

Mutation testing

Aurelijus Banelis



Aurelijus Banelis

Software developer
aurelijus.banelis.lt
aurelijus@banelis.lt



```
392 359 mutations were generated:
393     280 mutants were killed
394     46 mutants were not covered by tests
395     25 covered mutants were not detected
396     4 fatal errors were encountered
397     4 time outs were encountered
398
```

```
38 Mutation Testing is commencing on 48 files...
383 (.: killed, M: escaped, S: uncovered, E: fatal error, T: timed out)
384
385 .S.S.....E.....M...M..... | 60 (17/48)
386 M.....M.....M.....SS..... | 120 (24/48)
387 ....S.....M...M.....M... | 180 (25/48)
388 .....S.....EE.M.....E.T.SSSSSSSSSSS | 240 (35/48)
389 SSSSSSSSSSSSSSSSSSSSSSS.S...M.MMMM...M.MMM..... | 300 (41/48)
390 .M...M.M..S.MMM..M.....T..TT.....
391
```

```
399 Metrics:
400     Mutation Score Indicator (MSI): 80%
401     Mutation Code Coverage: 87%
402     Covered Code MSI: 92%
```

Mutation testing with PHP

60.0	src/Symfony/Component/CssSelector/Parser/Tokenizer/Tokenizer.php	73
85.71	...nent/CssSelector/Parser/Handler/CommentHandler.php	48
88.33	...Symfony/Component/CssSelector/XPath/Translator.php	298
88.89	...onent/CssSelector/Parser/Handler/StringHandler.php	88



Can You Trust Your Tests?

2015 Vaidas Pilkauskas & Tadas Ščerbinskas

Inspired by Vaidas' presentation

WHY

Why we need mutation testing

WHAT

What it is and how it works

HOW

How it can be used in practice

WHY

Why we need mutation testing

WHAT

What it is and how it works

HOW

How it can be used in practice

It is my code
It should be good



Small mistakes But huge impact


```
if ($number <= $min) {  
    return $colorMin;  
}  
elseif ($number >= $max) {  
    return $colorMax;  
}
```

Let's test everything

```
* @dataProvider providerInterpolate
```

```
*/
```

```
public function testInterpolate($value, $expectedValue)
```

```
{
```

```
    $this->assertEquals(  
        $expectedValue,
```

```
        $this->extension->interpolate($value,
```

```
        -100, 100, 0, 8),
```

```
        "$value <-> $expectedValue"
```

```
    );
```

```
}
```

```
number <= $min) {  
  return $colorMin  
} (number >=
```

Who tests the tester?

```
* @dataProvider providerInterpolate  
*/
```

```
public function testInterpolate($value, $colorMin, $colorMax)
```

```
    {  
        $this->assertEquals($colorMin, $this->interpolate($value, $colorMin, $colorMax));  
    }  
}
```

```
    {  
        $this->assertEquals($colorMax, $this->interpolate($value, $colorMin, $colorMax));  
    }  
}
```

Automation and metrics

WHY

**To be confident about our
code and tests**

WHAT

**What it is and how it
works**

HOW

**How it can be used in
practice**

WHY

Why we need mutation testing

WHAT

What it is and how it works

HOW

How it can be used in practice



Hambug version 1.0.0

Hambug running test suite to generate logs and code coverage data...

Running test suite for 1000 iterations (1000 tests)

Hambug has completed the initial test run successfully.

Total: 100 tests, 100% coverage, 100% pass.

Hambug is analyzing source files...

PHP source files analyzed: 100 files

File Name	Line	Code	Coverage
src/main.php	1	100%
src/main.php	2	100%
src/main.php	3	100%
src/main.php	4	100%
src/main.php	5	100%
src/main.php	6	100%
src/main.php	7	100%
src/main.php	8	100%
src/main.php	9	100%
src/main.php	10	100%
src/main.php	11	100%
src/main.php	12	100%
src/main.php	13	100%
src/main.php	14	100%
src/main.php	15	100%
src/main.php	16	100%
src/main.php	17	100%
src/main.php	18	100%
src/main.php	19	100%
src/main.php	20	100%
src/main.php	21	100%
src/main.php	22	100%
src/main.php	23	100%
src/main.php	24	100%
src/main.php	25	100%
src/main.php	26	100%
src/main.php	27	100%
src/main.php	28	100%
src/main.php	29	100%
src/main.php	30	100%
src/main.php	31	100%
src/main.php	32	100%
src/main.php	33	100%
src/main.php	34	100%
src/main.php	35	100%
src/main.php	36	100%
src/main.php	37	100%
src/main.php	38	100%
src/main.php	39	100%
src/main.php	40	100%
src/main.php	41	100%
src/main.php	42	100%
src/main.php	43	100%
src/main.php	44	100%
src/main.php	45	100%
src/main.php	46	100%
src/main.php	47	100%
src/main.php	48	100%
src/main.php	49	100%
src/main.php	50	100%
src/main.php	51	100%
src/main.php	52	100%
src/main.php	53	100%
src/main.php	54	100%
src/main.php	55	100%
src/main.php	56	100%
src/main.php	57	100%
src/main.php	58	100%
src/main.php	59	100%
src/main.php	60	100%
src/main.php	61	100%
src/main.php	62	100%
src/main.php	63	100%
src/main.php	64	100%
src/main.php	65	100%
src/main.php	66	100%
src/main.php	67	100%
src/main.php	68	100%
src/main.php	69	100%
src/main.php	70	100%
src/main.php	71	100%
src/main.php	72	100%
src/main.php	73	100%
src/main.php	74	100%
src/main.php	75	100%
src/main.php	76	100%
src/main.php	77	100%
src/main.php	78	100%
src/main.php	79	100%
src/main.php	80	100%
src/main.php	81	100%
src/main.php	82	100%
src/main.php	83	100%
src/main.php	84	100%
src/main.php	85	100%
src/main.php	86	100%
src/main.php	87	100%
src/main.php	88	100%
src/main.php	89	100%
src/main.php	90	100%
src/main.php	91	100%
src/main.php	92	100%
src/main.php	93	100%
src/main.php	94	100%
src/main.php	95	100%
src/main.php	96	100%
src/main.php	97	100%
src/main.php	98	100%
src/main.php	99	100%
src/main.php	100	100%

100 statements were generated.

100 statements were killed.

Hambug

One of PHP mutation testing tools

```
"require-dev": {  
  "humbug/humbug": "~1.0@dev"  
},
```

Getting started

```
{  
  "source": {  
    "directories": [  
      "."  
    ],  
    "excludes": [  
      "Tests",  
      "vendor"  
    ]  
  },  
  "timeout": 3,  
  "logs": {  
    "text": ".././../build/bumbug/humbuglog.txt",  
    "json": ".././../build/bumbug/humbuglog.json"  
  }  
}
```

<https://github.com/aurelijusb/symfony/tree/mutation-testing>

Let's test on symfony

coverage 97%

What is Symfony?


```

382 Mutation Testing is commencing on 48 files...
383 (.: killed, M: escaped, S: uncovered, E: fatal error, T: timed out)
384
385 .S.S.....E.....M...M..... | 60 (17/48)
386 M.....M.....M.....SS..... | 120 (24/48)
387 ...S.....M.....M.....M.. | 180 (25/48)
388 .....S.....EE.M.....E.T.SSSSSSSSSSS | 240 (35/48)
389 SSSSSSSSSSSSSSSSSSSSSSSSSSS..S...M.MMM...M.MMM..... | 300 (41/48)
390 .M...M.M..S.MMM..M.....T..TT.....
391

```

Mutation statistics

- **Killed Mutation (.)**: A mutation that caused unit tests to fail which is a positive outcome.
- **Escaped Mutation (M)**: A mutation where the unit tests still passed which is not what we want! Our unit tests should detect any behaviour changes.
- **Uncovered Mutation (S)**: A mutation which occurs on a line not covered by any unit test.
- **Fatal Error (E)**: A mutation created a fatal error.
- **Timeout (T)**: This is where unit tests exceed the allowed timeout

<https://github.com/padraic/humbug>
<https://travis-ci.org/aurelijusb/symfony/builds/188651125>

```
549 1) \Humbug\Mutator\Arithmetic\Multiplication
550 Diff on \Symfony\Component\CssSelector\Node\Specificity::getValue() in /home/travis/build/aurelijusb/sy
551 --- Original
552 +++ New
553 @@ @@
554     {
555 -     return $this->a * self::A_FACTOR + $this->b * self::B_FACTOR + $this->c * self::C_FACTOR;
556 +     return $this->a * self::A_FACTOR + $this->b * self::B_FACTOR + $this->c / self::C_FACTOR;
557     }
558
```

Mutation Details

```
1412     "line": 78,
1413     "diff": "--- Original\n+++ New\n@@\n    {\n        return $this->a * self::A_FACTOR + $this->b * self::B_FACTOR + $this->c * self::C_FACTOR;\n+        return $this->a * self::A_FACTOR +\n        $this->b * self::B_FACTOR + $this->c / self::C_FACTOR;\n    }\n\n    /**\n     * Returns -1 if the object specificity is lower than the argument,\n     * 0 if they are equal, and 1 if the argument is\n     * lower.\n     */
1414     "tests": {
1415         "0": "Symfony\\Component\\CssSelector\\Tests\\Node\\AttributeNodeTest::testSpecificityValue",
1416         "4": "Symfony\\Component\\CssSelector\\Tests\\Node\\ClassNodeTest::testSpecificityValue",
1417         "6": "Symfony\\Component\\CssSelector\\Tests\\Node\\CombinedSelectorNodeTest::testSpecificityValue",
1418         "9": "Symfony\\Component\\CssSelector\\Tests\\Node\\ElementNodeTest::testSpecificityValue",
1419         "12": "Symfony\\Component\\CssSelector\\Tests\\Node\\FunctionNodeTest::testSpecificityValue",
1420         "15": "Symfony\\Component\\CssSelector\\Tests\\Node\\HashNodeTest::testSpecificityValue",
1421         "17": "Symfony\\Component\\CssSelector\\Tests\\Node\\NegationNodeTest::testSpecificityValue",
1422         "18": "Symfony\\Component\\CssSelector\\Tests\\Node\\PseudoNodeTest::testSpecificityValue",
1423         "19": "Symfony\\Component\\CssSelector\\Tests\\Node\\SelectorNodeTest::testSpecificityValue",
1424         "21": "Symfony\\Component\\CssSelector\\Tests\\Node\\SpecificityTest::testValue",
1425         "26": "Symfony\\Component\\CssSelector\\Tests\\Node\\SpecificityTest::testPlusValue",
1426         "31": "Symfony\\Component\\CssSelector\\Tests\\Parser\\ParserTest::testSpecificity"
1427     },
1428     "stderr": "",
1429     "stdout": "TAP version 13"
1430 },
```

```
exec /usr/bin/php7.0 /vendor/.../phpunit
--configuration=/tmp/humbug/phpunit.humbug.xml
--stop-on-failure --tap
```

Under the hood

```
<logging>
  <log type="coverage-php"
    target="/tmp/humbug/coverage.humbug.php"/>
  <log type="coverage-text"
    target="/tmp/humbug/coverage.humbug.txt"/>
  <log type="junit" target="/tmp/humbug/junit.humbug.xml"/>
</logging>
```

```
356 Time: 8.54 seconds, Memory: 18.00MB
```

```
357
```

```
358 OK 1438 tests, 1062 assertions
```

```
406 Time: 3.29 minutes Memory: 16.00MB
```

```
407 Humbug results are being logged as JSON to: ../../../../build/humbug/humbuglog.json
```

```
408 Humbug results are being logged as TEXT to: ../../../../build/humbug/humbuglog.txt
```

```
0 [>-----] < 1 sec
```

```
[Humbug\Exception\InvalidArgumentException]  
File does not exist: /tmp/humbug/coverage.humbug.php
```

Drawbacks

- humbug/humbug 1.0.x-dev requires **phpunit/phpunit ^4.5|^5.0**
- phpunit/phpunit 5.7.5 requires php ^5.6 || ^7.0 -> **your PHP version (5.5.9)** does not satisfy that requirement.

```
4/ $this->length = strlen($source);
```



```
51
```

```
@return bool
```

WHY

Why we need mutation testing

WHAT

Testing framework, change code, run PHPUnit

HOW

How it can be used in practice

WHY

Why we need mutation testing

WHAT

What it is and how it works

HOW

How it can be used in practice


```
392 359 mutations were generated:
393     280 mutants were killed
394     46 mutants were not covered by tests
395     25 covered mutants were not detected
396     4 fatal errors were encountered
397     4 time outs were encountered
398
```

399 Metrics:

```
400     Mutation Score Indicator (MSI): 80%
401     Mutation Code Coverage: 87%
402     Covered Code MSI: 92%
```

Metrics for better release planning

403	-	100.00%	src/Symfony/Component/CssSelector/Parser/Tokenizer/Tokenizer.php		
404	Remember the	100.00%	src/Symfony/Component/CssSelector/Parser/Tokenizer/TokenizerEscaping.php		
453	-	100.00%	src/Symfony/Component/CssSelector/Parser/Tokenizer/TokenizerPattern.php		
454	-	100.00%	src/Symfony/Component/CssSelector/XPath/Extension/AbstractExtension.php		
455	-	100.00%	src/Symfony/Component/CssSelector/XPath/Extension/AttributeExtension.php		
457	-	8.00%	src/Symfony/Component/CssSelector/XPath/Extension/ClassExtension.php		
458	-	100.00%	src/Symfony/Component/CssSelector/XPath/Extension/FunctionExtension.php		
459	AL4	100.00%	src/Symfony/Component/CssSelector/XPath/Extension/HtmlExtension.php		
460	-	96.77%	src/Symfony/Component/CssSelector/XPath/Extension/NodeExtension.php		
461	▲	COVERAGE 100.00%	src/Symfony/Component/CssSelector/XPath/Extension/PseudoClassExtension.php	LINES	RE
462	-	88.33%	src/Symfony/Component/CssSelector/XPath/Translator.php		
	-	60.0	src/Symfony/Component/CssSelector/Exception/SyntaxErrorException.php		73
	-	85.71	src/Symfony/Component/CssSelector/Parser/Handler/CommentHandler.php		48
	-	88.33	src/Symfony/Component/CssSelector/XPath/Translator.php		298
	-	88.89	src/Symfony/Component/CssSelector/Parser/Handler/StringHandler.php		88

```
if ($this->a != $specificity->a) {  
    return $this->a > $specificity->a ? 1 : -1;  
    return $this->a >= $specificity->a ? 1 : -1;  
}  
  
if ($this->b != $specificity->b) {  
    return $this->b > $specificity->b ? 1 : -1;  
}
```

3) \xampp\mutator\ConditionalBoundary\GreaterThan

Diff on \syofony\Component\CssSelector\Node\Specificity::compareTo() in /home/eurelijus/Documents/OPEN-SOURCE

--- Original

+++ New

@@ -0,0

```
if ($this->b != $specificity->b) {  
    return $this->b > $specificity->b ? 1 : -1;  
    return $this->b >= $specificity->b ? 1 : -1;  
}  
  
if ($this->c != $specificity->c) {  
    return $this->c > $specificity->c ? 1 : -1;  
}
```

Edge cases for you

4) \xampp\mutator\ConditionalBoundary\GreaterThan

Diff on \syofony\Component\CssSelector\Node\Specificity::compareTo() in /home/eurelijus/Documents/OPEN-SOURCE

--- Original

+++ New

@@ -0,0

```
if ($this->c != $specificity->c) {  
    return $this->c > $specificity->c ? 1 : -1;  
    return $this->c >= $specificity->c ? 1 : -1;  
}
```

return 0;


```
Diff on \Symfony\Component\CssSelector\Parser\Parser: parseSeries() in /home/aurelijus/Documents/OPEN-SOURCE/symfony/src/...
--- Original
+++ New
@@ @@
     $split = explode('n', $joined);
-    $first = isset($split[0]) ? $split[0] : null;
+    $first = isset($split[1]) ? $split[0] : null;

    return array(
        $first ? ('-' === $first || '+' === $first ? $int($first.'1') : $int($first)) : 1,
        isset($split[1]) && $split[1] ? $int($split[1]) : 0,
    );
}
```

```
6) \Webbug\Mutator\Number\IntegerValue
Diff on \Symfony\Component\CssSelector\Parser\Parser: parseSeries() in /home/aurelijus/Documents/OPEN-SOURCE/symfony/src/...
--- Original
+++ New
@@ @@
     $first ? ('-' === $first || '+' === $first ? $int($first.'1') : $int($first)) : 1,
     isset($split[1]) && $split[1] ? $int($split[1]) : 0,
+    isset($split[0]) && $split[1] ? $int($split[1]) : 0,
    );
}

/**
 * Parses selector nodes.
 */
```

```
7) \Webbug\Mutator\Boolean\LogicalAnd
Diff on \Symfony\Component\CssSelector\Parser\Parser: parseAttributeNode() in /home/aurelijus/Documents/OPEN-SOURCE/symfony/src/...
--- Original
+++ New
@@ @@
 } elseif ($next->isDelimiter(array('"', '$', '*', '-', '[', '1'))
     && $stream->getPeek()->isDelimiter(array('='))
     || $stream->getPeek()->isDelimiter(array('='))
 ) {
     $operator = $next->getValue(). '=';
     $stream->getNext();
}
```

Quality for regression testing

```
75 */
76 public function getValue()
77 {
    return $this->a * self::A_FACTOR + $this->b * self::B_FACTOR + $this->c * self::C_FACTOR;
```

↑ ↓ 🔍 ✕
Hits: 53

```
81 */
82 * Returns -1 if the object specificity is lower than the argument,
83 * 0 if they are equal, and 1 if the argument is lower.
84 *
85 * @param Specificity $specificity
86 *
87 * @return int
88 */
```

What this line should do?

```
mutator
"class": "\\Symfony\\Component\\CssSelector\\...",
"method": "getValue",
"line": 78,
"diff": "--- Original\n+++ New\n@@ @@\n     {\n-         return $this->a * self::A_FACTOR + $this->b * self::B_FACTOR + $this->c * self::C_FACTOR;\n+         return $this->a * self::A_FACTOR + $this->b * self::B_FACTOR + $this->c * self::C_FACTOR;\n     }
"tests": {
    "0": "Symfony\\Component\\CssSelector\\Tests\\Node\\AttributeNodeTest::testSpecificityValue",
    "4": "Symfony\\Component\\CssSelector\\Tests\\Node\\ClassNodeTest::testSpecificityValue",
    "6": "Symfony\\Component\\CssSelector\\Tests\\Node\\CombinedSelectorNodeTest::testSpecificityValue",
    "9": "Symfony\\Component\\CssSelector\\Tests\\Node\\ElementNodeTest::testSpecificityValue",
    "12": "Symfony\\Component\\CssSelector\\Tests\\Node\\FunctionNodeTest::testSpecificityValue",
    "15": "Symfony\\Component\\CssSelector\\Tests\\Node\\HashNodeTest::testSpecificityValue",
    "17": "Symfony\\Component\\CssSelector\\Tests\\Node\\NegationNodeTest::testSpecificityValue",
    "18": "Symfony\\Component\\CssSelector\\Tests\\Node\\PseudoNodeTest::testSpecificityValue",
    "19": "Symfony\\Component\\CssSelector\\Tests\\Node\\SelectorNodeTest::testSpecificityValue",
    "21": "Symfony\\Component\\CssSelector\\Tests\\Node\\SpecificityTest::testValue",
    "26": "Symfony\\Component\\CssSelector\\Tests\\Node\\SpecificityTest::testPlusValue",
    "31": "Symfony\\Component\\CssSelector\\Tests\\Parser\\ParserTest::testSpecificity"
}
```



```

167 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #20"/>
168 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #21"/>
169 </line>
170 <line nr="59"...>
220 <line nr="68"...>
267 <line nr="78">
268 <covered by="Symfony\Component\CssSelector\Tests\Node\AttributeNodeTest::testSpecificityValue with data set #0"/>
269 <covered by="Symfony\Component\CssSelector\Tests\Node\AttributeNodeTest::testSpecificityValue with data set #1"/>
270 <covered by="Symfony\Component\CssSelector\Tests\Node\AttributeNodeTest::testSpecificityValue with data set #2"/>
271 <covered by="Symfony\Component\CssSelector\Tests\Node\AttributeNodeTest::testSpecificityValue with data set #3"/>
272 <covered by="Symfony\Component\CssSelector\Tests\Node\ClassNodeTest::testSpecificityValue with data set #0"/>
273 <covered by="Symfony\Component\CssSelector\Tests\Node\ClassNodeTest::testSpecificityValue with data set #1"/>
274 <covered by="Symfony\Component\CssSelector\Tests\Node\CombinedSelectorNodeTest::testSpecificityValue with data set #0"/>
275 <covered by="Symfony\Component\CssSelector\Tests\Node\CombinedSelectorNodeTest::testSpecificityValue with data set #1"/>
276 <covered by="Symfony\Component\CssSelector\Tests\Node\CombinedSelectorNodeTest::testSpecificityValue with data set #2"/>

```

./phpunit

--coverage-xml=build/xml

```

288 <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testValue with data set #0"/>
290 <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testValue with data set #1"/>
291 <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testValue with data set #2"/>
292 <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testValue with data set #3"/>
293 <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testValue with data set #4"/>
294 <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testPlusValue with data set #0"/>
295 <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testPlusValue with data set #1"/>
296 <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testPlusValue with data set #2"/>
297 <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testPlusValue with data set #3"/>
298 <covered by="Symfony\Component\CssSelector\Tests\Node\SpecificityTest::testPlusValue with data set #4"/>
299 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #0"/>
300 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #1"/>
301 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #2"/>
302 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #3"/>
303 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #4"/>
304 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #5"/>
305 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #6"/>
306 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #7"/>
307 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #8"/>
308 <covered by="Symfony\Component\CssSelector\Tests\Parser\ParserTest::testSpecificity with data set #9"/>

```

Generate random input and test code against predicate

100% mutation proof?

<https://github.com/steos/php-quickcheck>

WHY

Why we need mutation testing

WHAT

What it is and how it works

HOW

Planning, test cases, documentation by tests

References

- <http://www.slideshare.net/AgileLietuva/vaidas-pilkauskas-and-tadas-erbinskas-can-you-trust-your-tests>
- <http://mnapoli.fr/code-coverage-reports-with-github-travis-and/>
- <https://coveralls.io>
- <https://phpunit.de/manual/current/en/code-coverage-analysis.html#code-coverage-analysis.edge-cases>
- <https://akrobat.com/global-installation-of-php-tools-with-composer/>
- <https://travis-ci.org/aurelijusb/symfony/branches>
- <https://github.com/steos/php-quickcheck>

Questions?

Mutation testing with PHP