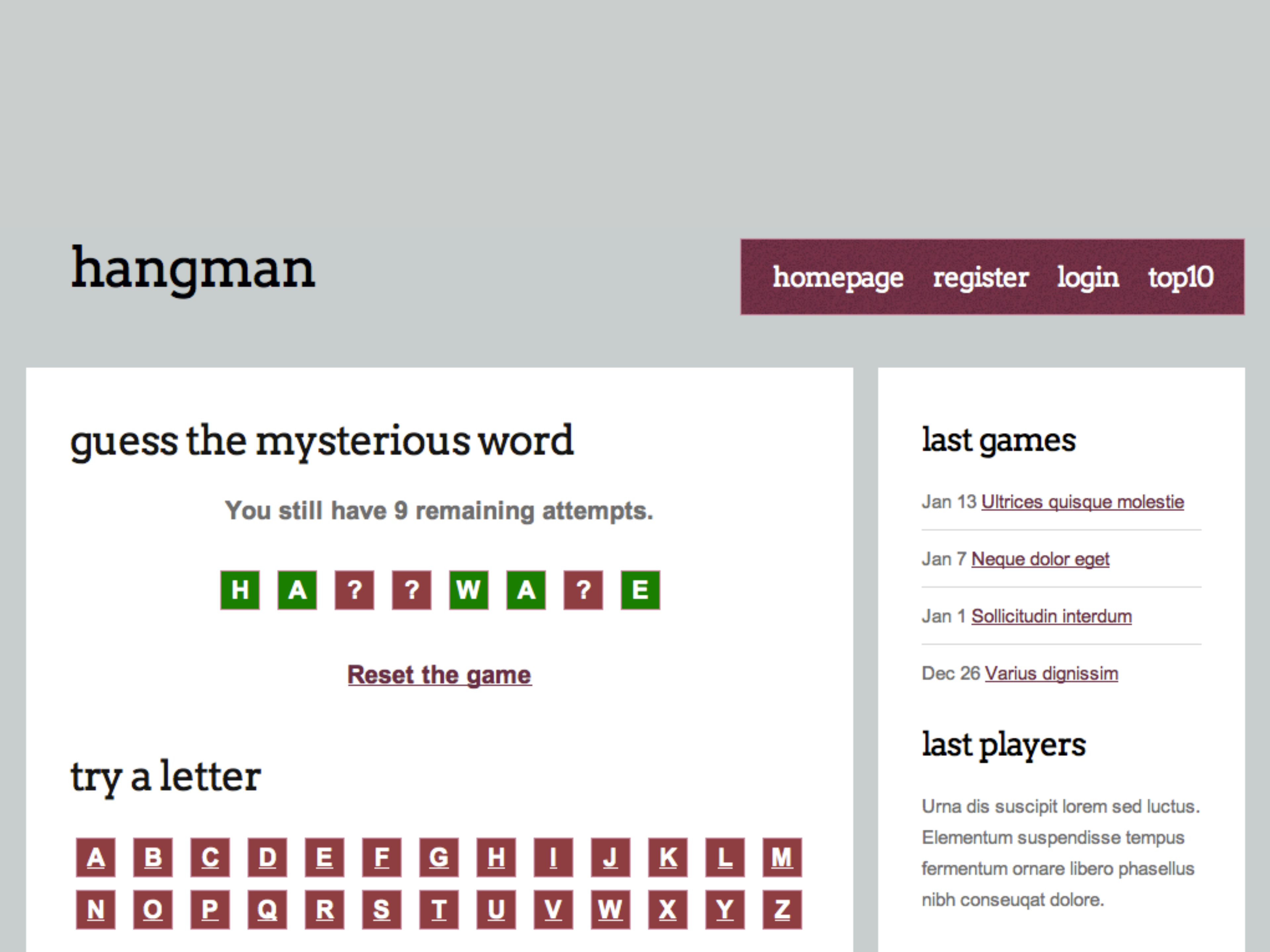
Jan 7 Neque dolor eget

Jan 1 Sollicitudin interdum

Dec 26 Varius dignissim

last players

Urna dis suscipit lorem sed luctus. Elementum suspendisse tempus fermentum ornare libero phasellus nibh conseuqat dolore.



guess the mysterious word

You still have 9 remaining attempts.

try a letter

last games

Jan 13 Ultrices quisque molestie

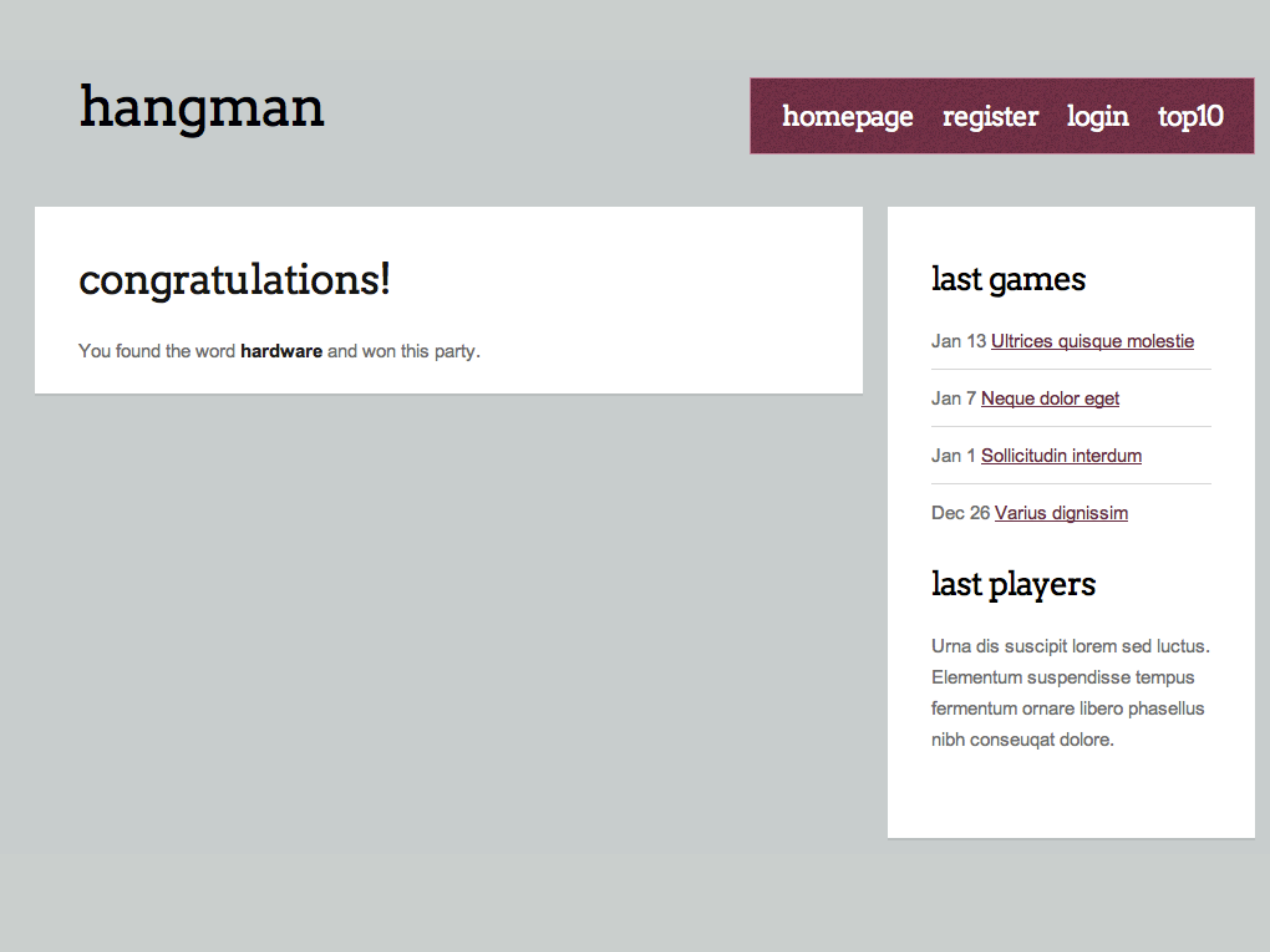
Jan 7 Neque dolor eget

Jan 1 Sollicitudin interdum

Dec 26 Varius dignissim

last players

Urna dis suscipit lorem sed luctus. Elementum suspendisse tempus fermentum ornare libero phasellus nibh conseuqat dolore.



hangman

congratulations!

You found the word **hardware** and won this party.

last games

Jan 13 Ultrices quisque molestie

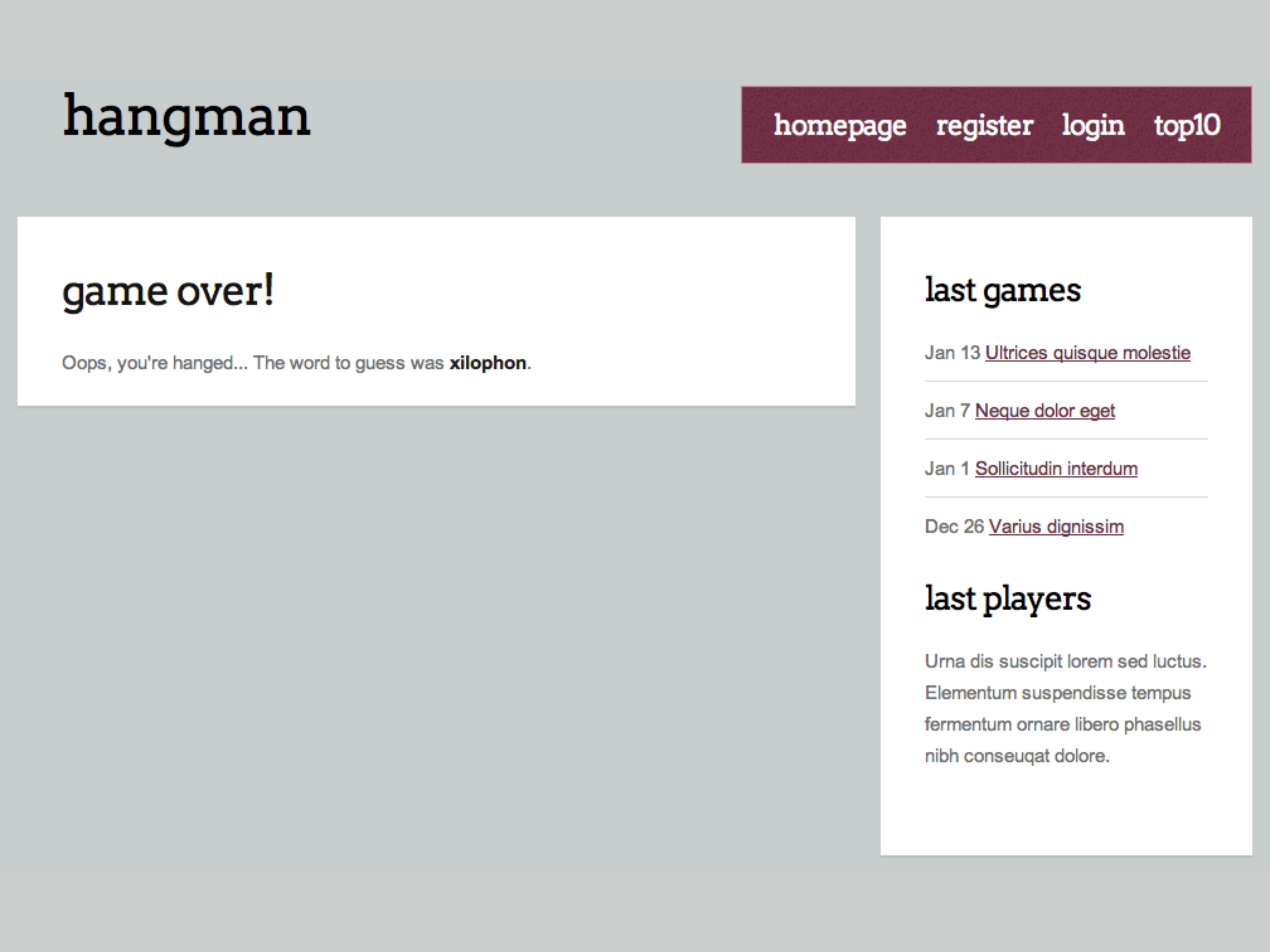
Jan 7 Neque dolor eget

Jan 1 Sollicitudin interdum

Dec 26 Varius dignissim

last players

Urna dis suscipit lorem sed luctus. Elementum suspendisse tempus fermentum ornare libero phasellus nibh conseuqat dolore.



game over!

Oops, you're hanged... The word to guess was **xilophon**.

last games

Jan 13 Ultrices quisque molestie

Jan 7 Neque dolor eget

Jan 1 Sollicitudin interdum

Dec 26 Varius dignissim

last players

Urna dis suscipit lorem sed luctus. Elementum suspendisse tempus fermentum ornare libero phasellus nibh conseuqat dolore.

Introducing
the GameContext service

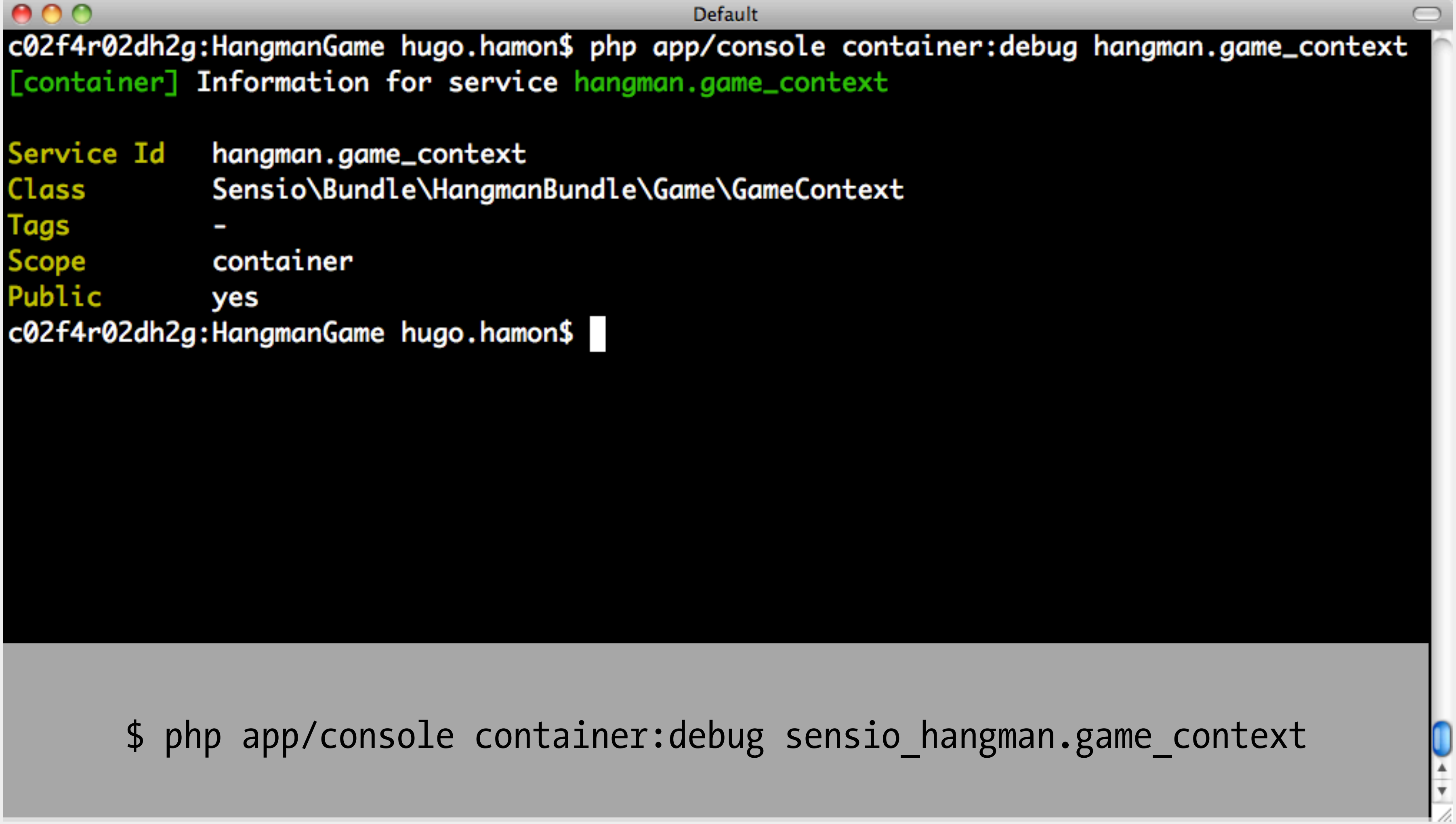
Injecting the Session in the GameContext

```php
namespace Sensio\Bundle\HangmanBundle\Game;

use Symfony\Component\HttpFoundation\Session\SessionInterface;

class GameContext
{
    private $session;

    public function __construct(SessionInterface $session)
    {
        $this->session = $session;
    }
}
```

Registering a new Game Context service

```xml
<?xml version="1.0" ?>
<container>
    <services>

        <service id="sensio_hangman.game_context"
            class="Sensio\Bundle\HangmanBundle\Game\GameContext">
            <argument type="service" id="session" />
        </service>

    </services>
</container>
```

Debugging the Container



```
c02f4r02dh2g:HangmanGame hugo.hamon$ php app/console container:debug hangman.game_context
[container] Information for service hangman.game_context

Service Id   hangman.game_context
Class        Sensio\Bundle\HangmanBundle\Game\GameContext
Tags         -
Scope        container
Public       yes
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ php app/console container:debug sensio_hangman.game_context
```

Refactoring the GameController class

```
// ...
class GameController extends Controller
{
    // ...
    private function getGameContext()
    {
        return $this->container->get('sensio_hangman.game_context');
    }
}
```

Registering the word list service and its loaders

```
<service
    id="sensio_hangman.loader.txt"
    class="Sensio\Bundle\HangmanBundle\Game\Loader\TextFileLoader"
    public="false" />

<service
    id="sensio_hangman.loader.xml"
    class="Sensio\Bundle\HangmanBundle\Game\Loader\XmlFileLoader"
    public="false" />
```

Registering the word list service and its loaders

```
<service
    id="sensio_hangman.word_list"
    class="Sensio\Bundle\HangmanBundle\Game\WordList">
    <call method="addLoader">
        <argument>txt</argument>
        <argument type="service" id="sensio_hangman.loader.txt"/>
    </call>
    <call method="addLoader">
        <argument>xml</argument>
        <argument type="service" id="sensio_hangman.loader.xml"/>
    </call>
    <call method="loadDictionaries">
        <argument>%sensio_hangman.dictionaries%</argument>
    </call>
</service>
```

Refactoring the GameController class

```
// ...
class GameController extends Controller
{
    // ...
    private function getWordList()
    {
        return $this->container->get('sensio_hangman.word_list');
    }
}
```

Using global parameters to configure classes

```
<?xml version="1.0" ?>
<container>
    <parameters>
        <parameter key="sensio_hangman.game_context.class">Sensio
\Bundle\HangmanBundle\Game\GameContext</parameter>
        <parameter key="sensio_hangman.word_list.class">Sensio
\Bundle\HangmanBundle\Game\WordList</parameter>
        <parameter key="sensio_hangman.loader.txt.class">Sensio
\Bundle\HangmanBundle\Game\Loader\TextFileLoader</parameter>
        <parameter key="sensio_hangman.loader.xml.class">Sensio
\Bundle\HangmanBundle\Game\Loader\XmlFileLoader</parameter>
    </parameters>
</container>
```

Using global parameters to configure classes

```
<?xml version="1.0" ?>
<container>
  <services>
    <service class="%sensio_hangman.game_context.class%" … >
    <service class="%sensio_hangman.word_list.class%" … >
    <service class="%sensio_hangman.loader.txt.class%" … >
    <service class="%sensio_hangman.loader.xml.class%" … >
  </services>
</container>
```

Loading the Configuration

```php
class SensioHangmanExtension extends Extension
{
    public function load(array $configs, ContainerBuilder $container)
    {
        $configuration = new Configuration();
        $config = $this->processConfiguration($configuration, $configs);

        $locator = new FileLocator(__DIR__.'/../Resources/config')
        $loader  = new Loader\XmlFileLoader($container, $locator);
        $loader->load('services.xml');
    }
}
```

« services.xml » is loaded by the bundle

Introducing a dedicated bundle configuration

```
# app/config/config.yml

imports:
    - { resource: parameters.yml }
    - { resource: security.yml }

sensio_hangman:
    word_length: 8
    dictionaries:
        - "%kernel.root_dir%/data/words.txt"
        - "%kernel.root_dir%/data/words.xml"
```

Introducing a dedicated bundle configuration

```php
class SensioHangmanExtension extends Extension
{
    public function load(array $configs, ContainerBuilder $container)
    {
        //$configuration = new Configuration();
        //$config = $this->processConfiguration($configuration, $configs);

        $config = array_pop($configs);
        if (!isset($config['dictionaries']) || !is_array($config['dictionaries'])) {
            throw new \InvalidArgumentException(...);
        }

        if (!isset($config['word_length'])) {
            throw new \InvalidArgumentException(...);
        }

        $container->setParameter('sensio_hangman.dictionaries', $config['dictionaries']);
        $container->setParameter('sensio_hangman.word_length', $config['word_length']);

        // ...
    }
}
```

Adding a semantic configuration

```php
class Configuration implements ConfigurationInterface
{
    public function getConfigTreeBuilder()
    {
        $treeBuilder = new TreeBuilder();
        $rootNode = $treeBuilder->root('sensio_hangman');

        $rootNode
            ->children()
                ->integerNode('word_length')
                    ->info('The default word length')
                    ->example('6')
                    ->min(3)
                    ->max(15)
                    ->defaultValue(8)
                ->end()
            ->end()
        ;
        return $treeBuilder;
    }
}
```

```php
class Configuration implements ConfigurationInterface
{
    public function getConfigTreeBuilder()
    {
        // ...
        $rootNode
            ->children()
                // ...
                ->arrayNode('dictionaries')
                    ->info('A list of dictionaries to load')
                    ->example('[ /path/to/words.txt, /path/to/words.xml, … ]')
                    ->isRequired()
                    ->performNoDeepMerging()
                    ->requiresAtLeastOneElement()
                    ->prototype('scalar')->end()
                ->end()
            ->end()
        ;

        return $treeBuilder;
    }
}
```

Validate dictionaries file paths

```
$rootNode
    ->children()
    // ...
        ->arrayNode('dictionaries')
        // ...
            ->prototype('scalar')
                ->validate()
                    ->ifTrue(function ($path) { return !is_readable($path); })
                    ->thenInvalid('Invalid file path %s')
                ->end()
            ->end()
        ->end()
    ->end()
;
```

Tweaking the configuration

```
class SensioHangmanExtension extends Extension
{
    public function load(array $configs, ContainerBuilder $container)
    {
        $configuration = new Configuration();
        $config = $this->processConfiguration($configuration, $configs);

        $loader = new Loader\XmlFileLoader(...);
        $loader->load('services.xml');

        $container->setParameter('sensio_hangman.word_length', $config['word_length']);

        $defintion = $container->findDefinition('sensio_hangman.word_list');
        $defintion->addMethodCall('loadDictionaries', array($config['dictionaries']));
    }
}
```

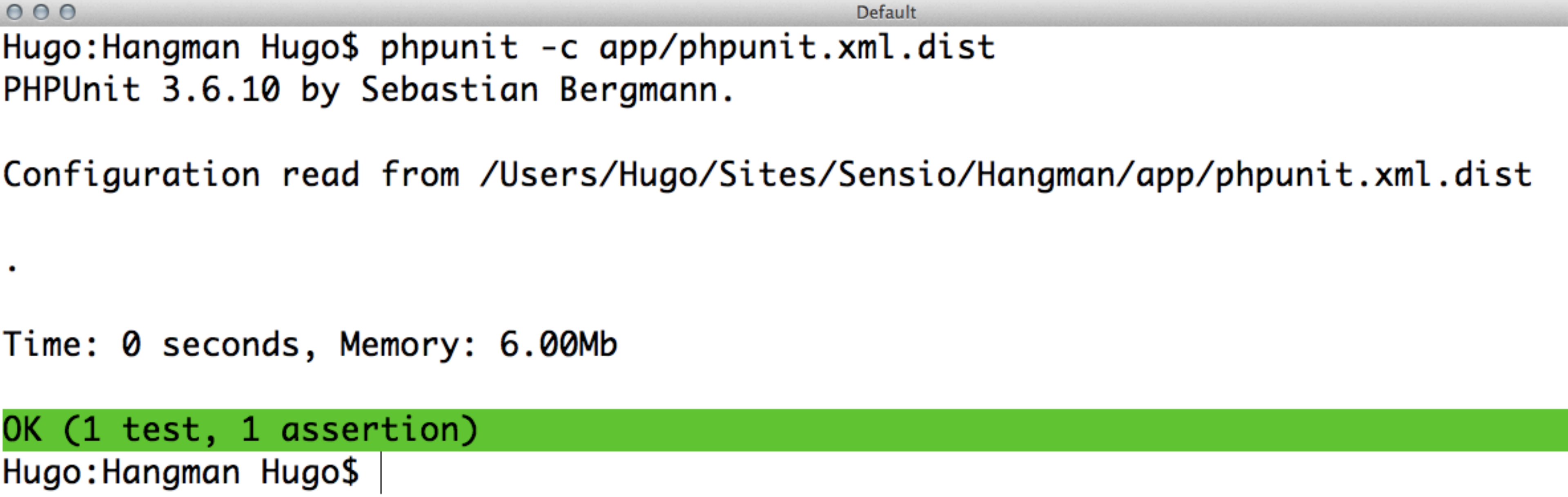
Tweaking the word list service definition

```
<service
    id="sensio_hangman.word_list"
    class="%sensio_hangman.word_list.class%">
    <call method="addLoader">
        <argument>txt</argument>
        <argument type="service" id="sensio_hangman.loader.txt"/>
    </call>
    <call method="addLoader">
        <argument>xml</argument>
        <argument type="service" id="sensio_hangman.loader.xml"/>
    </call>
</service>
```

Unit Testing

Exercise: validating the Game::tryWord() behavior

```
namespace Sensio\Bundle\HangmanBundle\Tests;

use Sensio\Bundle\HangmanBundle\Game\Game;

class GameTest extends \PHPUnit_Framework_TestCase
{
    public function testTryCorrectWord()
    {
        $game = new Game('php');
        $this->assertTrue($game->tryWord('php'));
    }
}
```

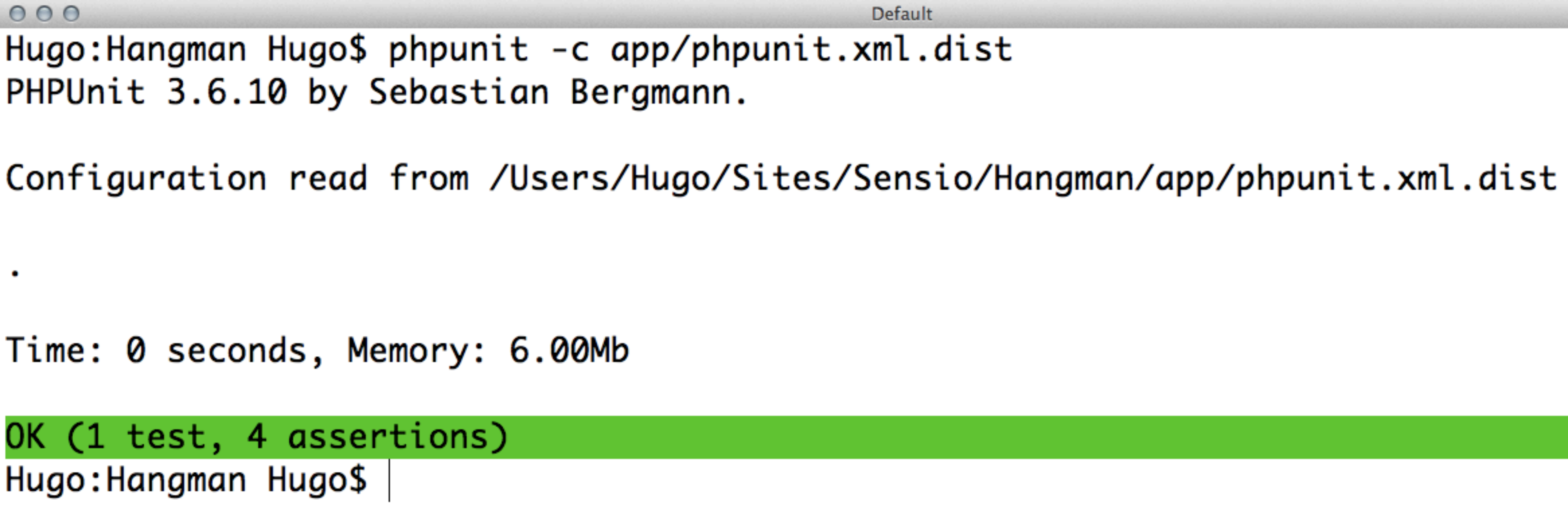


```
Hugo:Hangman Hugo$ phpunit -c app/phpunit.xml.dist
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/Hugo/Sites/Sensio/Hangman/app/phpunit.xml.dist

.

Time: 0 seconds, Memory: 6.00Mb

OK (1 test, 1 assertion)
Hugo:Hangman Hugo$
```

```
$ phpunit -c app
```

Exercise: adding extra assertions

```php
class GameTest extends \PHPUnit_Framework_TestCase
{
    public function testTryCorrectWord()
    {
        $game = new Game('php');
        $this->assertTrue($game->tryWord('php'));
        $this->assertTrue($game->isWon());
        $this->assertTrue($game->isOver());
        $this->assertFalse($game->isHanged());
    }
}
```

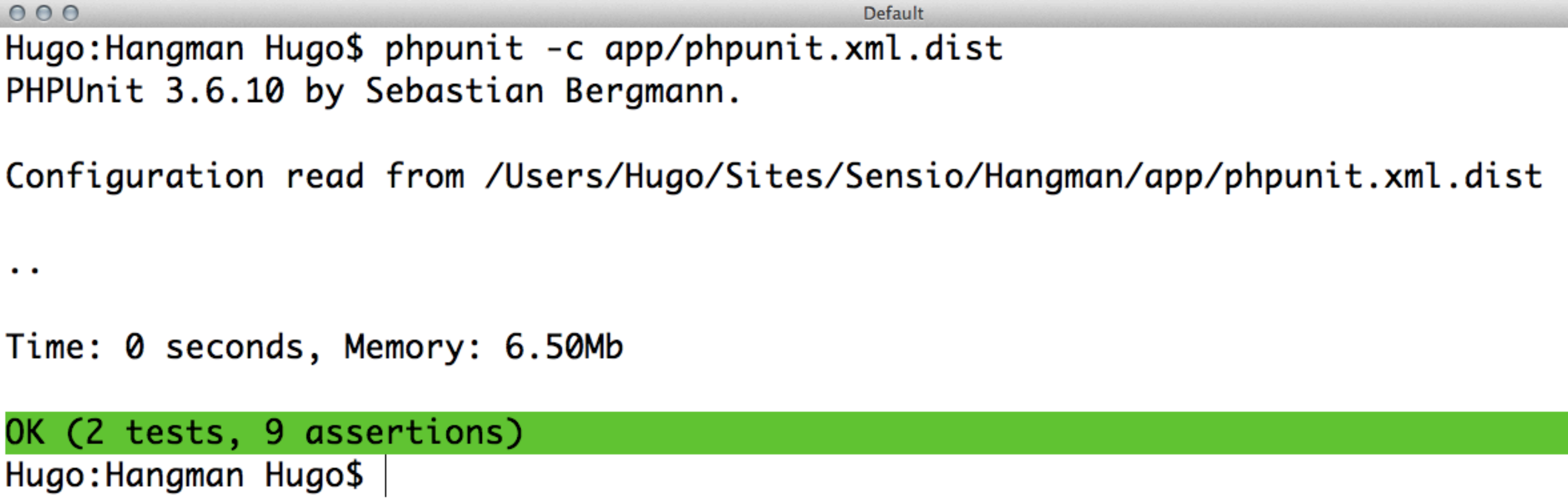


```
Hugo:Hangman Hugo$ phpunit -c app/phpunit.xml.dist
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/Hugo/Sites/Sensio/Hangman/app/phpunit.xml.dist

.

Time: 0 seconds, Memory: 6.00Mb

OK (1 test, 4 assertions)
Hugo:Hangman Hugo$
```

```
$ phpunit -c app
```

Exercise: testing Game::tryWord() with a wrong word

```php
class GameTest extends \PHPUnit_Framework_TestCase
{
    public function testTryWrongWord()
    {
        $game = new Game('php');
        $this->assertFalse($game->tryWord('foo'));
        $this->assertFalse($game->isWon());
        $this->assertTrue($game->isHanged());
        $this->assertTrue($game->isOver());
        $this->assertEquals(0, $game->getRemainingAttempts());
    }
}
```

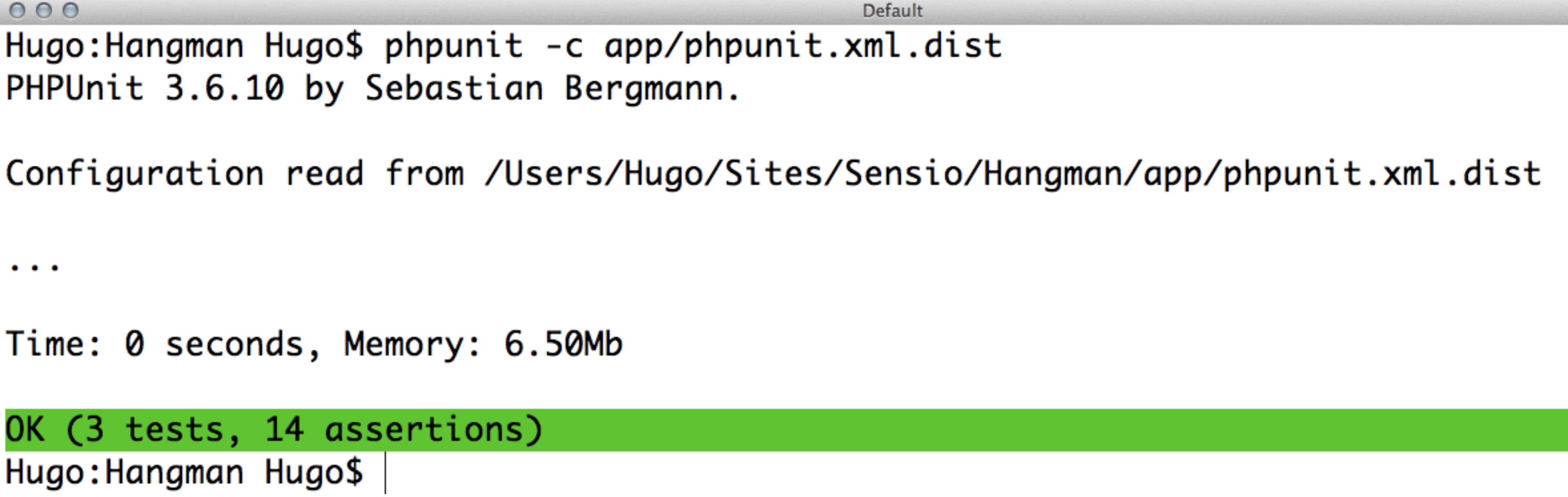


```
Hugo:Hangman Hugo$ phpunit -c app/phpunit.xml.dist
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/Hugo/Sites/Sensio/Hangman/app/phpunit.xml.dist


..

Time: 0 seconds, Memory: 6.50Mb

OK (2 tests, 9 assertions)
Hugo:Hangman Hugo$
```

```
$ phpunit -c app
```

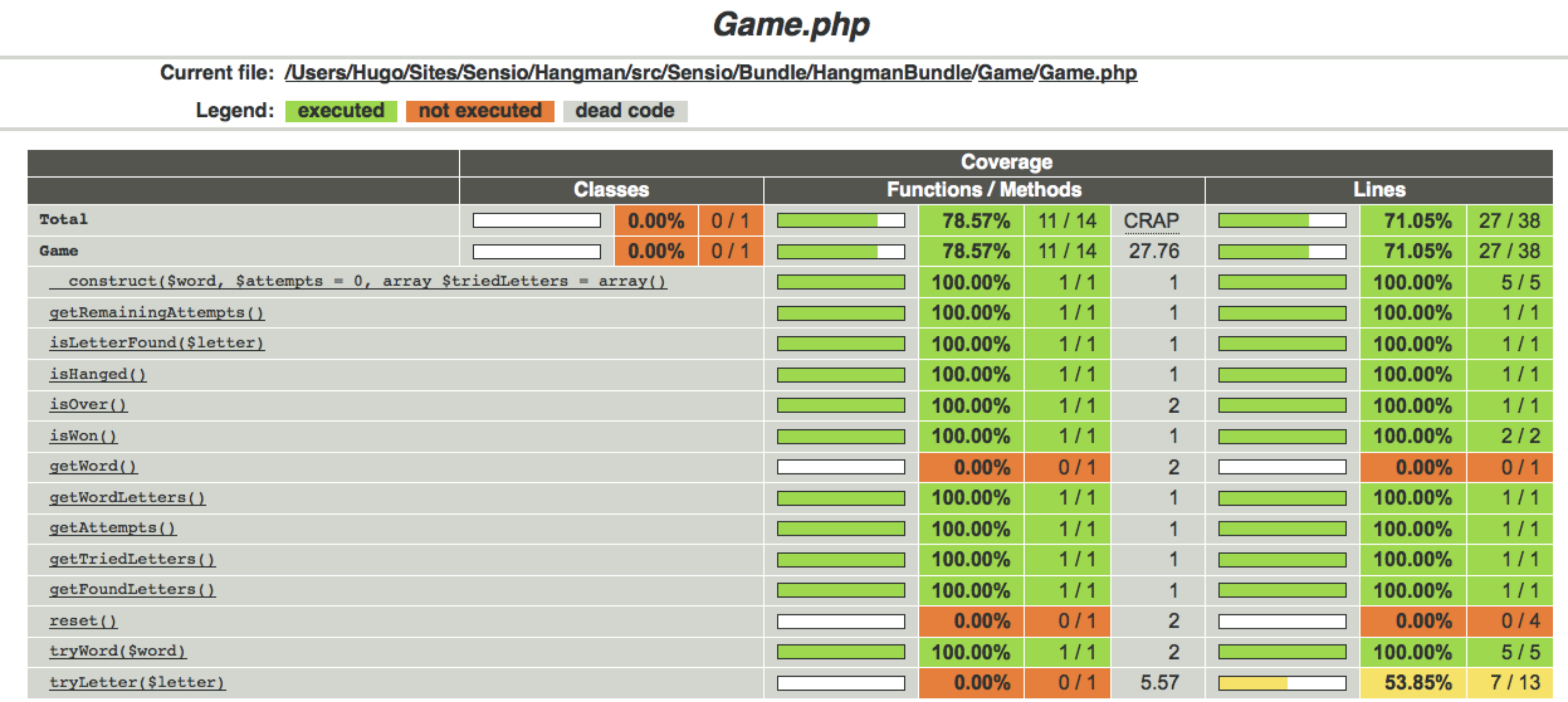
Exercise: testing Game::tryLetter() with a valid letter

```
class GameTest extends \PHPUnit_Framework_TestCase
{
    // ...

    public function testTryCorrectLetter()
    {
        $game = new Game('php');
        $this->assertTrue($game->tryLetter('P'));
        $this->assertTrue($game->isLetterFound('P'));
        $this->assertContains('p', $game->getFoundLetters());
        $this->assertContains('p', $game->getTriedLetters());
        $this->assertEquals(0, $game->getAttempts());
    }
}
```



```
Hugo:Hangman Hugo$ phpunit -c app/phpunit.xml.dist
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/Hugo/Sites/Sensio/Hangman/app/phpunit.xml.dist


...

Time: 0 seconds, Memory: 6.50Mb

OK (3 tests, 14 assertions)
Hugo:Hangman Hugo$
```

```
$ phpunit -c app
```



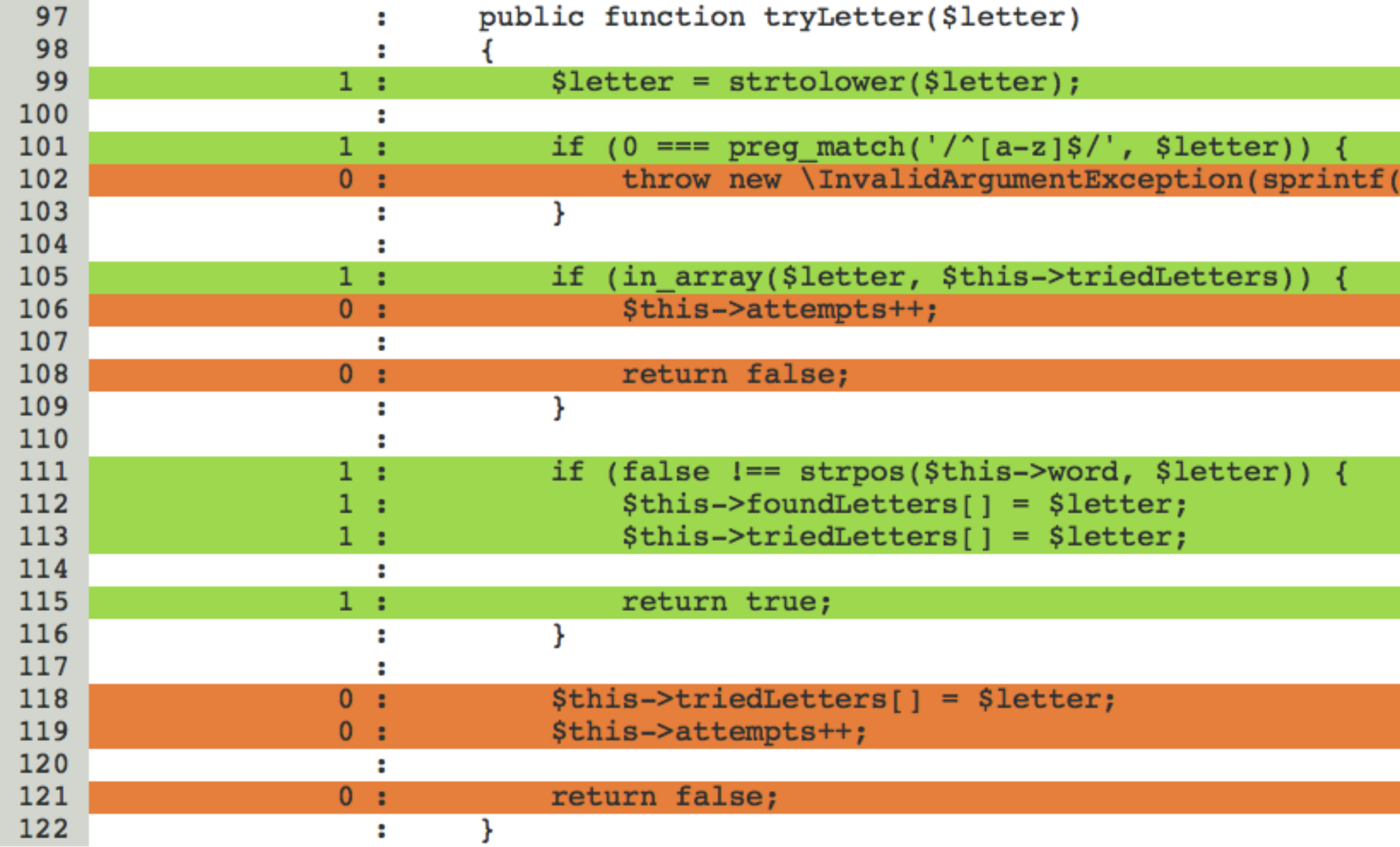
Game.php

Current file: /Users/Hugo/Sites/Sensio/Hangman/src/Sensio/Bundle/HangmanBundle/Game/Game.php

Legend: executed not executed dead code

	Coverage									
	Classes			Functions / Methods				Lines		
Total		0.00%	0 / 1		78.57%	11 / 14	CRAP		71.05%	27 / 38
Game		0.00%	0 / 1		78.57%	11 / 14	27.76		71.05%	27 / 38
__construct($word, $attempts = 0, array $triedLetters = array())					100.00%	1 / 1	1		100.00%	5 / 5
getRemainingAttempts()					100.00%	1 / 1	1		100.00%	1 / 1
isLetterFound($letter)					100.00%	1 / 1	1		100.00%	1 / 1
isHanged()					100.00%	1 / 1	1		100.00%	1 / 1
isOver()					100.00%	1 / 1	2		100.00%	1 / 1
isWon()					100.00%	1 / 1	1		100.00%	2 / 2
getWord()					0.00%	0 / 1	2		0.00%	0 / 1
getWordLetters()					100.00%	1 / 1	1		100.00%	1 / 1
getAttempts()					100.00%	1 / 1	1		100.00%	1 / 1
getTriedLetters()					100.00%	1 / 1	1		100.00%	1 / 1
getFoundLetters()					100.00%	1 / 1	1		100.00%	1 / 1
reset()					0.00%	0 / 1	2		0.00%	0 / 4
tryWord($word)					100.00%	1 / 1	2		100.00%	5 / 5
tryLetter($letter)					0.00%	0 / 1	5.57		53.85%	7 / 13

Game class is covered up to 71%

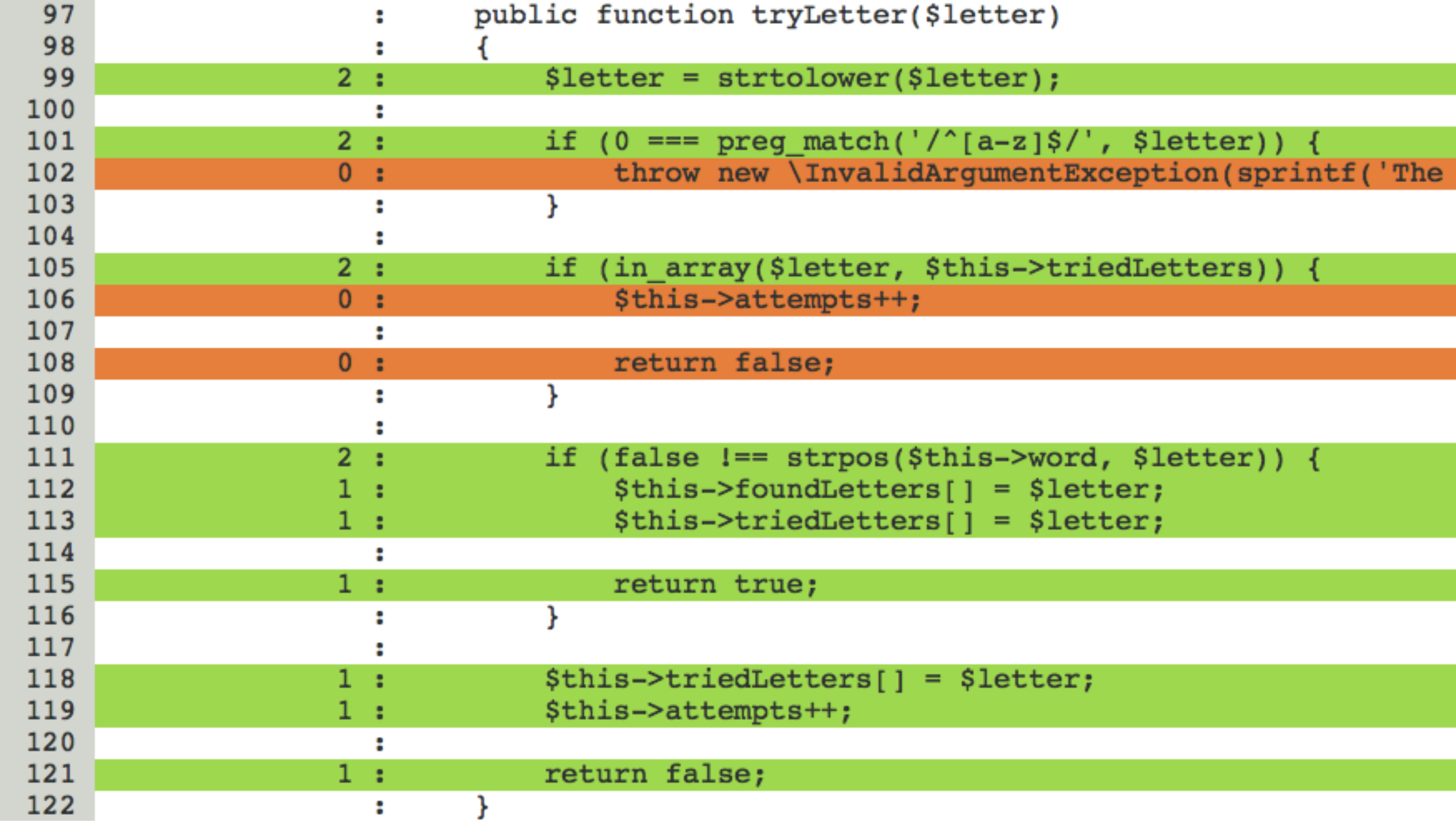


```
    :     public function tryLetter($letter)
    :     {
  1 :         $letter = strtolower($letter);
    :
  1 :         if (0 === preg_match('/^[a-z]$/', $letter)) {
  0 :             throw new \InvalidArgumentException(sprintf(
    :         }
    :
  1 :         if (in_array($letter, $this->triedLetters)) {
  0 :             $this->attempts++;
    :
  0 :             return false;
    :         }
    :
  1 :         if (false !== strpos($this->word, $letter)) {
  1 :             $this->foundLetters[] = $letter;
  1 :             $this->triedLetters[] = $letter;
    :
  1 :             return true;
    :         }
    :
  0 :         $this->triedLetters[] = $letter;
  0 :         $this->attempts++;
    :
  0 :         return false;
    :     }
```

Game::tryLetter method is not yet fully covered!

Exercise: testing Game::tryLetter() with a wrong letter

```php
class GameTest extends \PHPUnit_Framework_TestCase
{
    // ...

    public function testTryWrongLetter()
    {
        $game = new Game('php');
        $this->assertFalse($game->tryLetter('X'));
        $this->assertFalse($game->isLetterFound('X'));
        $this->assertNotContains('x', $game->getFoundLetters());
        $this->assertContains('x', $game->getTriedLetters());
        $this->assertSame(1, $game->getAttempts());
    }
}
```



```
  :     public function tryLetter($letter)
  :     {
2 :         $letter = strtolower($letter);
  :
2 :         if (0 === preg_match('/^[a-z]$/', $letter)) {
0 :             throw new \InvalidArgumentException(sprintf('The
  :         }
  :
2 :         if (in_array($letter, $this->triedLetters)) {
0 :             $this->attempts++;
  :
0 :             return false;
  :         }
  :
2 :         if (false !== strpos($this->word, $letter)) {
1 :             $this->foundLetters[] = $letter;
1 :             $this->triedLetters[] = $letter;
  :
1 :             return true;
  :         }
  :
1 :         $this->triedLetters[] = $letter;
1 :         $this->attempts++;
  :
1 :         return false;
  :     }
```

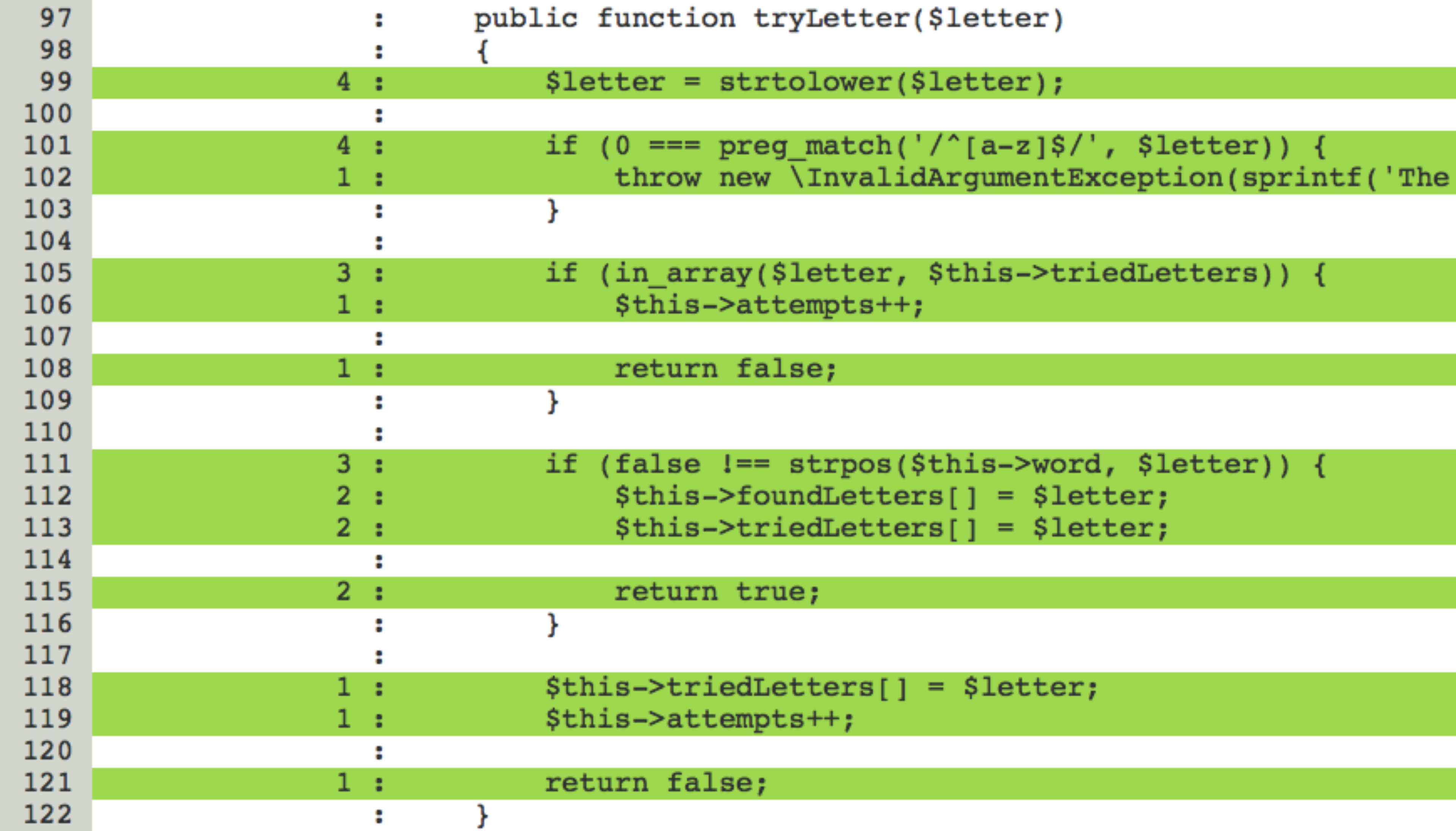
Game::tryLetter method is almost fully covered!

Exercise: testing Game::tryLetter() exception

```
class GameTest extends \PHPUnit_Framework_TestCase
{
    // ...

    public function testTryNumber()
    {
        $this->setExpectedException('InvalidArgumentException');

        $game = new Game('php');
        $game->tryLetter(10);
    }
}
```

Exercise: testing same letter twice in a row

```php
class GameTest extends \PHPUnit_Framework_TestCase
{
    // ...

    public function testTryLetterTwice()
    {
        $game = new Game('php');
        $this->assertTrue($game->tryLetter('P'));
        $this->assertFalse($game->tryLetter('P'));
    }
}
```



```
    :    public function tryLetter($letter)
    :    {
  4 :        $letter = strtolower($letter);
    :
  4 :        if (0 === preg_match('/^[a-z]$/', $letter)) {
  1 :            throw new \InvalidArgumentException(sprintf('The
    :        }
    :
  3 :        if (in_array($letter, $this->triedLetters)) {
  1 :            $this->attempts++;
    :
  1 :            return false;
    :        }
    :
  3 :        if (false !== strpos($this->word, $letter)) {
  2 :            $this->foundLetters[] = $letter;
  2 :            $this->triedLetters[] = $letter;
    :
  2 :            return true;
    :        }
    :
  1 :        $this->triedLetters[] = $letter;
  1 :        $this->attempts++;
    :
  1 :        return false;
    :    }
```

Game::tryLetter method is now fully covered!

Exercise: using a data provider

```php
class GameTest extends \PHPUnit_Framework_TestCase
{
    // ...

    public function provideWords()
    {
        return array(
            array('php'),
            array('java'),
            array('aircraft'),
            array('software'),
        );
    }
}
```

Exercise: using a data provider

```
class GameTest extends \PHPUnit_Framework_TestCase
{
    // ...

    /**
     * @dataProvider provideWords
     */
    public function testTryCorrectWord($word)
    {
        $game = new Game($word);
        $this->assertTrue($game->tryWord($word));
        // ...
    }
}
```

Exercise: Testing game is loaded

```
class GameContextTest extends \PHPUnit_Framework_TestCase
{
    public function testLoadTheGame()
    {
        // ... Mock for session object

        $context = new GameContext($session);

        $this->assertInstanceOf(
            'Sensio\Bundle\HangmanBundle\Game\Game',
            $context->loadGame()
        );
    }
}
```

```
class GameContextTest extends \PHPUnit_Framework_TestCase
{
    public function testLoadTheGame()
    {
        $data = array(
            'word'            => 'php',
            'attempts'        => 1,
            'tried_letters'   => array('P', 'X'),
            'found_letters'   => array('P'),
        );

        // ...
    }
}
```

```php
public function testLoadTheGame()
{
    // ...
    $session = $this
        ->getMock('Symfony\Component\HttpFoundation\Session\SessionInterface')
    ;

    $session
      ->expects($this->once())
      ->method('get')
      ->with($this->equalTo('hangman'))
      ->will($this->returnValue($data))
    ;

    // ...
}
```

Exercise: Testing game is not loaded

```php
class GameContextTest extends \PHPUnit_Framework_TestCase
{
    public function testCantLoadTheGame()
    {
        // ... Mock for session object

        $context = new GameContext($session);

        $this->assertFalse($context->loadGame());
    }
}
```

```php
public function testLoadTheGame()
{
    $session = $this
      ->getMock('Symfony\Component\HttpFoundation\Session\SessionInterface')
    ;

    $session
      ->expects($this->once())
      ->method('get')
      ->with($this->equalTo('hangman'))
      ->will($this->returnValue(null))
    ;

    // ...
}
```

Exercise: Testing Game is saved

```php
class GameContextTest extends \PHPUnit_Framework_TestCase
{
    public function testSaveTheGame()
    {
        // ... Mocks for session and game objects

        $context = new GameContext($session);

        $this->assertNull($context->save($game));
    }
}
```

```php
class GameContextTest extends \PHPUnit_Framework_TestCase
{
    public function testSaveTheGame()
    {
        $data = array(
            'word'          => 'php',
            'attempts'      => 1,
            'tried_letters' => array('P', 'X'),
            'found_letters' => array('P'),
        );

        // ...
    }
}
```

```php
public function testSaveTheGame()
{
    // ...
    $session = $this
      ->getMock('Symfony\Component\HttpFoundation\Session\SessionInterface')
    ;

    $session
      ->expects($this->once())
      ->method('set')
      ->with($this->equalTo('hangman'), $this->equalTo($data))
    ;

    // ...
}
```

```
public function testSaveTheGame()
{
    // ...
    $game = $this
        ->getMockBuilder('Sensio\Bundle\HangmanBundle\Game\Game')
        ->setMethods(array('getContext'))
        ->setConstructorArgs(array_values($data))
        ->getMock()
    ;

    $game
        ->expects($this->once())
        ->method('getContext')
        ->will($this->returnValue($data))
    ;

        // ...
    }
```

Functional Testing

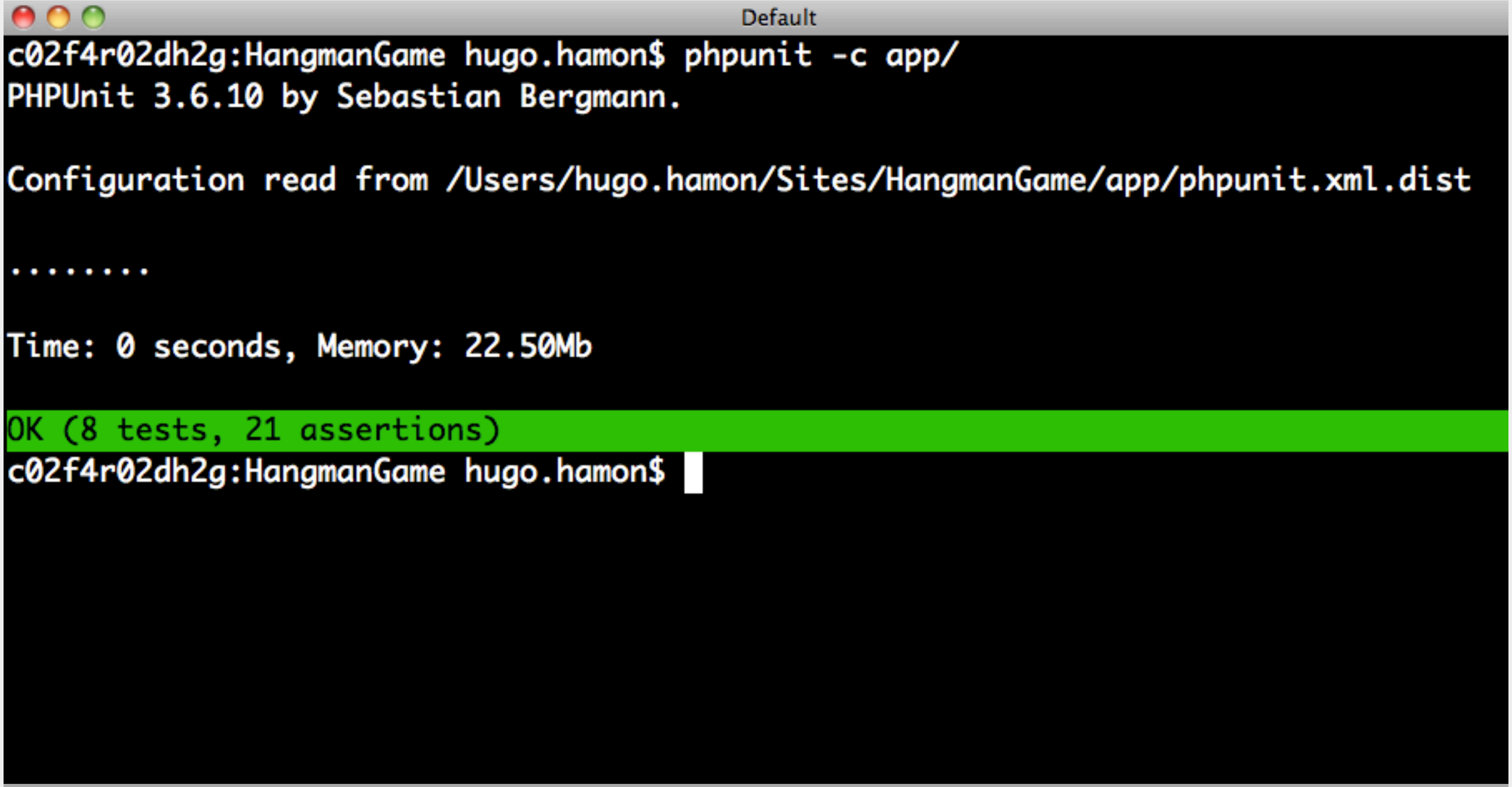
Setting up the environment

```
# app/config/config_test.yml

sensio_hangman:
    word_length: 3
    dictionaries:
        - "%kernel.root_dir%/data/test.txt"
```

```php
namespace Sensio\Bundle\HangmanBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class GameControllerTest extends WebTestCase
{
    private $client;

    public function setUp()
    {
        $this->client = static::createClient();
        $this->client->followRedirects(true);
    }

    public function tearDown()
    {
        $this->client = null;
    }
}
```

Scenario: filling the form with the valid word

```
class GameControllerTest extends WebTestCase
{
    // ...
    public function testTryWord()
    {
        $crawler = $this->client->request('GET', '/game/');
        $form = $crawler->selectButton('Let me guess...')->form();
        $crawler = $this->client->submit($form, array('word' => 'php'));

        // Check the game is won
        $this->assertSame(
            'Congratulations!',
            $crawler->filter('#content > h2:first-child')->text()
        );
    }
}
```

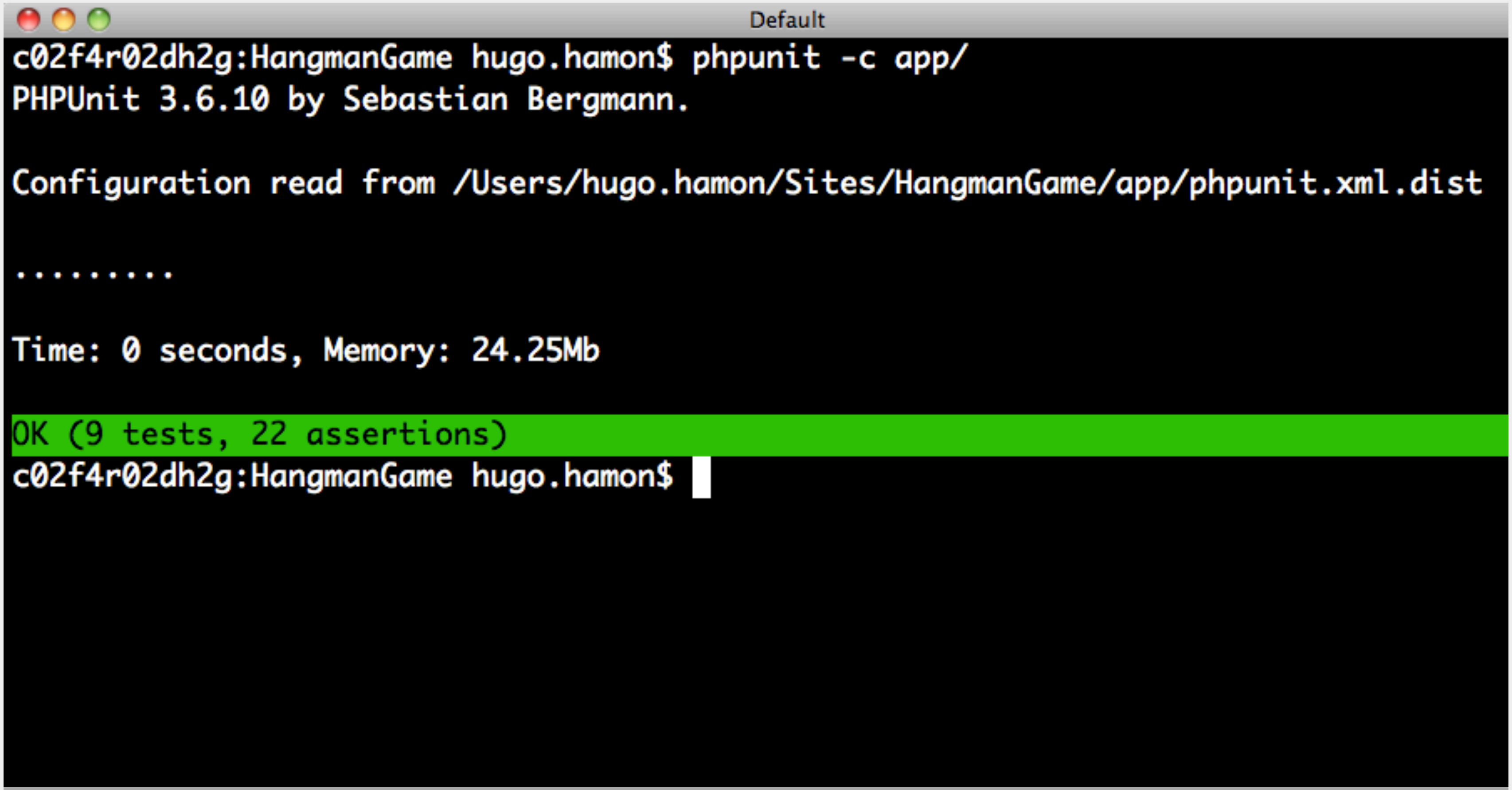


```
c02f4r02dh2g:HangmanGame hugo.hamon$ phpunit -c app/
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/hugo.hamon/Sites/HangmanGame/app/phpunit.xml.dist

........

Time: 0 seconds, Memory: 22.50Mb

OK (8 tests, 21 assertions)
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ phpunit -c app
```

Scenario: game is over with invalid word

```
class GameControllerTest extends WebTestCase
{
    // ...
    public function testGameOverHanged()
    {
        $crawler = $this->client->request('GET', '/game/');
        $form    = $crawler->selectButton('Let me guess...')->form();
        $crawler = $this->client->submit($form, array('word' => 'foo'));

        $this->assertSame(
            'Game Over!',
            $crawler->filter('#content > h2:first-child')->text()
        );
    }
}
```



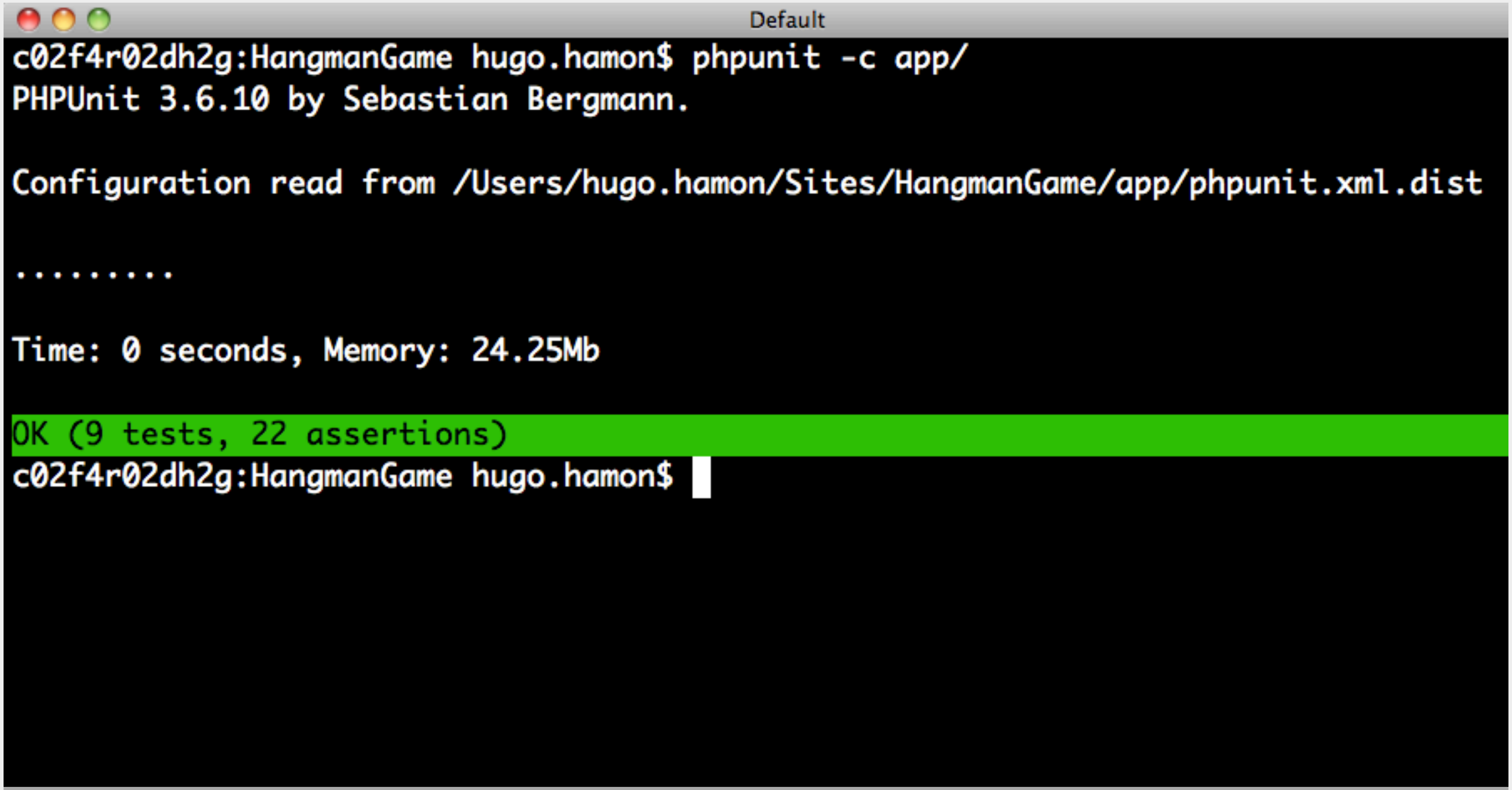
```
c02f4r02dh2g:HangmanGame hugo.hamon$ phpunit -c app/
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/hugo.hamon/Sites/HangmanGame/app/phpunit.xml.dist

.........

Time: 0 seconds, Memory: 24.25Mb

OK (9 tests, 22 assertions)
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ phpunit -c app
```

Refactoring the two scenarii

```php
class GameControllerTest extends WebTestCase
{
    // ...

    private function playWord($word)
    {
        $crawler = $this->client->getCrawler();

        $form = $crawler->selectButton('Let me guess...')->form();

        return $this->client->submit($form, array('word' => $word));
    }
}
```

```php
class GameControllerTest extends WebTestCase
{
    // ...
    public function testTryWord()
    {
        $this->client->request('GET', '/game/');

        $crawler = $this->playWord('php');

        // ...
    }

    public function testGameOverHanged()
    {
        $this->client->request('GET', '/game/');

        $crawler = $this->playWord('foo');

        // ...
    }
}
```



```
c02f4r02dh2g:HangmanGame hugo.hamon$ phpunit -c app/
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/hugo.hamon/Sites/HangmanGame/app/phpunit.xml.dist

.........

Time: 0 seconds, Memory: 24.25Mb

OK (9 tests, 22 assertions)
c02f4r02dh2g:HangmanGame hugo.hamon$
```

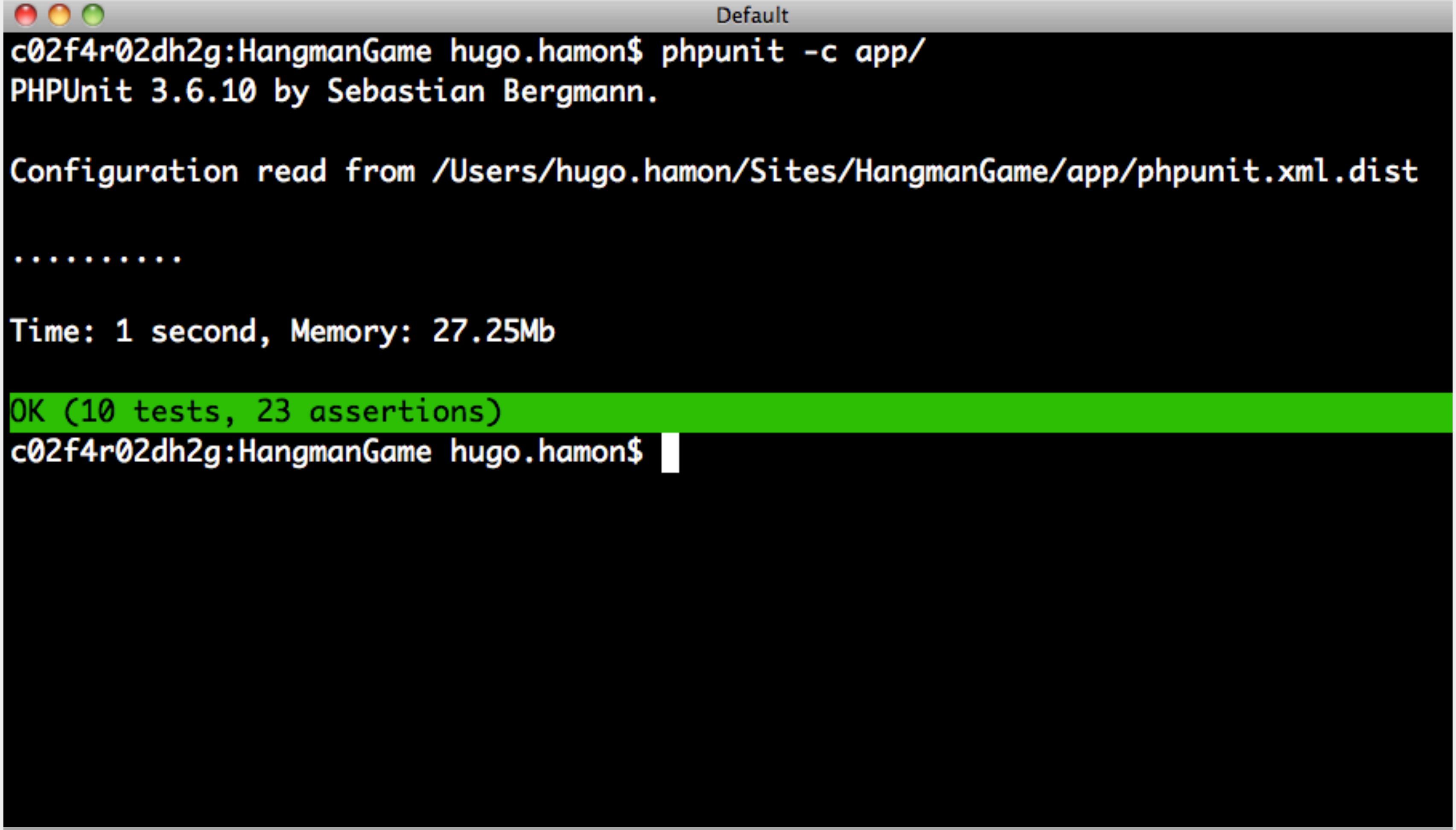
Test suite still passes!

Scenario: guessing the word by clicking letters

```php
class GameControllerTest extends WebTestCase
{
    // ...
    public function testGuessWord()
    {
        $this->client->request('GET', '/game/');

        foreach (array('H', 'X', 'P') as $letter) {
            $crawler = $this->playLetter($letter);
        }

        $this->assertSame(
            'Congratulations!',
            $crawler->filter('#content > h2:first-child')->text()
        );
    }
}
```

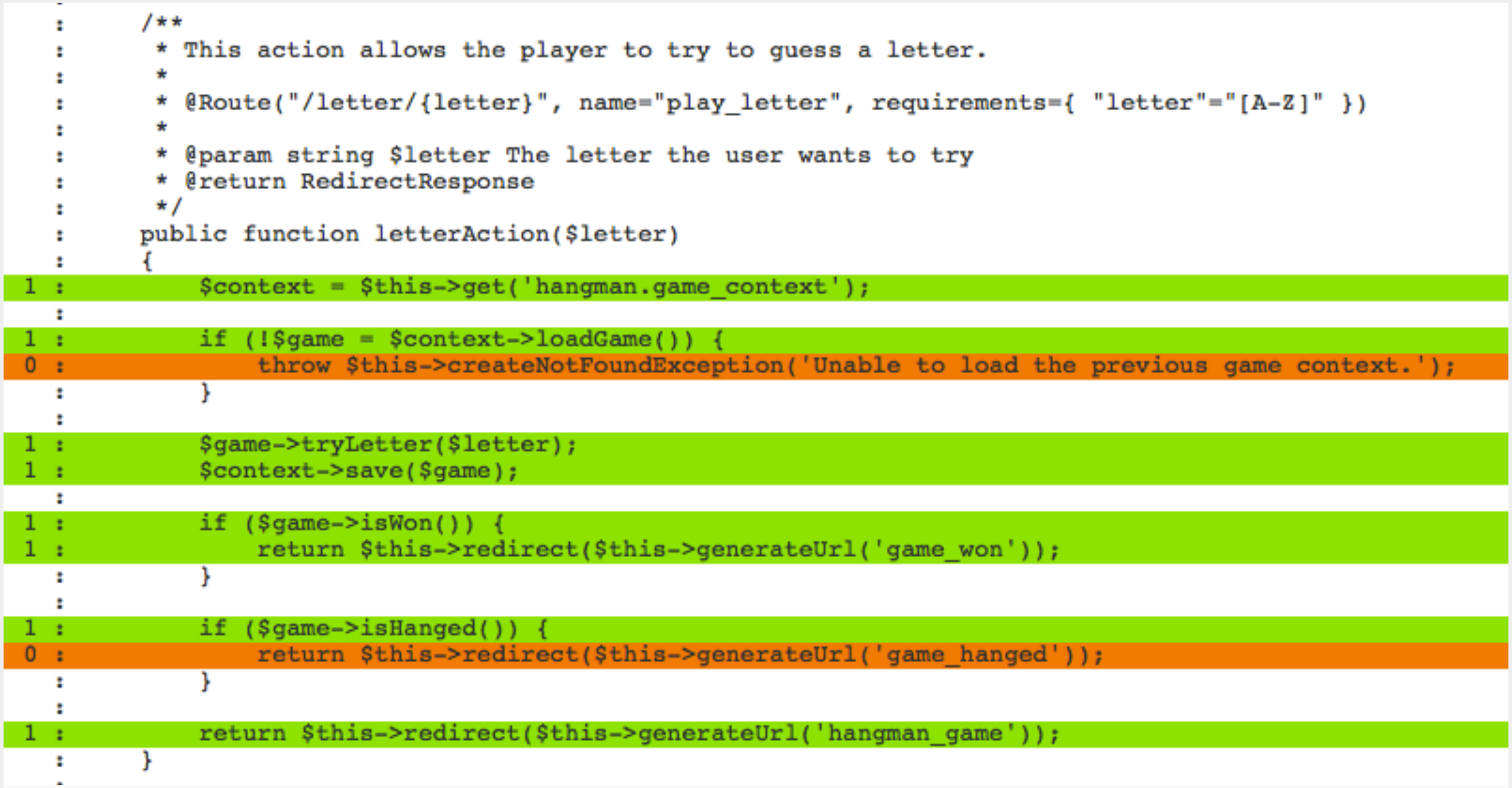
Scenario: guessing the word by clicking letters

```
class GameControllerTest extends WebTestCase
{
    // ...

    private function playLetter($letter)
    {
        $crawler = $this->client->getCrawler();

        $link = $crawler->selectLink($letter)->link();

        return $this->client->click($link);
    }
}
```



```
c02f4r02dh2g:HangmanGame hugo.hamon$ phpunit -c app/
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/hugo.hamon/Sites/HangmanGame/app/phpunit.xml.dist


..........

Time: 1 second, Memory: 27.25Mb

OK (10 tests, 23 assertions)
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ phpunit -c app/
```

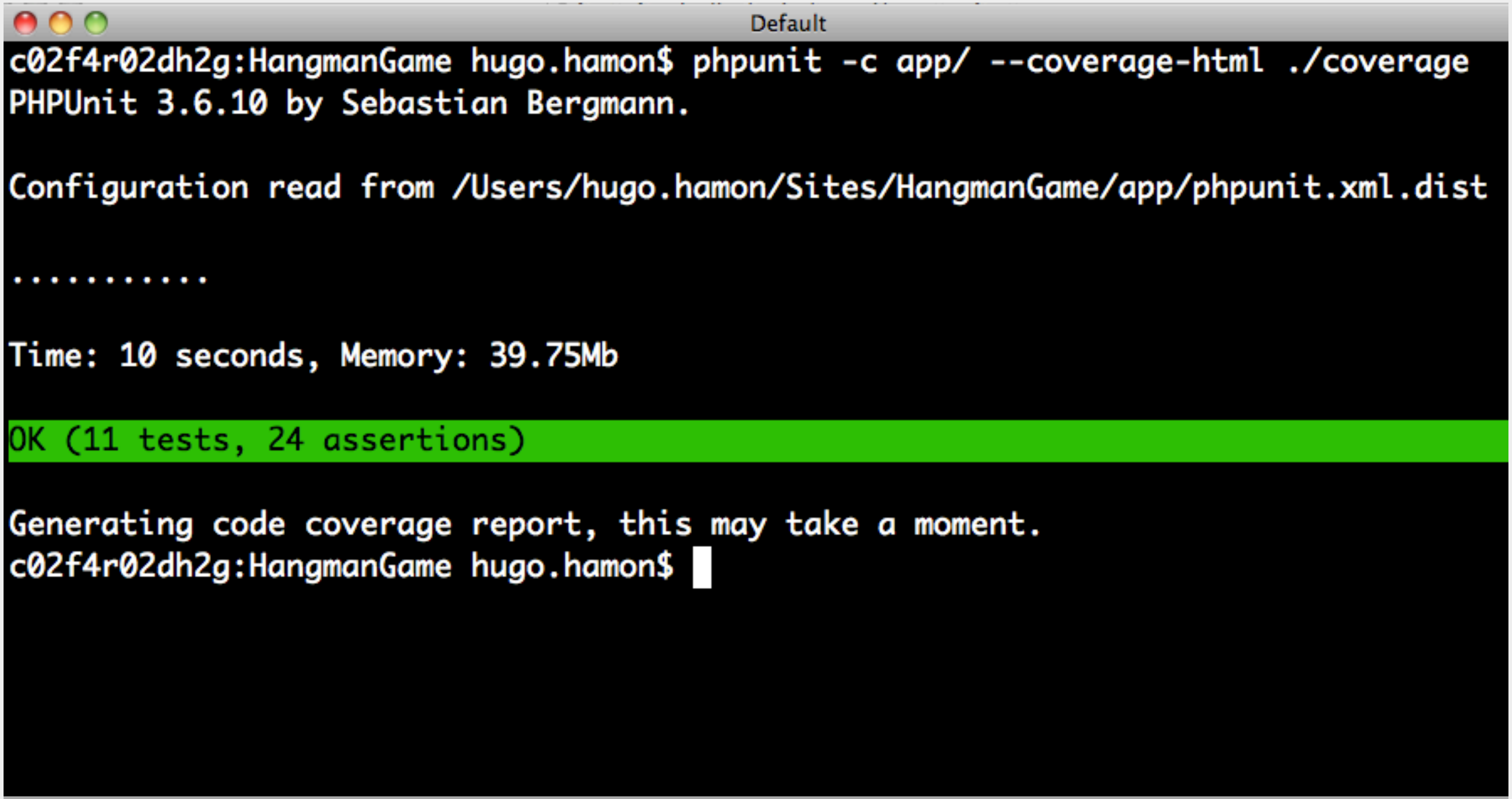


```
  :     /**
  :      * This action allows the player to try to guess a letter.
  :      *
  :      * @Route("/letter/{letter}", name="play_letter", requirements={ "letter"="[A-Z]" })
  :      *
  :      * @param string $letter The letter the user wants to try
  :      * @return RedirectResponse
  :      */
  :     public function letterAction($letter)
  :     {
1 :         $context = $this->get('hangman.game_context');
  :
1 :         if (!$game = $context->loadGame()) {
0 :             throw $this->createNotFoundException('Unable to load the previous game context.');
  :         }
  :
1 :         $game->tryLetter($letter);
1 :         $context->save($game);
  :
1 :         if ($game->isWon()) {
1 :             return $this->redirect($this->generateUrl('game_won'));
  :         }
  :
1 :         if ($game->isHanged()) {
0 :             return $this->redirect($this->generateUrl('game_hanged'));
  :         }
  :
1 :         return $this->redirect($this->generateUrl('hangman_game'));
  :     }
```

The letter action still needs to be covered!

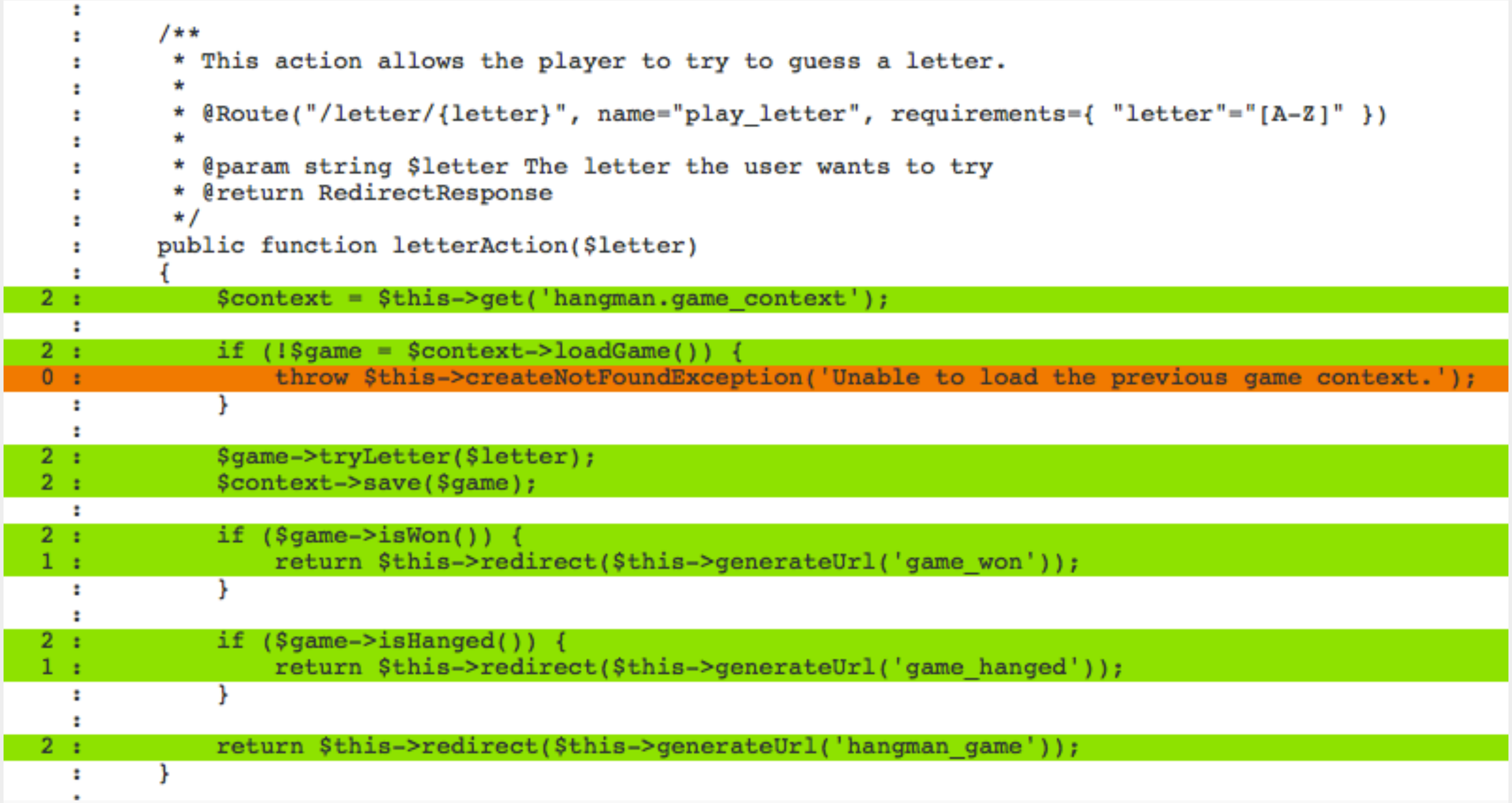
Scenario: game over after max attempts

```
use Sensio\Bundle\HangmanBundle\Game\Game;

class GameControllerTest extends WebTestCase
{
    // ...
    public function testGuessLetterAndGetHanged()
    {
        $this->client->request('GET', '/game/');

        // Play the same letter until being hanged
        for ($i = 1; $i <= Game::MAX_ATTEMPTS; $i++) {
            $this->playLetter('Z');
        }

        $crawler = $this->client->getCrawler();
        $this->assertSame(
            'Game Over!',
            $crawler->filter('#content > h2:first-child')->text()
        );
    }
}
```



```
c02f4r02dh2g:HangmanGame hugo.hamon$ phpunit -c app/ --coverage-html ./coverage
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/hugo.hamon/Sites/HangmanGame/app/phpunit.xml.dist


...........

Time: 10 seconds, Memory: 39.75Mb

OK (11 tests, 24 assertions)

Generating code coverage report, this may take a moment.
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ phpunit -c app/ --coverage-html ./coverage
```

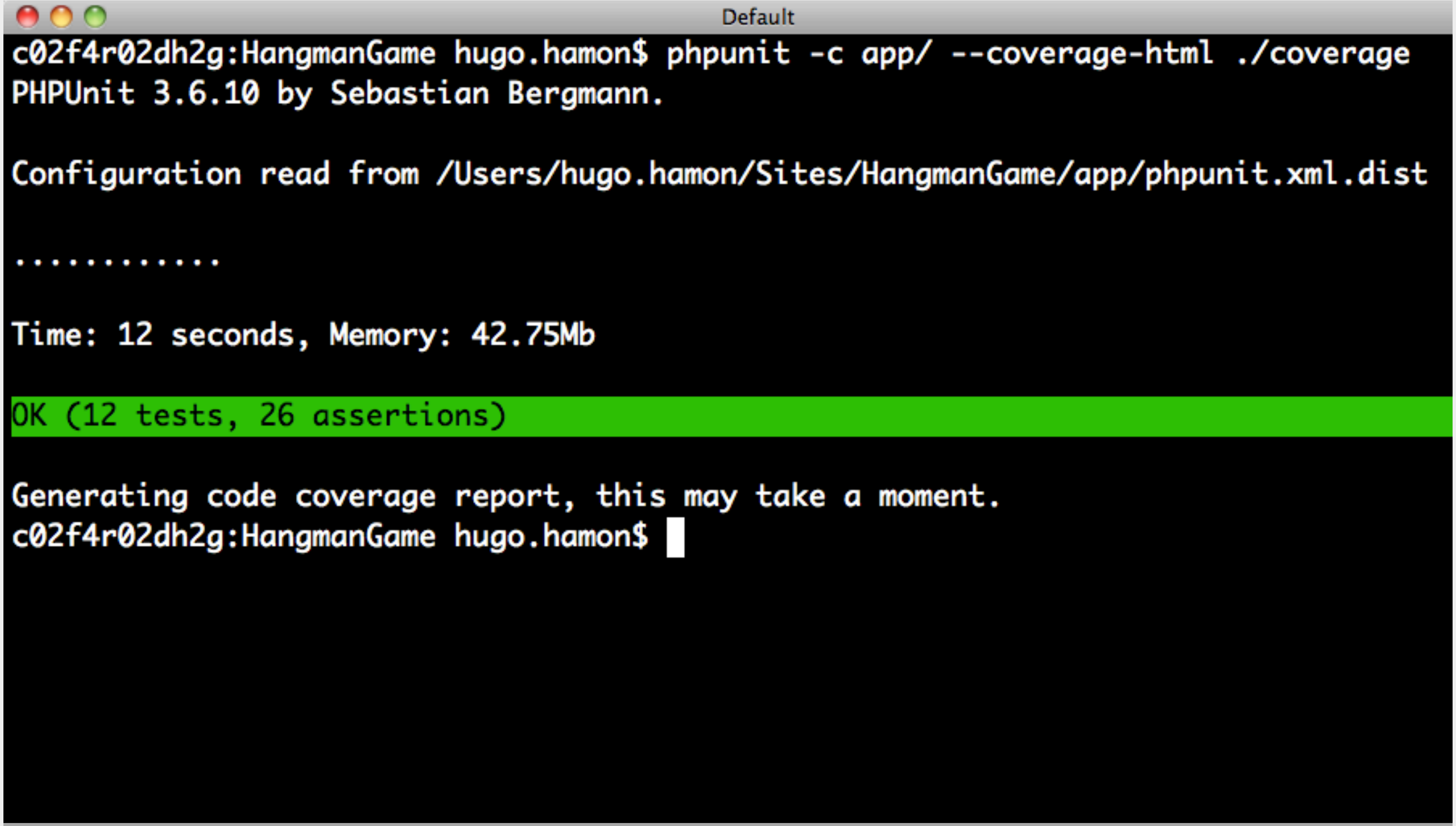


```
  :
  :     /**
  :      * This action allows the player to try to guess a letter.
  :      *
  :      * @Route("/letter/{letter}", name="play_letter", requirements={ "letter"="[A-Z]" })
  :      *
  :      * @param string $letter The letter the user wants to try
  :      * @return RedirectResponse
  :      */
  :     public function letterAction($letter)
  :     {
2 :         $context = $this->get('hangman.game_context');
  :
2 :         if (!$game = $context->loadGame()) {
0 :             throw $this->createNotFoundException('Unable to load the previous game context.');
  :         }
  :
2 :         $game->tryLetter($letter);
2 :         $context->save($game);
  :
2 :         if ($game->isWon()) {
1 :             return $this->redirect($this->generateUrl('game_won'));
  :         }
  :
2 :         if ($game->isHanged()) {
1 :             return $this->redirect($this->generateUrl('game_hanged'));
  :         }
  :
2 :         return $this->redirect($this->generateUrl('hangman_game'));
  :     }
  :
```

The letter action is almost fully covered!

Scenario: reseting an already started game

```php
class GameControllerTest extends WebTestCase
{
    // ...
    public function testResetGame()
    {
        $this->client->request('GET', '/game/');
        $crawler = $this->playLetter('P');

        // Click the reset link
        $link = $crawler->selectLink('Reset the game')->link();
        $crawler = $this->client->click($link);

        // Check the initial state
        $this->assertCount(0, $crawler->filter('.word_letters .guessed'));
        $this->assertCount(3, $crawler->filter('.word_letters .hidden'));
    }
}
```

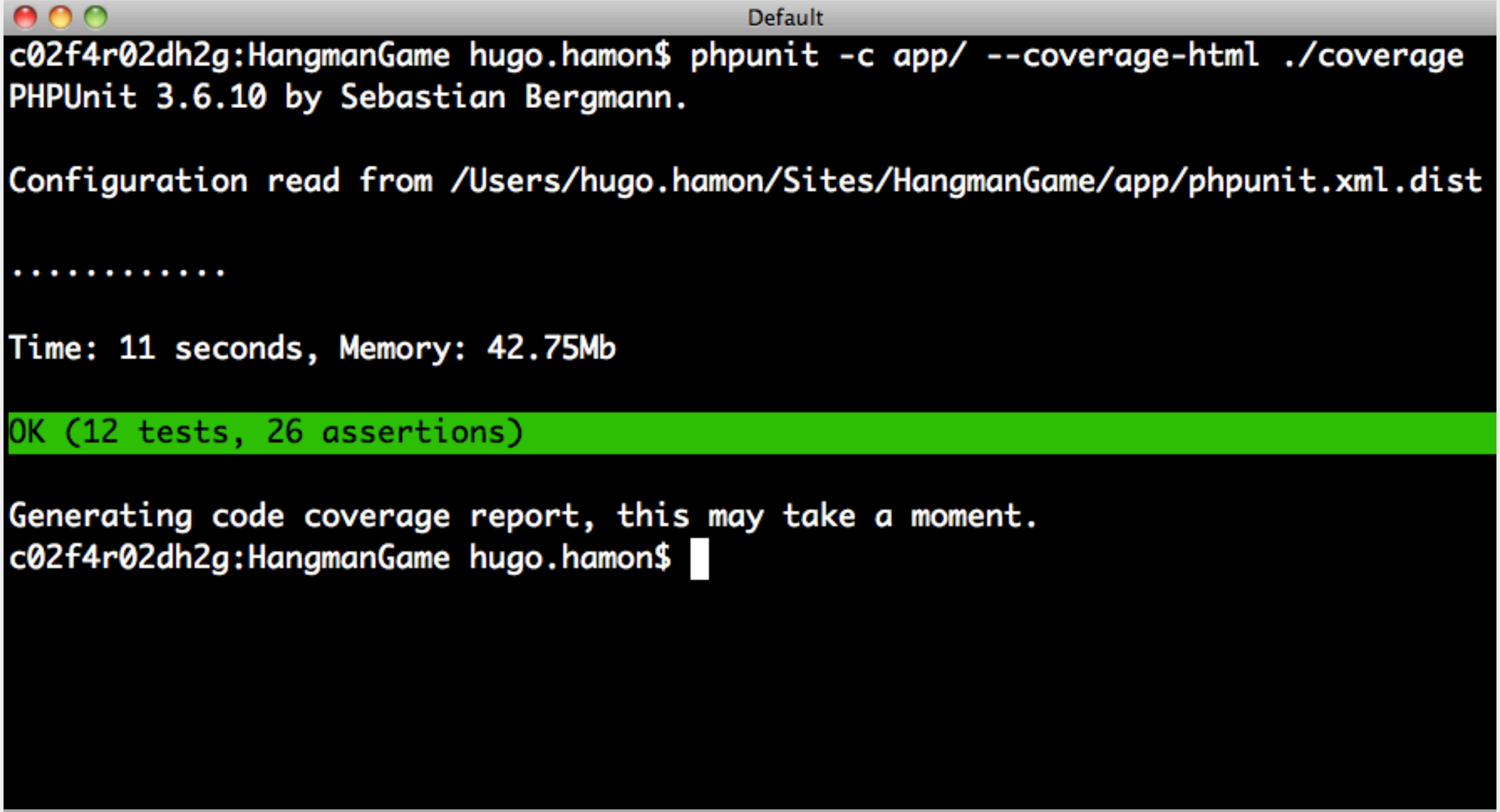


```
c02f4r02dh2g:HangmanGame hugo.hamon$ phpunit -c app/ --coverage-html ./coverage
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/hugo.hamon/Sites/HangmanGame/app/phpunit.xml.dist

............

Time: 12 seconds, Memory: 42.75Mb

OK (12 tests, 26 assertions)

Generating code coverage report, this may take a moment.
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ phpunit -c app/ --coverage-html ./coverage
```

Scenarii: playing without a started game

```php
class GameControllerTest extends WebTestCase
{
    // ...
    public function testGuessLetterWithoutStartedGame()
    {
        $this->client->request('GET', '/game/letter/H');
        $this->assertTrue($this->client->getResponse()->isNotFound());
    }

    public function testGuessWordWithoutStartedGame()
    {
        $this->client->request('POST', '/game/word', array(
            'word' => 'php'
        ));

        $this->assertTrue($this->client->getResponse()->isNotFound());
    }
}
```



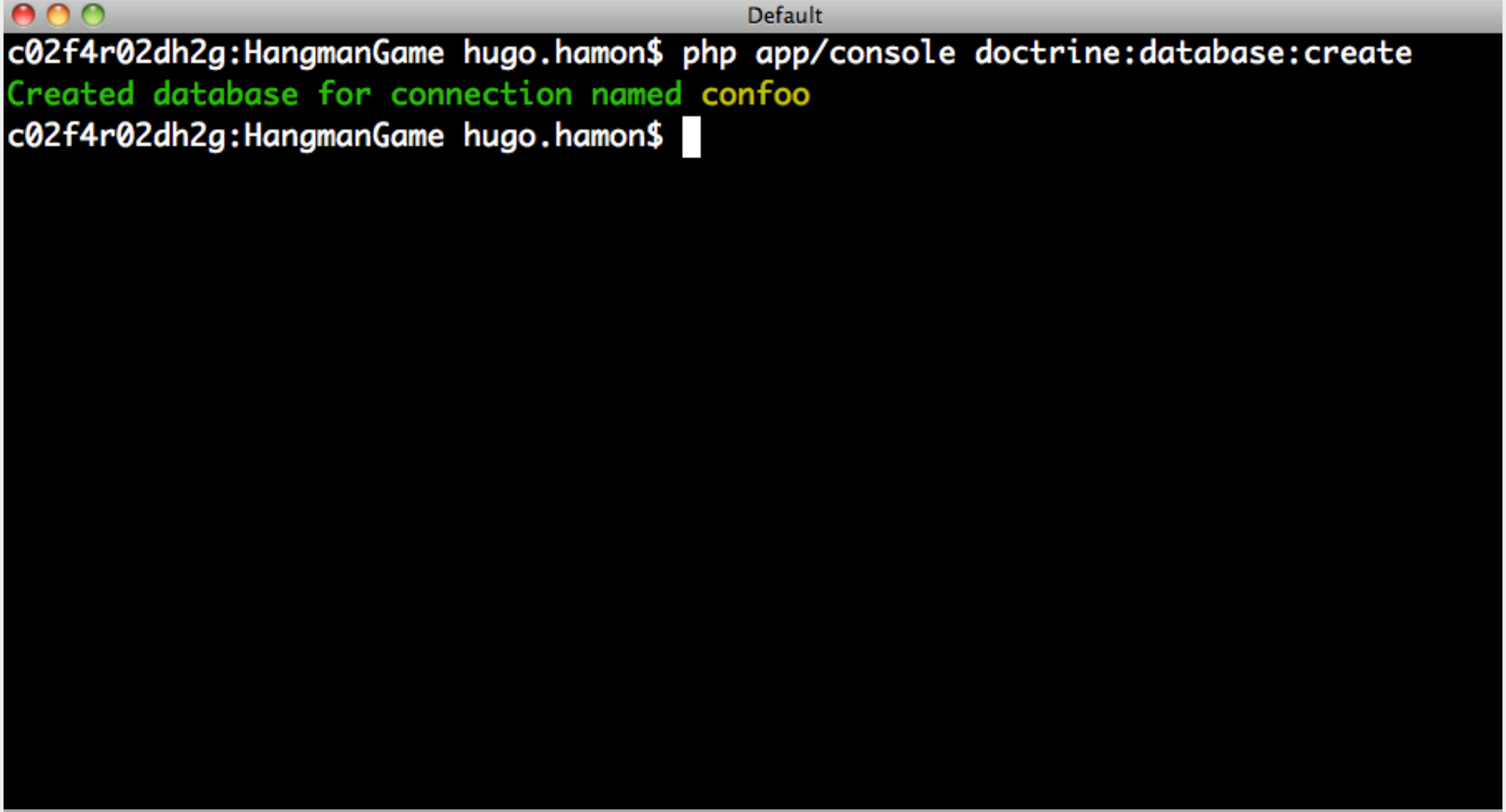
```
c02f4r02dh2g:HangmanGame hugo.hamon$ phpunit -c app/ --coverage-html ./coverage
PHPUnit 3.6.10 by Sebastian Bergmann.

Configuration read from /Users/hugo.hamon/Sites/HangmanGame/app/phpunit.xml.dist

............

Time: 11 seconds, Memory: 42.75Mb

OK (12 tests, 26 assertions)

Generating code coverage report, this may take a moment.
c02f4r02dh2g:HangmanGame hugo.hamon$ 
```

```
$ phpunit -c app/ --coverage-html ./coverage
```

Database Handling with Doctrine

Configuring the database access

```
# app/config/parameters.yml
parameters:
    database_driver     = pdo_mysql
    database_host       = localhost
    database_port       = 3306
    database_name       = hangman
    database_user       = root
    database_password   = root
```



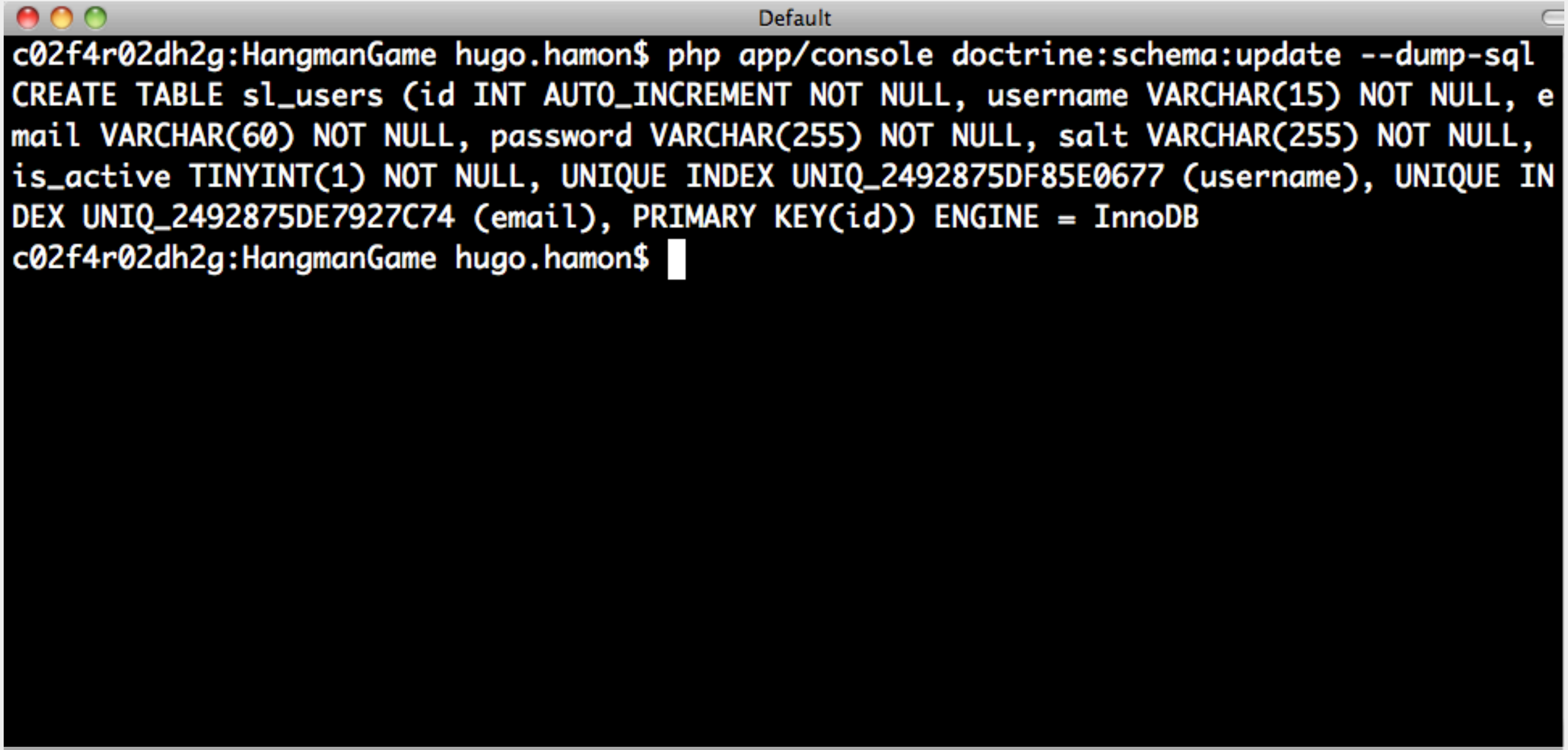
```
c02f4r02dh2g:HangmanGame hugo.hamon$ php app/console doctrine:database:create
Created database for connection named confoo
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ php app/console doctrine:database:create
```

The Player entity class

```php
/**
 * @ORM\Table(name="sl_players")
 * @ORM\Entity(repositoryClass="Sensio\Bundle\HangmanBundle\Entity\PlayerRepository")
 */
class Player
{
    // ...
    /** @ORM\Column(name="username", type="string", length=15, unique=true) */
    private $username;

    /** @ORM\Column(name="email", type="string", length=60, unique=true) */
    private $email;

    // ...
    /** @ORM\Column(name="is_active", type="boolean") */
    private $isActive;

    public function __construct()
    {
        $this->isActive = true;
    }
}
```

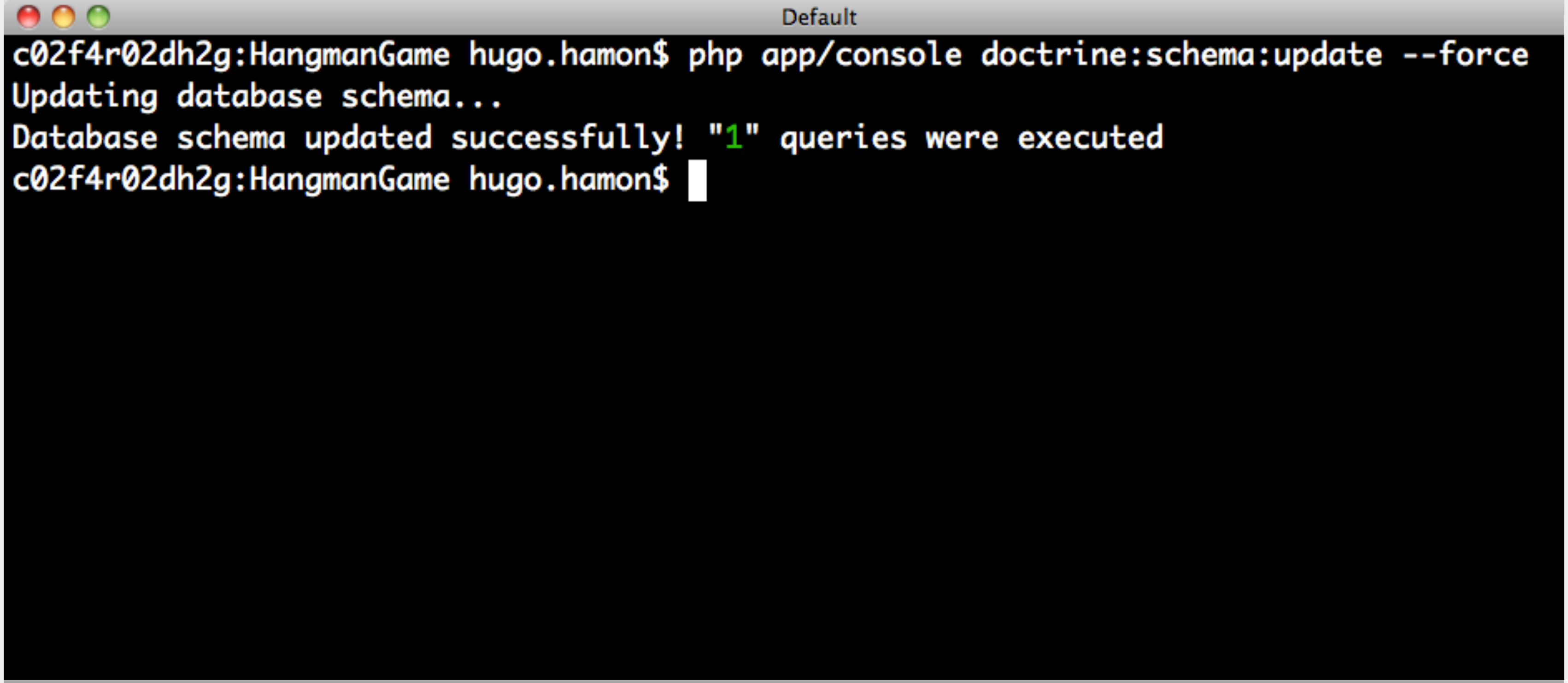
Generating the schema in the database



```
c02f4r02dh2g:HangmanGame hugo.hamon$ php app/console doctrine:schema:update --dump-sql
CREATE TABLE sl_users (id INT AUTO_INCREMENT NOT NULL, username VARCHAR(15) NOT NULL, e
mail VARCHAR(60) NOT NULL, password VARCHAR(255) NOT NULL, salt VARCHAR(255) NOT NULL,
is_active TINYINT(1) NOT NULL, UNIQUE INDEX UNIQ_2492875DF85E0677 (username), UNIQUE IN
DEX UNIQ_2492875DE7927C74 (email), PRIMARY KEY(id)) ENGINE = InnoDB
c02f4r02dh2g:HangmanGame hugo.hamon$ 
```

```
$ php app/console doctrine:schema:update --dump-sql
```

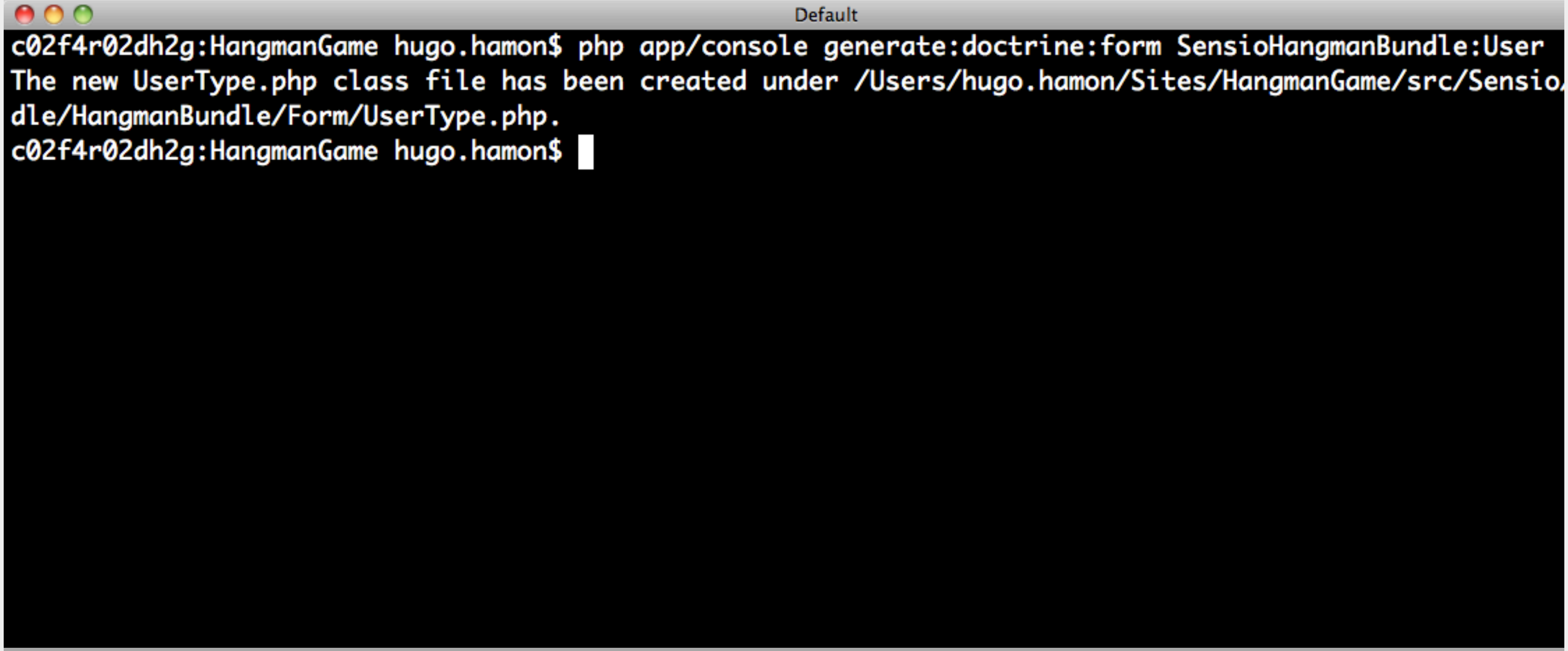
Generating the schema in the database



```
c02f4r02dh2g:HangmanGame hugo.hamon$ php app/console doctrine:schema:update --force
Updating database schema...
Database schema updated successfully! "1" queries were executed
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ php app/console doctrine:schema:update --force
```

Generating the form class



```
c02f4r02dh2g:HangmanGame hugo.hamon$ php app/console generate:doctrine:form SensioHangmanBundle:User
The new UserType.php class file has been created under /Users/hugo.hamon/Sites/HangmanGame/src/Sensio/
dle/HangmanBundle/Form/UserType.php.
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ php app/console generate:doctrine:form SensioHangmanBundle:Player
```

Tweaking the PlayerType form

```
class PlayerType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('username')
            ->add('email', 'email')
            ->add('rawPassword', 'repeated', array(
                'type' => 'password',
                'first_options' => array('label' => 'Password'),
                'second_options' => array('label' => 'Confirmation'),
            ))
        ;
    }
}
```

Tweaking the PlayerType form

```
class PlayerType extends AbstractType
{
    public function getName()
    {
        return 'player';
    }

    public function setDefaultOptions(OptionsResolverInterface $resolver)
    {
        $resolver->setDefaults(array(
            'data_class' => 'Sensio\Bundle\HangmanBundle\Entity\Player',
        ));
    }
}
```

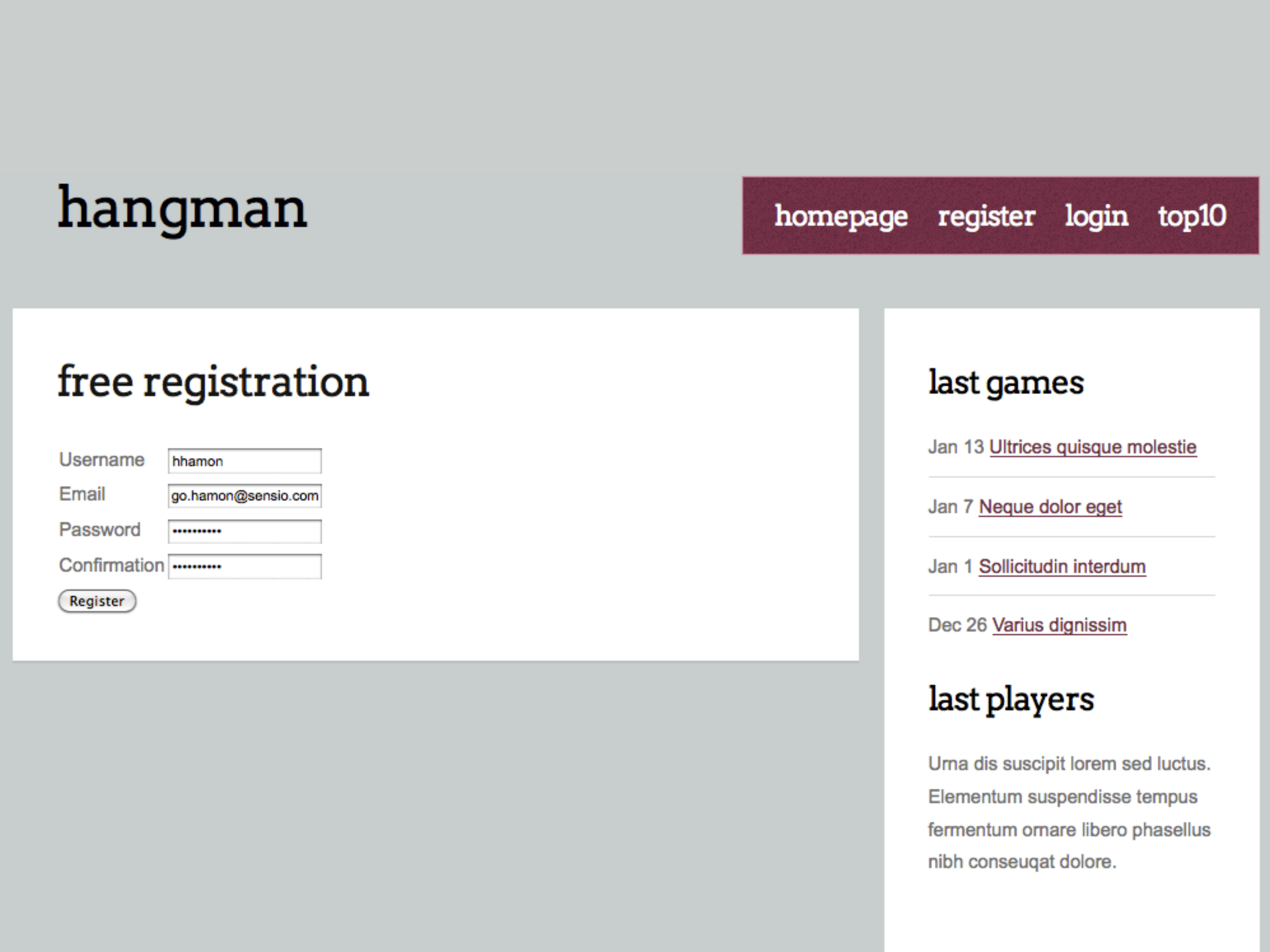
Adding the new user controller class

```php
namespace Sensio\Bundle\HangmanBundle\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;
use Sensio\Bundle\HangmanBundle\Entity\Player;
use Sensio\Bundle\HangmanBundle\Form\PlayerType;

class PlayerController extends Controller
{
    public function signupAction(Request $request)
    {

    }
}
```

```
class PlayerController extends Controller
{
    /**
     * @Route("/signup", name="signup")
     * @Template()
     */
    public function signupAction(Request $request)
    {
        $player = new Player();
        $form = $this->createForm(new PlayerType(), $player);

        if ($request->isMethod('POST')) {
            $form->submit($request);
            if ($form->isValid()) {
                $em = $this->get('doctrine')->getManager();
                $em->persist($player);
                $em->flush();

                return $this->redirect($this->generateUrl('hangman_game'));
            }
        }

        return array('form' => $form->createView());
    }
}
```

Adding the corresponding template

```
{% extends "SensioHangmanBundle::layout.html.twig" %}

{% form_theme form "form_table_layout.html.twig" %}

{% block body %}

    <h2>Free registration</h2>

    <form novalidate action="{{ path('signup') }}" method="post">
        {{ form_widget(form) }}
        <div>
            <button type="submit">Register</button>
        </div>
    </form>

{% endblock body %}
```



hangman

free registration

Username	hhamon
Email	go.hamon@sensio.com
Password	•••••••••••
Confirmation	•••••••••••

last games

Jan 13 Ultrices quisque molestie

Jan 7 Neque dolor eget

Jan 1 Sollicitudin interdum

Dec 26 Varius dignissim

last players

Urna dis suscipit lorem sed luctus. Elementum suspendisse tempus fermentum ornare libero phasellus nibh conseuqat dolore.

Adding some input validation

```
// ...
use Symfony\Component\Validator\Constraints as Assert;

// ...
class Player
{
    // ...
    /**
     * ...
     * @Assert\NotBlank()
     * @Assert\Length(min = 6, max = 15)
     * @Assert\Regex("/^[a-z][a-z0-9]+$/i")
     */
    private $username;
    // ...
}
```

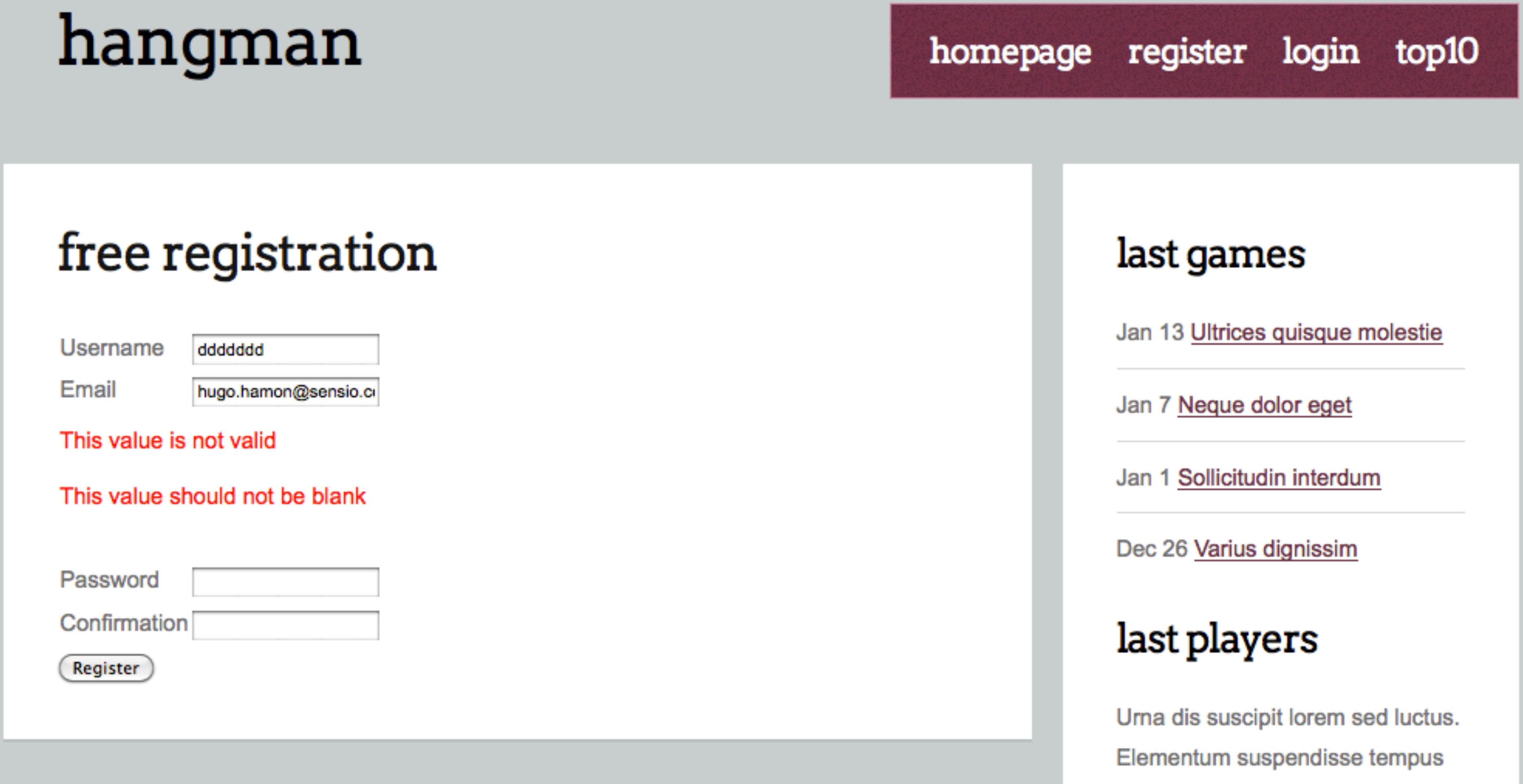
```
class Player
{
    // ...

    /**
     * ...
     * @Assert\NotBlank()
     * @Assert\Email()
     */
    private $email;

    /**
     *  ...
     * @Assert\NotBlank()
     * @Assert\Length(min = 8)
     */
    private $rawPassword;

    // ...
}
```

Avoiding the username as the password

```
// ...

class Player
{
    // ...

    /**
     * @Assert\True(message="Your password must not contain your
username")
     *
     */
    public function isPasswordValid()
    {
        return 0 === preg_match('/'.preg_quote($this->username).'/i',
$this->rawPassword);
    }
}
```

Checking for username and email uniqness

```
// ...

use Symfony\Bridge\Doctrine\Validator\Constraints\UniqueEntity;

/**
 * ...
 *
 * @UniqueEntity(fields="username", message="Username already taken")
 * @UniqueEntity(fields="email", message="Email already taken")
 */
class Player
{
    // ...
}
```



free registration

Username: ddddddd

Email: hugo.hamon@sensio.c

This value is not valid

This value should not be blank

Password:

Confirmation:

last games

Jan 13 Ultrices quisque molestie

Jan 7 Neque dolor eget

Jan 1 Sollicitudin interdum

Dec 26 Varius dignissim

last players

Urna dis suscipit lorem sed luctus. Elementum suspendisse tempus

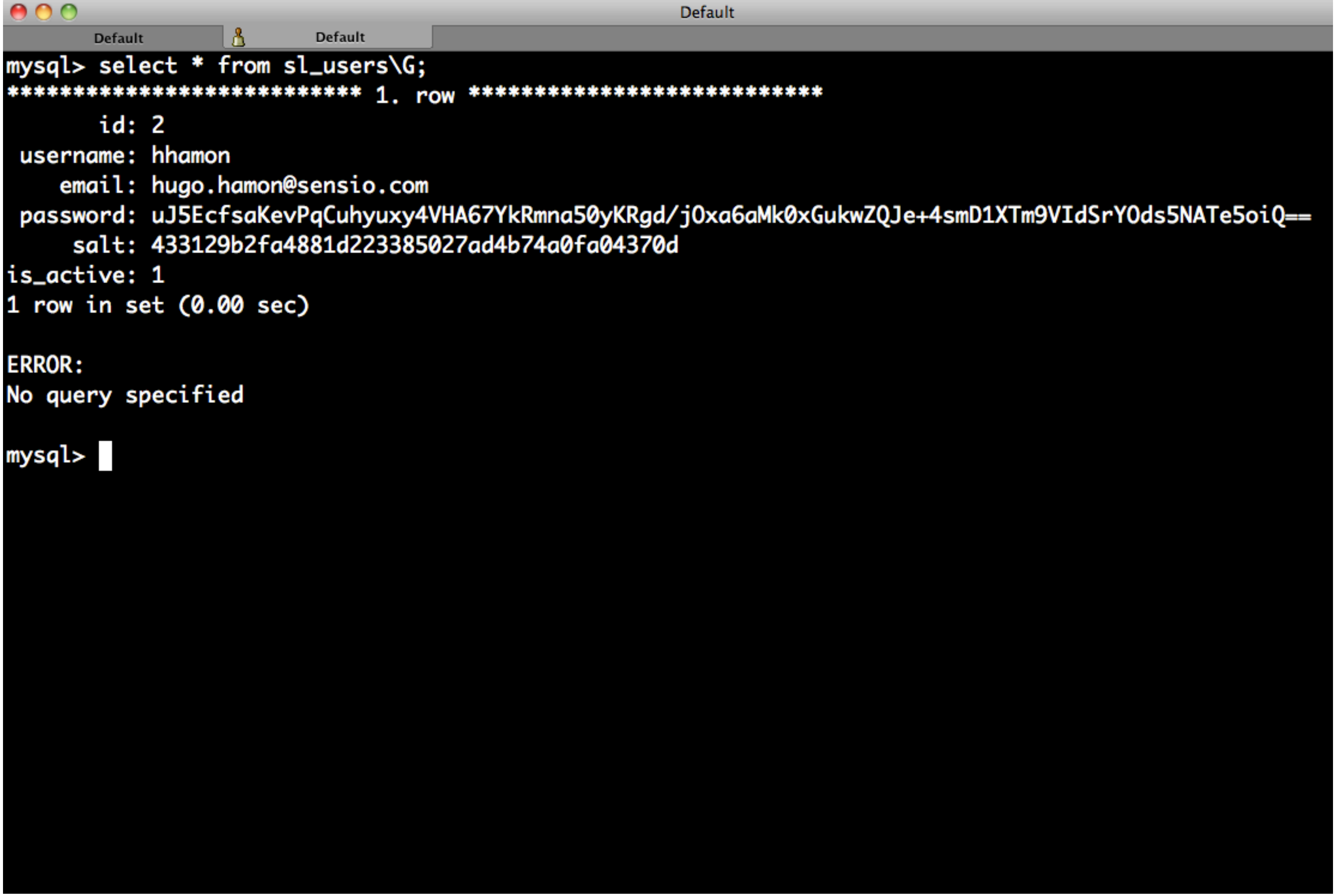
Securing the Game

Dealing with
hashed passwords

```
security:
    encoders:
        Sensio\Bundle\HangmanBundle\Entity\Player: sha512

    providers:
        players:
            entity:      { class: SensioHangmanBundle:Player, property: username }

    firewalls:
        dev:
            pattern:                 ^/(_(profiler|wdt)|css|images|js)/
            security:                false

        game:
            pattern:                 ^/
            anonymous:               true
            form_login:
                login_path:                     signin
                check_path:                     login_check
                default_target_path:            hangman_game
                always_use_default_target_path: true
            logout:
                path:                           signout
                target:                         signin

    access_control:
        - { path: ^/game, roles: IS_AUTHENTICATED_FULLY }
```

```php
use Symfony\Component\Security\Core\Encoder\PasswordEncoderInterface;

class Player
{
    public function encodePassword(PasswordEncoderInterface $encoder)
    {
        if ($this->rawPassword) {
            $this->salt = sha1(uniqid(mt_rand()));
            $this->password = $encoder->encodePassword(
                $this->rawPassword,
                $this->salt
            );
            $this->rawPassword = null;
        }
    }
}
```

```
class PlayerController extends Controller
{
    // ...
    public function registrationAction(Request $request)
    {
        // ...
        if ($request->isMethod('POST')) {
            $form->submit($request);
            if ($form->isValid()) {

                $factory = $this->get('security.encoder_factory');
                $encoder = $factory->getEncoder($player);
                $player->encodePassword($encoder);

                $em = $this->get('doctrine')->getManager();
                // ...
            }
        }
        // ...
    }
}
```



```
mysql> select * from sl_users\G;
*************************** 1. row ***************************
       id: 2
 username: hhamon
    email: hugo.hamon@sensio.com
 password: uJ5EcfsaKevPqCuhyuxy4VHA67YkRmna50yKRgd/jOxa6aMk0xGukwZQJe+4smD1XTm9VIdSrYOds5NATe5oiQ==
     salt: 433129b2fa4881d223385027ad4b74a0fa04370d
is_active: 1
1 row in set (0.00 sec)

ERROR:
No query specified

mysql> 
```

The UserInterface

```
interface UserInterface
{
    function getPassword();

    function getSalt();

    function getUsername();

    function getRoles();

    function eraseCredentials();
}
```

```
use Symfony\Component\Security\Core\User\UserInterface;

class Player implements UserInterface
{
    // ...

    public function eraseCredentials()
    {
        $this->rawPassword = null;
    }
}
```

```
use Symfony\Component\Security\Core\User\UserInterface;

// ...
class Player implements UserInterface
{
    // ...

    public function getRoles()
    {
        if ($this->isAdmin) {
            return array('ROLE_ADMIN');
        }

        return array('ROLE_PLAYER');
    }
}
```

Registering new security routes

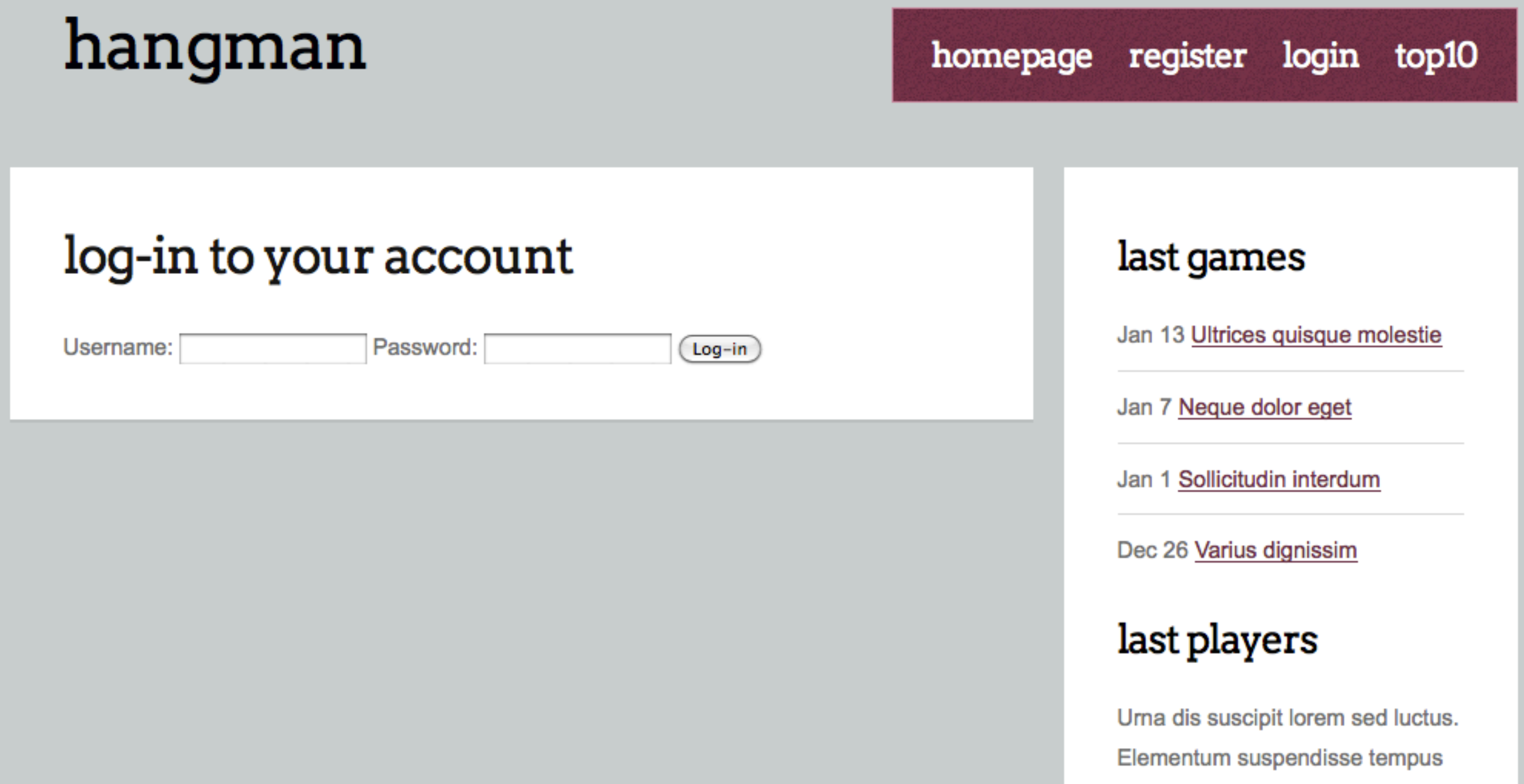
```
# app/config/routing.yml
login_check:
    pattern:  /auth
    requirements: { _method: POST }

signout:
    pattern: /logout
```

```
class PlayerController extends Controller
{
    /**
     * @Route("/signin", name = "signin")
     * @Template()
     */
    public function signinAction(Request $request)
    {
        $session = $request->getSession();

        // ...
    }
}
```

```php
use Symfony\Component\Security\Core\SecurityContext;

public function signinAction(Request $request)
{
    $session = $request->getSession();

    if ($username = $session->get(SecurityContext::LAST_USERNAME)) {
        $session->remove(SecurityContext::LAST_USERNAME);
    }

    if ($error = $session->get(SecurityContext::AUTHENTICATION_ERROR)) {
        $session->remove(SecurityContext::AUTHENTICATION_ERROR);
    }

    return array('last_username' => $username, 'error' => $error);
}
```

The Login Form

```
{% extends "SensioHangmanBundle::layout.html.twig" %}

{% block title "Signin to your account" %}

{% block body %}

    <h2>Log-in to your account</h2>

    {% if error %}
        <div class="error">{{ error.message }}</div>
    {% endif %}

    <form action="{{ path('login_check') }}" method="post">
        <label for="username">Username:</label>
        <input type="text" id="username" name="_username" value="{{ last_username }}" />
        <label for="password">Password:</label>
        <input type="password" id="password" name="_password" />
        <button type="submit">Log-in</button>
    </form>

{% endblock %}
```



hangman

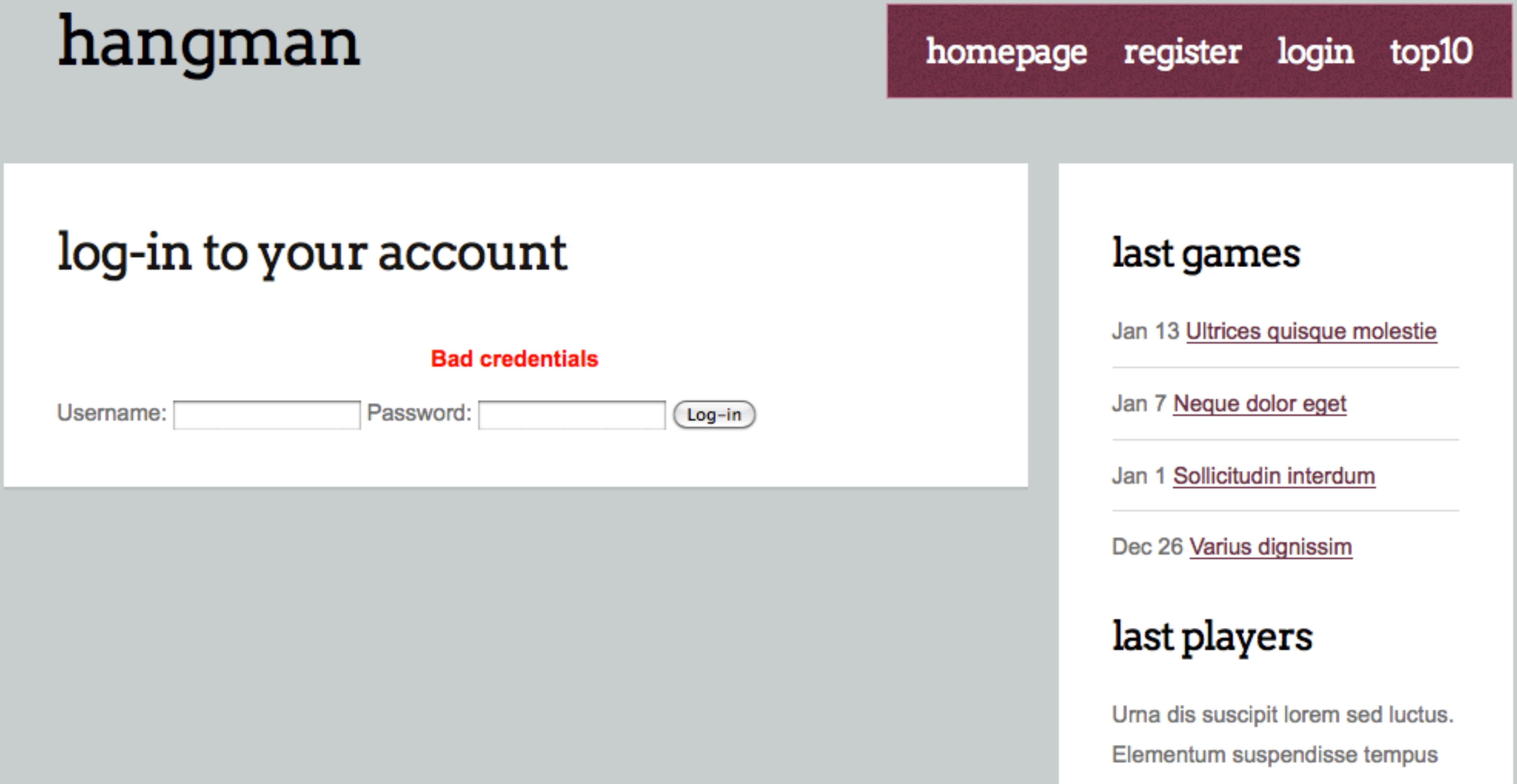
log-in to your account

last games

- Jan 13 Ultrices quisque molestie
- Jan 7 Neque dolor eget
- Jan 1 Sollicitudin interdum
- Dec 26 Varius dignissim

last players

Urna dis suscipit lorem sed luctus. Elementum suspendisse tempus



log-in to your account

Bad credentials

Username: ______ Password: ______

last games

Jan 13 Ultrices quisque molestie

Jan 7 Neque dolor eget

Jan 1 Sollicitudin interdum

Dec 26 Varius dignissim

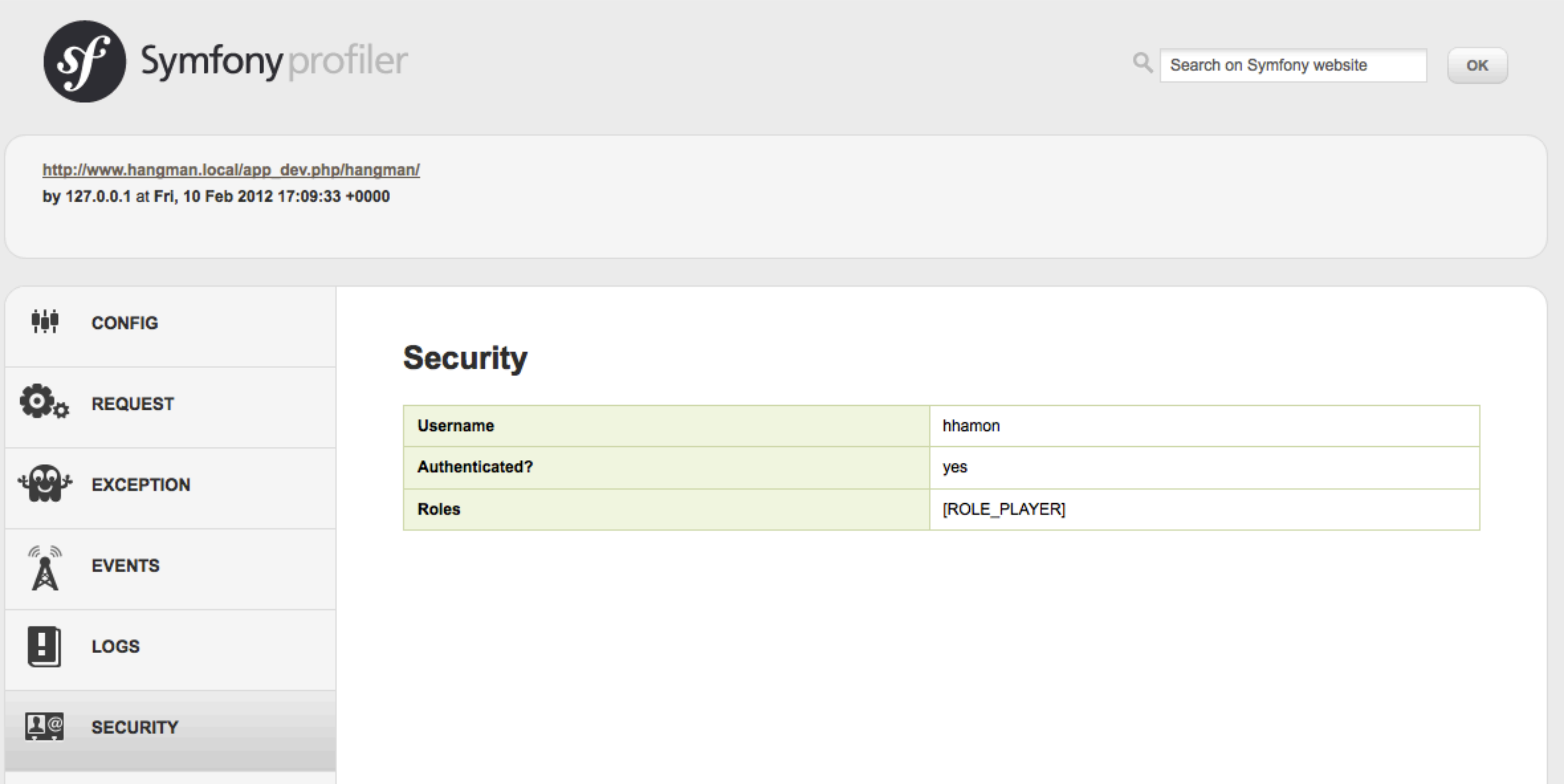
last players

Urna dis suscipit lorem sed luctus.

Elementum suspendisse tempus

Updating the main navigation bar

```
<div id="menu">
    <ul>
        <li class="first current_page_item">
            <a href="{{ path('homepage') }}">Homepage</a>
        </li>
        <li>
            <a href="{{ path('signup') }}">Register</a>
        </li>
        <li>
            <a href="{{ path('signin') }}">Login</a>
        </li>
        <li>
            <a href="#">Top10</a>
        </li>
    </ul>
    <br class="clearfix" />
</div>
```



http://www.hangman.local/app_dev.php/hangman/

by 127.0.0.1 at Fri, 10 Feb 2012 17:09:33 +0000

Security

Username	hhamon
Authenticated?	yes
Roles	[ROLE_PLAYER]

HTTP Caching

Caching with ESI.

Caching the list of last players with an ESI

```
<div id="sidebar">
    <!-- ... -->
    <h3>Last players</h3>
    {{
       render_esi(controller(
            'SensioHangmanBundle:Player:players',
            { 'max': 10 }
       ))
    }}
</div>
```

```
// ...
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Cache;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;

class PlayerController extends Controller
{
    /**
     * @Cache(smaxage = 120)
     * @Template()
     */
    public function playersAction($max)
    {
        $em = $this->get('doctrine')->getManager();
        $repository = $em->getRepository('SensioHangmanBundle:Player');

        return array(
            'players' => $repository->getMostRecentPlayers($max),
        );
    }
}
```

```php
class PlayerRepository extends EntityRepository
{
    public function getMostRecentPlayers($max)
    {
        $q = $this
            ->createQueryBuilder('p')
            ->select('p.id, p.username')
            ->where('p.isActive = :active')
            ->andWhere('p.isAdmin = :admin')
            ->orderBy('p.id', 'DESC')
            ->setMaxResults($max)
            ->setParameter('active', true)
            ->setParameter('admin', false)
            ->getQuery()
        ;

        return $q->getArrayResult();
    }
}
```

Caching the list of last players with an ESI

```
{# @SensioHangman/Player/players.html.twig #}
<ul>
    {% for player in players %}
        <li>{{ player.username|capitalize }}</li>
    {% else %}
        <li>No players yet.</li>
    {% endfor %}
</ul>
```

Training Department

Sensio S.A.
92-98 Boulevard Victor Hugo
92 115 Clichy Cedex
FRANCE

Tél. : +33 140 998 211

symfony.com - trainings.sensiolabs.com