

**ConFoo.CA**  
WEB TECHNO CONFERENCE

**Sensio**Labs



Symfony

# Diving Into Symfony Advanced Features

# The Hangman Game

# Installing the Project

# hangman

[homepage](#) [register](#) [login](#) [top10](#)

## guess the mysterious word

You still have 11 remaining attempts.

? ? ? ? ? ? ? ?

[Reset the game](#)

## try a letter

A B C D E F G H I J K L M  
N O P Q R S T U V W X Y Z

## 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 consequat dolore.

# hangman

[homepage](#) [register](#) [login](#) [top10](#)

## guess the mysterious word

You still have 9 remaining attempts.

H A ? ? W A ? E

[Reset the game](#)

## try a letter

A B C D E F G H I J K L M  
N O P Q R S T U V W X Y Z

## 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 consequat dolore.

# hangman

[homepage](#) [register](#) [login](#) [top10](#)

## 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 consequat dolore.



# hangman

[homepage](#) [register](#) [login](#) [top10](#)

## 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 consequat dolore.

# Introducing the GameContext service

# Injecting the SessionStorage in the Context

```
# src/Sensio/Bundle/HangmanBundle/Game/GameContext.php

namespace Sensio\Bundle\HangmanBundle\Game;

use Symfony\Component\HttpFoundation\SessionStorage\SessionStorageInterface;

class GameContext
{
    private $storage;

    private $wordList;

    public function __construct(SessionStorageInterface $storage, WordList $list)
    {
        $this->storage = $storage;
        $this->wordList = $list;
    }
}
```

```
class GameContext
{
    // ...

    public function reset()
    {
        $this->storage->write('hangman', array());
    }

    public function loadGame()
    {
        $data = $this->storage->read('hangman');

        // ...
    }

    public function save(Game $game)
    {
        $this->storage->write('hangman', $game->getContext());
    }
}
```

# Registering a new Game Context service

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

```
<container>
```

```
  <parameters>
```

```
    <parameter key="hangman.game_context.class">Sensio\Bundle  
    \HangmanBundle\Game\GameContext</parameter>
```

```
    <parameter key="hangman.word_list.class">Sensio\Bundle  
    \HangmanBundle\Game\WordList</parameter>
```

```
    <parameter key="hangman.word_list.dictionaries"  
type="collection"/>
```

```
  </parameters>
```

```
</container>
```

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

  <services>
    <service id="hangman.game_context"
      class="%hangman.game_context.class%"
    >
      <argument type="service" id="session.storage" />
      <argument type="service" id="hangman.word_list" />
    </service>

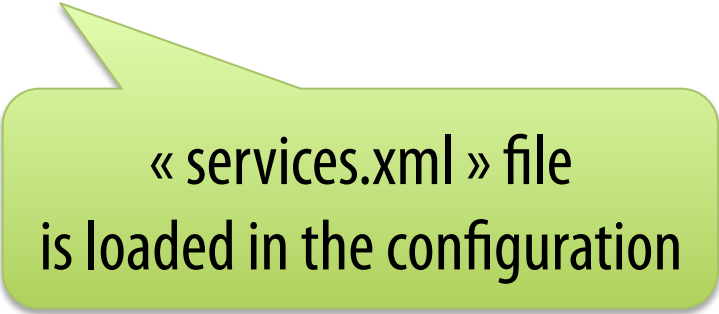
    <service id="hangman.word_list"
      class="%hangman.word_list.class%" public="false"
    >
      <argument>%hangman.word_list.dictionaries%</argument>
      <call method="loadDictionaries"/>
    </service>

  </services>
</container>
```

# Loading the Configuration

```
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 » file  
is loaded in the configuration

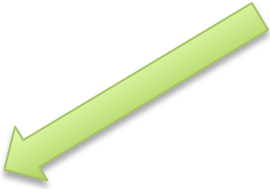
```
Default
c02f4r02dh2g:HangmanGame hugo.hamon$ php app/console container:debug hangman.game_context
[container] Information for service hangman.game_context

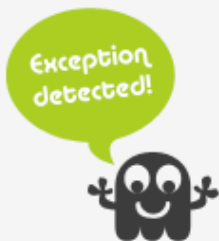
Service Id    hangman.game_context
Class         Sensio\Bundle\HangmanBundle\Game\GameContext
Tags          -
Scope         container
Public        yes
c02f4r02dh2g:HangmanGame hugo.hamon$
```



# Refactoring the GameController class

```
// ...  
class GameController extends Controller  
{  
    // ...  
    public function indexAction(Request $request)  
    {  
        $context = $this->get('hangman.game_context');  
  
        $max      = $this->container->getParameter('hangman.word_length');  
        $length   = $request->query->get('length', $max);  
        if (!$game = $context->loadGame()) {  
            $game = $context->newGame($length);  
            $context->save($game);  
        }  
  
        return array('game' => $game);  
    }  
}
```





There is no word of length 8.  
500 Internal Server Error - InvalidArgumentException



## Stack Trace

1. in /Users/hugo.hamon/Sites/HangmanGame/src/Sensio/Bundle/HangmanBundle/Game/WordList.php at line 46 

```
43.     public function getRandomWord($length)
44.     {
45.         if (!isset($this->words[$length])) {
46.             throw new \InvalidArgumentException(sprintf('There is no word of length %u.', $length));
47.         }
```

# Fixing the main configuration

```
# app/config/config.yml
```

```
imports:
```

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

```
parameters:
```

```
hangman.word_list.dictionaries:
```

- "%kernel.root\_dir%/data/words.txt"
- "%kernel.root\_dir%/data/words.xml"



# Unit Testing

```
Default
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

F

Time: 0 seconds, Memory: 6.00Mb

There was 1 failure:

1) Warning
No tests found in class "Sensio\Bundle\HangmanBundle\Tests\WordTest".

FAILURES!
Tests: 1, Assertions: 0, Failures: 1.
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ phpunit -c app
```

# Testing the Word Class

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

use Sensio\Bundle\HangmanBundle\Game\Word;

class WordTest extends \PHPUnit_Framework_TestCase
{
    public function testIsGuessed()
    {
        $word = new Word('php');
        $word->tryLetter('h');
        $this->assertFalse($word->isGuessed());

        $word->tryLetter('p');
        $this->assertTrue($word->isGuessed());
    }
}
```

```
Default
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: 6.00Mb

OK (1 test, 2 assertions)
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ phpunit -c app
```

# Testing the Word Class

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

use Sensio\Bundle\HangmanBundle\Game\Word;

class WordTest extends \PHPUnit_Framework_TestCase
{
    // ...

    public function testTryLetter()
    {
        $word = new Word('hangman');
        $this->assertTrue($word->tryLetter('g'));
        $this->assertEquals(array('g'), $word->getFoundLetters());
    }
}
```



```
Default
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: 6.50Mb

OK (2 tests, 4 assertions)
c02f4r02dh2g:HangmanGame hugo.hamon$
```

```
$ phpunit -c app
```

# Testing the Word Class

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

use Sensio\Bundle\HangmanBundle\Game\Word;

class WordTest extends \PHPUnit_Framework_TestCase
{
    // ...

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

        $word = new Word('hangman');
        $word->tryLetter('3');
    }
}
```

```
Default
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: 0 seconds, Memory: 7.25Mb

OK (3 tests, 5 assertions)





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

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

## Game

Current directory: [/Users/hugo.hamon/Sites/HangmanGame/src/Sensio/Bundle/HangmanBundle/Game](#) (dashboard)

Legend: Low: 0% to 35% Medium: 35% to 70% High: 70% to 100%

	Coverage					
	Lines			Functions / Methods		
Total	<div><div></div></div>	15.69%	16 / 102	<div><div></div></div>	11.76%	4 / 34
 <a href="#">Game.php</a>	<div><div></div></div>	0.00%	0 / 27	<div><div></div></div>	0.00%	0 / 11
 <a href="#">GameContext.php</a>	<div><div></div></div>	0.00%	0 / 15	<div><div></div></div>	0.00%	0 / 7
 <a href="#">Word.php</a>	<div><div></div></div>	66.67%	16 / 24	<div><div></div></div>	40.00%	4 / 10
 <a href="#">WordList.php</a>	<div><div></div></div>	0.00%	0 / 36	<div><div></div></div>	0.00%	0 / 6

Generated by [PHP CodeCoverage 1.1.1](#) using [PHP 5.3.10](#) and [PHPUnit 3.6.10](#) at Thu Feb 9 14:48:59 L

Word class is covered up to 67%

# Word.php

Current file: /Users/hugo.hamon/Sites/HangmanGame/src/Sensio/Bundle/HangmanBundle/Game/Word.php

Legend: executed not executed dead code

	Coverage									
	Classes			Functions / Methods				Lines		
Total	<div></div>	0.00%	0 / 1	<div></div>	40.00%	4 / 10	CRAP	<div></div>	66.67%	16 / 24
Word	<div></div>	0.00%	0 / 1	<div></div>	40.00%	4 / 10	19.26	<div></div>	66.67%	16 / 24
<u>__construct(\$word, array \$foundLetters = array())</u>	<div></div>			<div></div>	100.00%	1 / 1	1	<div></div>	100.00%	4 / 4
<u>toString()</u>	<div></div>			<div></div>	0.00%	0 / 1	2	<div></div>	0.00%	0 / 1
<u>split()</u>	<div></div>			<div></div>	0.00%	0 / 1	2	<div></div>	0.00%	0 / 1
<u>guessed()</u>	<div></div>			<div></div>	0.00%	0 / 1	2	<div></div>	0.00%	0 / 2
<u>getFoundLetters()</u>	<div></div>			<div></div>	100.00%	1 / 1	1	<div></div>	100.00%	1 / 1
<u>getTriedLetters()</u>	<div></div>			<div></div>	0.00%	0 / 1	2	<div></div>	0.00%	0 / 1
<u>getLetters()</u>	<div></div>			<div></div>	100.00%	1 / 1	1	<div></div>	100.00%	1 / 1
<u>isGuessed()</u>	<div></div>			<div></div>	100.00%	1 / 1	1	<div></div>	100.00%	2 / 2
<u>tryLetter(\$letter)</u>	<div></div>			<div></div>	0.00%	0 / 1	4.13	<div></div>	80.00%	8 / 10
<u>getWord()</u>	<div></div>			<div></div>	0.00%	0 / 1	2	<div></div>	0.00%	0 / 1

```

1      : <?php
2      :
3      : namespace Sensio\Bundle\HangmanBundle\Game;
4      :
5      : class Word
6      : {
7      :     private $word;
8      :
9      :     private $foundLetters;
10     :
11     :     private $triedLetters;
12     :

```

Word class is covered up to 67%

```

:      /**
:      * Tries to guess a letter.
:      *
:      * @param string $letter A letter to try
:      * @return Boolean
:      * @throws InvalidArgumentException
:      */
:      public function tryLetter($letter)
:      {
3 :          $letter = strtolower($letter);
3 :          if (0 === preg_match('/^[a-z]$/', $letter)) {
1 :              throw new \InvalidArgumentException(sprintf('The letter "%s" is not a valid ASCII character matching [a-z].', $letter));
:          }
:
2 :          if (in_array($letter, $this->triedLetters)) {
0 :              throw new \InvalidArgumentException(sprintf('The letter "%s" has already been tried.', $letter));
:          }
:
2 :          $this->triedLetters[] = $letter;
:
2 :          if (false !== strpos($this->word, $letter)) {
2 :              $this->foundLetters[] = $letter;
:
2 :              return true;
:          }
:
0 :          return false;
:      }
:

```

Two lines are not yet covered by unit tests for the  
Word::tryLetter method.

# Testing the Word Class

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

    public function testTryLetterWithTriedLetter()
    {
        $word = new Word('hangman');
        $this->assertFalse($word->tryLetter('p'));

        $this->setExpectedException('InvalidArgumentException');
        $word->tryLetter('p');
    }
}
```

```
Default
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: 0 seconds, Memory: 7.25Mb

OK (4 tests, 7 assertions)

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

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



```

:  /**
:   * Returns whether or not the word has been guessed.
:   *
:   * @return Boolean
:   */
:  public function isGuessed()
:  {
:      $diff = array_diff($this->getLetters(), $this->foundLetters);
:
:      return 0 === count($diff);
:  }
:
:  /**
:   * Tries to guess a letter.
:   *
:   * @param string $letter A letter to try
:   * @return Boolean
:   * @throws InvalidArgumentException
:   */
:  public function tryLetter($letter)
:  {
:      $letter = strtolower($letter);
:      if (0 === preg_match('/^[a-z]$/', $letter)) {
:          throw new \InvalidArgumentException(sprintf('The letter "%s" is not a valid ASCII character matching [a-z].', $letter));
:      }
:
:      if (in_array($letter, $this->triedLetters)) {
:          throw new \InvalidArgumentException(sprintf('The letter "%s" has already been tried.', $letter));
:      }
:
:      $this->triedLetters[] = $letter;
:
:      if (false !== strpos($this->word, $letter)) {
:          $this->foundLetters[] = $letter;
:
:          return true;
:      }
:
:      return false;
:  }
:
:  }
:

```

The Word::tryLetter method is now fully covered!

# Testing the Game Class

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

use Sensio\Bundle\HangmanBundle\Game\Game;
use Sensio\Bundle\HangmanBundle\Game\Word;

class GameTest extends \PHPUnit_Framework_TestCase
{
    public function testTryWord()
    {
        $class = 'Sensio\\Bundle\\HangmanBundle\\Game\\Word';
        $game = new Game(new Word('php'));
        $this->assertInstanceOf($class, $game->getWord());
        $this->assertTrue($game->tryWord('php'));
        $this->assertEquals(0, $game->getAttempts());
    }
}
```

```
Default
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: 1 second, Memory: 7.75Mb

OK (5 tests, 10 assertions)

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

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

```

:      public function getAttempts()
:      {
1 :          return $this->attempts;
:      }
:
:      public function tryWord($word)
:      {
1 :          if ($word === $this->word->getWord()) {
1 :              $this->word->guessed();
:
1 :          return true;
:      }
:
0 :          $this->attempts = self::MAX_ATTEMPTS;
:
0 :          return false;
:      }
:
:      public function tryLetter($letter)
:      {
:          try {
0 :              $result = $this->word->tryLetter($letter);
0 :          } catch (\InvalidArgumentException $e) {
0 :              $result = false;
:          }
:
0 :          if (false === $result) {
0 :              $this->attempts++;
0 :          }
:
:      }
:      return $result;
:      }

```

Game class still needs to be covered...

# Testing the Game Class

```
class GameTest extends \PHPUnit_Framework_TestCase
{
    public function testTryLetter()
    {
        $game = new Game(new Word('php'));

        $this->assertFalse($game->tryLetter('3'));
        $this->assertEquals(1, $game->getAttempts());

        $this->assertFalse($game->tryLetter('e'));
        $this->assertEquals(2, $game->getAttempts());

        $this->assertTrue($game->tryLetter('p'));
        $this->assertEquals(2, $game->getAttempts());

        $this->assertFalse($game->tryLetter('p'));
        $this->assertEquals(3, $game->getAttempts());
    }
}
```

```
Default
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: 0 seconds, Memory: 7.75Mb

OK (6 tests, 18 assertions)

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

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

# Testing the Game Class

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

use Sensio\Bundle\HangmanBundle\Game\Game;
use Sensio\Bundle\HangmanBundle\Game\Word;

class GameTest extends \PHPUnit_Framework_TestCase
{
    // ...

    public function testTryInvalidWord()
    {
        $game = new Game(new Word('php'));

        $this->assertFalse($game->tryWord('html'));
        $this->assertEquals(Game::MAX_ATTEMPTS, $game->getAttempts());
    }
}
```

```
Default
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: 0 seconds, Memory: 7.75Mb

OK (7 tests, 20 assertions)

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

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



```

:
:     public function isWon()
:     {
0 :         return $this->word->isGuessed();
:     }
:
:     public function getWord()
:     {
1 :         return $this->word;
:     }
:
:     public function getWordLetters()
:     {
0 :         return $this->word->split();
:     }
:
:     public function getAttempts()
:     {
3 :         return $this->attempts;
:     }
:
:     public function tryWord($word)
:     {
2 :         if ($word === $this->word->getWord()) {
1 :             $this->word->guessed();
:
1 :             return true;
:         }
:
1 :         $this->attempts = self::MAX_ATTEMPTS;
:
1 :         return false;
:
:

```

Game class is now covered up 63%

# Functional Testing

# Setting up the environment

```
# app/config/config_test.yml

imports:
  - { resource: config_dev.yml }

parameters:
  hangman.word_length: 3

  hangman.word_list.dictionaries:
    - "%kernel.root_dir%/../src/Sensio/Bundle/HangmanBundle/Tests/Fixtures/words.txt"
```

We need to define a dedicated list of words for the testing environment and avoid to mixup production and test configurations.

```
# @SensioHangmanBundle/Tests/Controller/GameControllerTest.php
```

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

```
use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;
```

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

```
class GameControllerTest extends WebTestCase
```

```
{
```

```
    private $client;
```

```
    public function setUp()
```

```
{
```

```
        $this->client = static::createClient();
```

```
}
```

```
    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/hangman/');
        $form = $crawler->selectButton('Let me guess...')->form();
        $this->client->submit($form, array('word' => 'php'));
        $crawler = $this->client->followRedirect();

        // Check the game is won
        $response = $this->client->getResponse();
        $this->assertRegexp(
            '#You found the word <strong>php</strong>#',
            $response->getContent()
        );
    }
}
```

```
Default
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/hangman/');
        $form = $crawler->selectButton('Let me guess...')->form();
        $this->client->submit($form, array('word' => 'foo'));
        $crawler = $this->client->followRedirect();

        $response = $this->client->getResponse();
        $this->assertRegexp(
            "/Oops, you're hanged/",
            $response->getContent()
        );
    }
}
```

```
Default
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

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

    private function playWord($word)
    {
        $crawler = $this->client->request('GET', '/game/hangman/');

        $form = $crawler->selectButton('Let me guess...')->form();
        $this->client->submit($form, array('word' => $word));

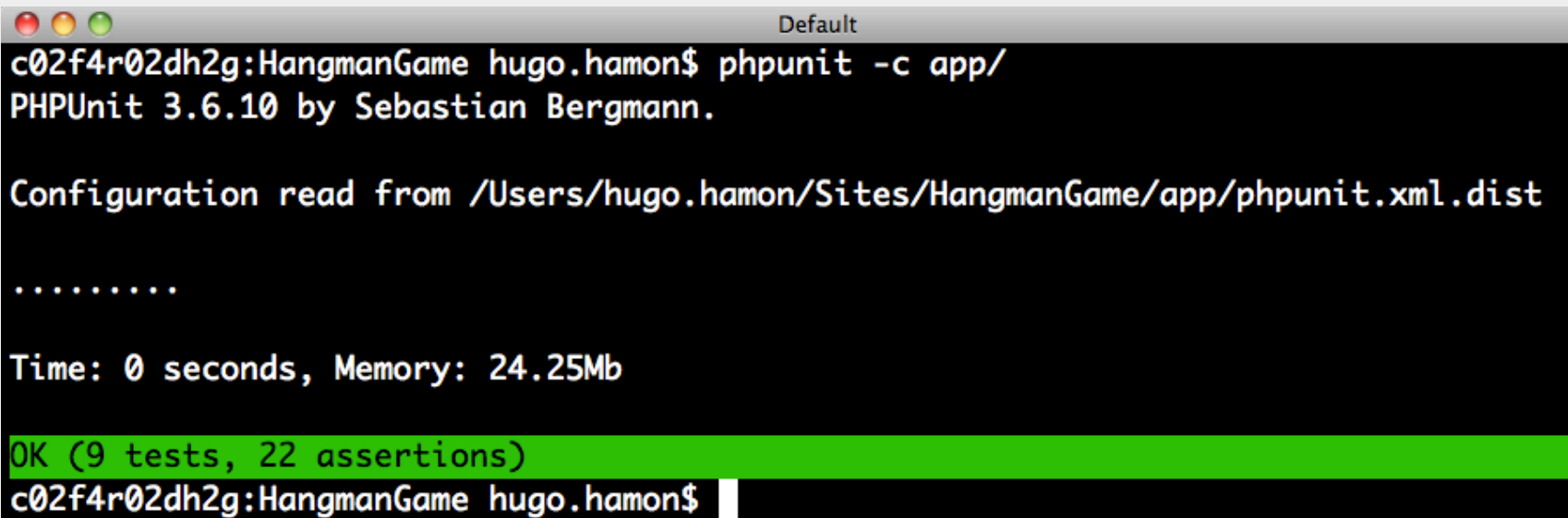
        return $this->client->followRedirect();
    }
}
```

```
class GameControllerTest extends WebTestCase
{
    // ...
    public function testTryWord()
    {
        $this->playWord('php');

        // ...
    }

    public function testGameOverHanged()
    {
        $this->playWord('foo');

        // ...
    }
}
```



A terminal window titled "Default" with a standard macOS window header (red, yellow, green buttons). The terminal shows the execution of PHPUnit in a directory named "c02f4r02dh2g:HangmanGame" owned by "hugo.hamon". The command "phpunit -c app/" is entered. The output includes the PHPUnit version (3.6.10) and the configuration file path. A series of dots indicates the progress of the test suite. The execution time is 0 seconds and memory usage is 24.25Mb. The final result is "OK (9 tests, 22 assertions)", which is highlighted with a green background. The prompt "c02f4r02dh2g:HangmanGame hugo.hamon\$" is visible at the bottom.

```
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

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

        $this->playLetter('P');
        $this->playLetter('H');

        $this->assertRegexp(
            '#You found the word <strong>php</strong>#',
            $this->client->getResponse()->getContent()
        );
    }
}
```

# Scenario: guessing the word by clicking letters

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

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

        $link = $crawler->selectLink($letter)->link();
        $this->client->click($link);

        $this->client->followRedirect();
    }
}
```

```
Default
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.');
```

The letter action still needs to be covered!

# Scenario: game over after max attempts

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

        // Play the same letter until being hanged
        for ($i = 0; $i < Game::MAX_ATTEMPTS; $i++) {
            $this->playLetter('Z');
        }

        $this->assertRegexp(
            "/Oops, you're hanged/",
            $this->client->getResponse()->getContent()
        );
    }
}
```



```
Default
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: resetting an already started game

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

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

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

```
Default
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

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

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

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



Default

```
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.ini

[parameters]

database_driver	=	pdo_mysql
database_host	=	localhost
database_port	=	3306
database_name	=	confoo
database_user	=	root
database_password	=	root





Default

```
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
```

```
c02f4r02dh2g:HangmanGame hugo.hamon$ php app/console generate:doctrine:entity
```

Welcome to the Doctrine2 entity generator

This command helps you generate Doctrine2 entities.

First, you need to give the entity name you want to generate.  
You must use the shortcut notation like **AcmeBlogBundle:Post**.

**The Entity shortcut name:** SensioHangmanBundle:User

Determine the format to use for the mapping information.

**Configuration format (yml, xml, php, or annotation) [annotation]:**

Instead of starting with a blank entity, you can add some fields now.  
Note that the primary key will be added automatically (named **id**).

**Available types:** array, object, boolean, integer, smallint,  
bigint, string, text, datetime, datetimetz, date, time, decimal, float.

**New field name (press <return> to stop adding fields):** username

```
$ php app/console generate:doctrine:entity
```

# Tweaking the generated entity class

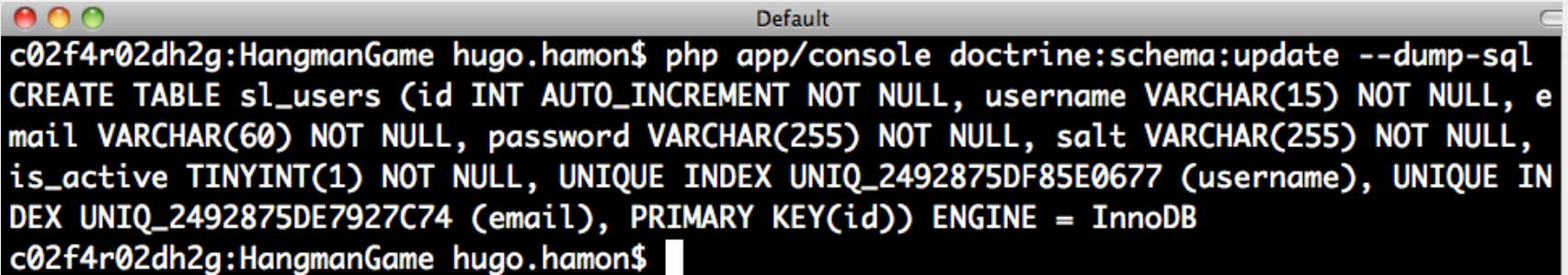
```
/**
 * @ORM\Table(name="sl_users")
 * @ORM\Entity(repositoryClass="Sensio\Bundle\HangmanBundle\Entity\UserRepository")
 */
class User
{
    // ...
    /** @ORM\Column(name="username", type="string", length=15, unique=true) */
    private $username;

    /** @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

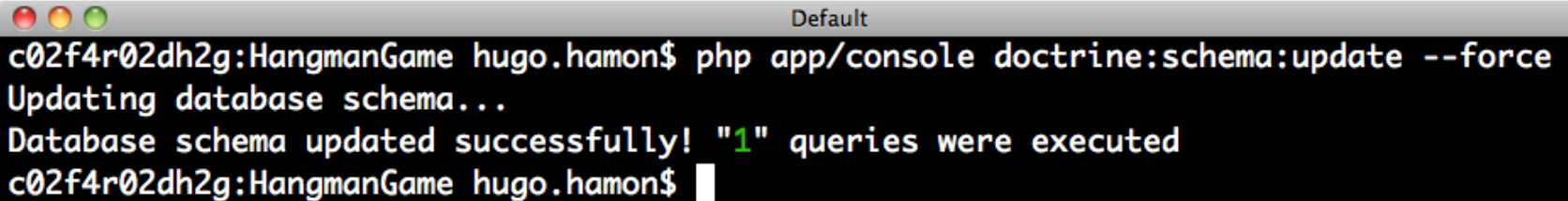


A terminal window titled "Default" showing a command prompt. The prompt is "c02f4r02dh2g:HangmanGame hugo.hamon\$". The command entered is "php app/console doctrine:schema:update --dump-sql". The output is a SQL statement to create a table named "sl\_users". The table has columns: "id" (INT AUTO\_INCREMENT NOT NULL), "username" (VARCHAR(15) NOT NULL), "email" (VARCHAR(60) NOT NULL), "password" (VARCHAR(255) NOT NULL), "salt" (VARCHAR(255) NOT NULL), and "is\_active" (TINYINT(1) NOT NULL). There are two unique indexes: "UNIQ\_2492875DF85E0677" on "username" and "UNIQ\_2492875DE7927C74" on "email". The primary key is "id". The engine is "InnoDB". The prompt returns to "c02f4r02dh2g:HangmanGame hugo.hamon\$".

```
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, email 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 INDEX UNIQ_2492875DE7927C74 (email), PRIMARY KEY(id)) ENGINE = InnoDB
c02f4r02dh2g:HangmanGame hugo.hamon$
```

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

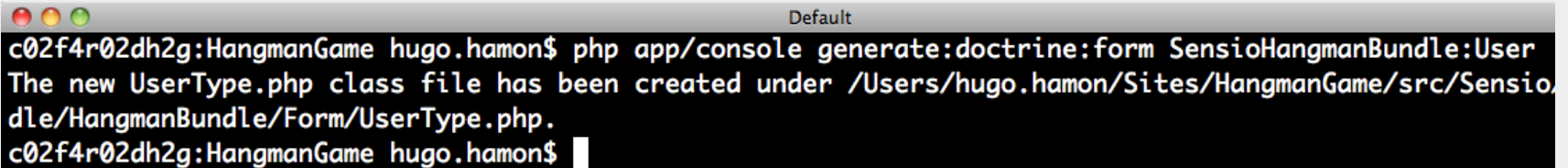
# Generating the schema in the database



```
Default
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



A terminal window with a title bar containing three colored circles (red, yellow, green) and the word "Default". The terminal text shows a command being executed in a directory named "HangmanGame" by a user named "hugo.hamon". The command is "php app/console generate:doctrine:form SensioHangmanBundle:User". The output message states that a new "UserType.php" class file has been created at a specific path. The prompt "c02f4r02dh2g:HangmanGame hugo.hamon\$" is shown again at the end of the output.

```
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:User
```

# Tweaking the form class

```
class UserType extends AbstractType
{
    public function buildForm(FormBuilder $builder, array $options)
    {
        $builder
            ->add('username')
            ->add('email', 'email')
            ->add('password', 'repeated', array(
                'type' => 'password',
                'first_name' => 'password',
                'second_name' => 'confirmation'
            ))
        ;
    }

    public function getName()
    {
        return 'user';
    }
}
```

# Adding the new user controller class

```
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\User;
use Sensio\Bundle\HangmanBundle\Form\UserType;

class UserController extends Controller
{
    public function registrationAction(Request $request)
    {

    }
}
```



```
class UserController extends Controller
{
    /**
     * @Route("/registration", name="registration")
     * @Template()
     */
    public function registrationAction(Request $request)
    {
        $user = new User();
        $form = $this->createForm(new UserType(), $user);

        if ('POST' === $request->getMethod()) {
            $form->bindRequest($request);
            if ($form->isValid()) {
                $em = $this->getDoctrine()->getEntityManager();
                $em->persist($user);
                $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 action="{{ path('registration') }}" method="post">
```

```
        {{ form_widget(form) }}
```

```
        <div>
```

```
            <button type="submit">Register</button>
```

```
        </div>
```

```
    </form>
```

```
{% endblock %}
```

# hangman

[homepage](#) [register](#) [login](#) [top10](#)

## free registration

Username	<input type="text" value="hhamon"/>
Email	<input type="text" value="go.hamon@sensio.com"/>
Password	<input type="password" value="*****"/>
Confirmation	<input type="password" value="*****"/>

## last games

Jan 13 [Ultrices quisque molestie](#)

---

Jan 7 [Neque dolor eget](#)

---

Jan 1 [Sollicitudin interdum](#)

---

Dec 26 [Varius dignissim](#)

## last players

Uma dis suscipit lorem sed luctus.  
Elementum suspendisse tempus  
fermentum omare libero phasellus  
nibh consequat dolore.

# Auto generating the user's salt

```
// ...
```

```
class User
```

```
{
```

```
    // ...
```

```
    public function __construct()
```

```
{
```

```
        $this->isActive = true;
```

```
        $this->salt = sha1(uniqid().microtime().rand(0, 999999));
```

```
}
```

```
    // ...
```

```
}
```

# Adding some input validation

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

// ...
class User
{
    // ...
    /**
     * ...
     * @Assert\NotBlank()
     * @Assert\MinLength(6)
     * @Assert\MaxLength(15)
     * @Assert\Regex("/^[a-zA-Z][a-zA-Z0-9]+$/")
     */
    private $username;
    // ...
}
```

```
class User
{
    // ...

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

    /**
     * ...
     * @Assert\NotBlank()
     * @Assert\MinLength(8)
     */
    private $password;

    // ...
}
```

# Checking for username and email uniqueness

```
// ...
```

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

```
/**
```

```
 * ...
```

```
 *
```

```
 * @UniqueEntity({"username"})
```

```
 * @UniqueEntity({"email"})
```

```
 */
```

```
class User
```

```
{
```

```
    // ...
```

```
}
```

# hangman

[homepage](#) [register](#) [login](#) [top10](#)

## free registration

Username

Email

This value is not valid

This value should not be blank

Password

Confirmation

[Register](#)

## 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



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

  providers:
    players:
      entity: { class: SensioHangmanBundle\User, property: username }

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

    game:
      pattern: ^/
      anonymous: ~
      form_login:
        login_path: /login
        check_path: /auth
        default_target_path: /hangman/
      logout:
        path: /logout
        target: /login

  access_control:
    - { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
    - { path: ^/_internal, roles: IS_AUTHENTICATED_ANONYMOUSLY, ip: 127.0.0.1 }
```

# Registering new security routes

```
# app/config/routing.yml
```

```
login_check:  
    pattern: /auth
```

```
logout:  
    pattern: /logout
```

# The Login Action

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

use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Security\Core\SecurityContext;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;

class SecurityController extends Controller
{

}
```

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

        return array(

            // last username entered by the user

            'last_username' => $session->get(SecurityContext::LAST_USERNAME),
            'error' => $session->get(SecurityContext::AUTHENTICATION_ERROR),
        );
    }
}
```

# The Login Form

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

{% 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

[homepage](#) [register](#) [login](#) [top10](#)

## log-in to your account

Username:  Password:

## last games

Jan 13 [Ultrices quisque molestie](#)

---

Jan 7 [Neque dolor eget](#)

---

Jan 1 [Sollicitudin interdum](#)

---

Dec 26 [Varius dignissim](#)

## last players

Uma dis suscipit lorem sed luctus.  
Elementum suspendisse tempus

# hangman

[homepage](#) [register](#) [login](#) [top10](#)

## 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

Uma dis suscipit lorem sed luctus.  
Elementum suspendisse tempus





# Hashing the user's password at registration

```
class UserType extends AbstractType
{
    public function buildForm(FormBuilder $builder, array $options)
    {
        $builder
            ->add('username')
            ->add('email', 'email')
            ->add('rawPassword', 'repeated', array(
                'type' => 'password',
                'first_name' => 'password',
                'second_name' => 'confirmation'
            ))
        ;
    }
}
```

# Hashing the user's password at registration

```
class UserType extends AbstractType
{
    public function buildForm(FormBuilder $builder, array $options)
    {
        $builder
            ->add('username')
            ->add('email', 'email')
            ->add('rawPassword', 'repeated', array(
                'type' => 'password',
                'first_name' => 'password',
                'second_name' => 'confirmation'
            ))
        ;
    }
}
```

```
class User
{
    // ...

    /**
     * @ORM\Column(...)
     *
     */
    private $password;

    /**
     * @Assert\NotBlank()
     * @Assert\MinLength(8)
     */
    private $rawPassword;

    // ...
}
```

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

class User implements UserInterface
{
    public function setRawPassword($password)
    {
        $this->rawPassword = $password;
    }

    public function getRawPassword()
    {
        return $this->rawPassword;
    }

    public function encodePassword(PasswordEncoderInterface $encoder)
    {
        if (null !== $this->rawPassword) {
            $this->salt = sha1(uniqid().microtime().rand(0, 999999));
            $this->password = $encoder->encodePassword(
                $this->rawPassword,
                $this->salt
            );
            $this->rawPassword = null;
        }
    }
}
```

```
class UserController extends Controller
{
    // ...
    public function registrationAction(Request $request)
    {
        // ...
        if ('POST' === $request->getMethod()) {
            $form->bindRequest($request);
            if ($form->isValid()) {

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

                $em = $this->getDoctrine()->getEntityManager();
                // ...
            }
        }
        // ...
    }
}
```

```
mysql> select * from sl_users\G;
***** 1. row *****
      id: 2
username: hhamon
   email: hugo.hamon@sensio.com
password: uJ5EcfsaKevPqCuhyuxy4VHA67YkRmna50yKRgd/j0xa6aMk0xGukwZQJe+4smD1XTm9VIdSrY0ds5NATe5oiQ==
      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();

    function equals(UserInterface $user);
}
```



# Making a User Entity a Security User

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

// ...
class User implements UserInterface
{
    // ...

    public function getRoles()
    {
        return array('ROLE_PLAYER');
    }
}
```

# Making a User Entity a Security User

```
class User implements UserInterface
{
    // ...

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

    public function equals(UserInterface $user)
    {
        return $this->username === $user->getUsername();
    }
}
```



[http://www.hangman.local/app\\_dev.php/hangman/](http://www.hangman.local/app_dev.php/hangman/)

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



CONFIG



REQUEST



EXCEPTION



EVENTS



LOGS



SECURITY

## Security

<b>Username</b>	hhamon
<b>Authenticated?</b>	yes
<b>Roles</b>	[ROLE_PLAYER]

# Authenticating the user after registration



# Creating a new authentication token

```
$token = new UsernamePasswordToken(  
    'hhamon',  
    'p4SSw0rD',  
    'game',  
    array('ROLE_USER')  
);  
  
$this->get('security.context')->setToken($token);
```

```
// UserController::registrationAction()
```

```
if ($form->isValid()) {
```

```
    // ...
```

```
    $token = new UsernamePasswordToken(
        $user,
        $user->getPassword(),
        'game',
        $user->getRoles()
    );
```

```
    $this->get('security.context')->setToken($token);
```

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

# Restricting access to the game

security:

# ...

access\_control:

- { path: ^/\$, roles: ROLE\_PLAYER }
- { path: ^/hangman, roles: ROLE\_PLAYER }

# ...

# Saving Games in the Database





```
/**
 * @ORM\Table(name="sl_game_data")
 * @ORM\Entity(
 *     repositoryClass="Sensio\Bundle\HangmanBundle\Entity\GameDataRepository »
 * )
 */
class GameData
{
    /**
     * @ORM\Column(type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    private $id;

    /** @ORM\Column(type="string", length=7, unique=true) */
    private $token;

    /** @ORM\Column(type="string", length=25) */
    private $word;

    /** @ORM\Column(type="smallint") */
    private $attempts;
}
```

```
class GameData
{
    // ...

    /**
     * @ORM\Column(name="tried_letters", type="string", length=40, nullable=true)
     */
    private $triedLetters;

    /**
     * @ORM\Column(name="found_letters", type="string", length=40, nullable=true)
     */
    private $foundLetters;

    /** @ORM\Column(type="smallint") */
    private $score;

    /** @ORM\Column(type="string", length=10) */
    private $status;

    /** @ORM\Column(name="start_at", type="datetime") */
    private $startAt;
}
```

```
class GameData
{
    /**
     * @ORM\ManyToOne(targetEntity="User", inversedBy="games")
     *
     */
    private $player;

    public function setPlayer(User $player)
    {
        $this->player = $player;
    }

    public function getPlayer()
    {
        return $this->player;
    }
}
```

# Constructor

```
class GameData
{
    public function __construct()
    {
        $this->triedLetters = array();
        $this->foundLetters = array();
        $this->score = 0;
    }
}
```

```
class GameData
{
    public function setTriedLetters(array $triedLetters)
    {
        $this->triedLetters = implode(',', $triedLetters);
    }

    public function getTriedLetters()
    {
        return explode(',', $this->triedLetters);
    }

    public function setFoundLetters(array $foundLetters)
    {
        $this->foundLetters = implode(',', $foundLetters);
    }

    public function getFoundLetters()
    {
        return explode(',', $this->foundLetters);
    }
}
```

```
/**
 * ...
 * @ORM\HasLifecycleCallbacks()
 */
class GameData
{
    /**
     * @ORM\PrePersist()
     *
     */
    public function preSave()
    {
        $token = sha1(uniqid().microtime().rand(0, 99999));

        $this->token = substr($token, 0, 7);
        $this->startAt = new \DateTime();
    }
}
```

```
use Sensio\Bundle\HangmanBundle\Entity\GameData;
```

```
public function indexAction(Request $request)
{
    //...
    if (!$game = $context->loadGame()) {
        //...
        $word = $game->getWord();
        $user = $this->get('security.context')->getToken()->getUser();

        $data = new GameData();
        $data->setPlayer($user);
        $data->setAttempts($game->getAttempts());
        $data->setTriedLetters($word->getTriedLetters());
        $data->setFoundLetters($word->getFoundLetters());
        $data->setWord($word->getWord());

        $em = $this->getDoctrine()->getEntityManager();
        $em->persist($data);
        $em->flush();
    }

    return array('game' => $game);
}
```

```
mysql> select * from sl_game_data\G;
***** 1. row *****
      id: 1
  player_id: 1
      token: 7816707
      word: software
  attempts: 0
  tried_letters:
  found_letters:
      score: 0
      status: started
  start_at: 2012-02-27 00:07:58
1 row in set (0.00 sec)
```





## Trainings Business Unit

Sensio S.A.

92-98 Boulevard Victor Hugo

92 115 Clichy Cedex

FRANCE

Tél. : +33 140 998 211

[symphony.com](http://symphony.com) - [trainings.sensiolabs.com](http://trainings.sensiolabs.com)