Empty Stub

```
diff --git geospatial_arginfo.h geospatial_arginfo.h
index 50b6b77..314034f 100644
--- geospatial_arginfo.h
+++ geospatial_arginfo.h
@@ -1,18 +1,4 @@
 /* This is a generated file, edit the .stub.php file instead.
- * Stub hash: 54b0ffc3af871b189435266df516f7575c1b9675 */
+ * Stub hash: 832a6690044ee7ad49c2e929a99000c3d54339de */

-ZEND_BEGIN_ARG_WITH_RETURN_TYPE_INFO_EX(arginfo_test1, 0, 0, IS_VOID, 0)
-ZEND_END_ARG_INFO()

-ZEND_BEGIN_ARG_WITH_RETURN_TYPE_INFO_EX(arginfo_test2, 0, 0, IS_STRING, 0)
-     ZEND_ARG_TYPE_INFO_WITH_DEFAULT_VALUE(0, str, IS_STRING, 0, "\"\"")
-ZEND_END_ARG_INFO()
-
-ZEND_FUNCTION(test1);
-ZEND_FUNCTION(test2);
-
-static const zend_function_entry ext_functions[] = {
-     ZEND_FE(test1, arginfo_test1)
-     ZEND_FE(test2, arginfo_test2)
-     ZEND_FE_END
-};
```

Bring In Algorithms

```
diff --git config.m4 config.m4
index 12a23c5..2f36d47 100644
--- config.m4
+++ config.m4
@@ -14,5 +14,5 @@ PHP_ARG_ENABLE([geospatial],
 if test "$PHP_GEOSPATIAL" != "no"; then
   AC_DEFINE(HAVE_GEOSPATIAL, 1, [ Have geospatial support ])

-  PHP_NEW_EXTENSION(geospatial, geospatial.c, $ext_shared)
+  PHP_NEW_EXTENSION(geospatial, geospatial.c lib/rdp.c lib/geo_array.c, $ext_shared)
 fi

diff --git lib/geo_array.c lib/geo_array.c
diff --git lib/geo_array.h lib/geo_array.h
diff --git lib/rdp.c lib/rdp.c
diff --git lib/rdp.h lib/rdp.h
```

Empty Implementation and Stub

```
diff --git geospatial.c geospatial.c
index 5ac7463..2002d53 100644
--- geospatial.c
+++ geospatial.c
@@ -52,3 +52,7 @@ ZEND_TSRMLS_CACHE_DEFINE()
 # endif
 ZEND_GET_MODULE(geospatial)
 #endif
+
+ZEND_FUNCTION(rdp_simplify)
+{
+}

diff --git geospatial.stub.php geospatial.stub.php
index 4e32832..1453fda 100644
--- geospatial.stub.php
+++ geospatial.stub.php
@@ -3,3 +3,5 @@
 /**
  * @generate-class-entries
  */
+
+function rdp_simplify(array $coordinates, float $epsilon) : array {}

diff --git geospatial_arginfo.h geospatial_arginfo.h
```

Empty Array as Return Value

```
diff --git geospatial.c geospatial.c
index 2002d53..8d03071 100644
--- geospatial.c
+++ geospatial.c
@@ -55,4 +55,5 @@ ZEND_GET_MODULE(geospatial)

 ZEND_FUNCTION(rdp_simplify)
 {
+     array_init(return_value);
 }
```

Implement RDP Simplify

```
diff --git geospatial.c geospatial.c
index 8d03071..0253b53 100644
--- geospatial.c
+++ geospatial.c
@@ -7,6 +7,8 @@
 #include "php.h"
 #include "ext/standard/info.h"
 #include "php_geospatial.h"
+#include "lib/geo_array.h"
+#include "lib/rdp.h"
 #include "geospatial_arginfo.h"


@@ -53,7 +55,109 @@ ZEND_TSRMLS_CACHE_DEFINE()
 ZEND_GET_MODULE(geospatial)
 #endif

+static bool parse_points_pair(zval *coordinates, double *lon, double *lat)
+{
+     HashTable *coords;
+     zval *z_lon, *z_lat;
+
+     if (Z_TYPE_P(coordinates) != IS_ARRAY) {
+           zend_argument_value_error(1, "expected an array coordinate pair, but %s
given", zend_zval_type_name(coordinates));
+           return false;
+     }
+
+     coords = HASH_OF(coordinates);
+
+     if (zend_hash_num_elements(coords) != 2) {
+           zend_argument_value_error(1, "a coordinate pair must have 2 elements, %d
given", zend_hash_num_elements(coords));
+           return false;
+     }
+
+     if ((z_lon = zend_hash_index_find(coords, 0)) == NULL) {
+           zend_argument_value_error(1, "a coordinate pair misses index #0");
+           return false;
+     }
+     if ((z_lat = zend_hash_index_find(coords, 1)) == NULL) {
+           zend_argument_value_error(1, "a coordinate pair misses index #1");
+           return false;
+     }
+
+     convert_to_double_ex(z_lon);
+     convert_to_double_ex(z_lat);
+     *lon = Z_DVAL_P(z_lon);
+     *lat = Z_DVAL_P(z_lat);
+
+     return true;
+}
+
+geo_array *geo_hashtable_to_array(zval *array)
+{
+     geo_array *tmp;
+     zval *entry;
+     int i = 0;
+     double lon, lat;
+
+     tmp = geo_array_ctor(zend_hash_num_elements(Z_ARRVAL_P(array)));
+
+     ZEND_HASH_FOREACH_VAL(Z_ARRVAL_P(array), entry) {
+           if (!parse_points_pair(entry, &lon, &lat)) {
+                 goto failure;
```

```
+           }
+
+           tmp->x[i] = lon;
+           tmp->y[i] = lat;
+           tmp->status[i] = true;
+
+           i++;
+     } ZEND_HASH_FOREACH_END();
+
+     return tmp;
+
+failure:
+     geo_array_dtor(tmp);
+     return NULL;
+}
+
 ZEND_FUNCTION(rdp_simplify)
 {
+     zval *points_array;
+     double epsilon;
+     geo_array *points;
+     int i;
+
+     /* Parse incoming arguments */
+     ZEND_PARSE_PARAMETERS_START(2,2)
+           Z_PARAM_ARRAY(points_array)
+           Z_PARAM_DOUBLE(epsilon)
+     ZEND_PARSE_PARAMETERS_END();
+
+     /* Prepare return value */
      array_init(return_value);
+
+     /* Convert PHP variables into algorithm data structures */
+     points = geo_hashtable_to_array(points_array);
+     if (!points) {
+           return;
+     }
+
+     /* Run algorithm */
+     rdp_simplify(points, epsilon, 0, points->count -1);
+
+     /* Prepare and return result */
+     for (i = 0; i < points->count; i++)
+     {
+           zval pair;
+
+           if (!points->status[i]) {
+                continue;
+           }
+
+           array_init(&pair);
+           add_next_index_double(&pair, points->x[i]);
+           add_next_index_double(&pair, points->y[i]);
+
+           add_next_index_zval(return_value, &pair);
+     }
+
+     /* Clean Up */
+     geo_array_dtor(points);
 }
```

Download Test Data

diff --git tests/geojson-belgium.json tests/geojson-belgium.json

Add Test File

```
diff --git tests/rdp-simplify-001.phpt tests/rdp-simplify-001.phpt
new file mode 100644
index 0000000..b1661bb
--- /dev/null
+++ tests/rdp-simplify-001.phpt
@@ -0,0 +1,19 @@
+--TEST--
+Test for rdp_simplify
+--FILE--
+<?php
+$contents = file_get_contents(__DIR__ . '/geojson-belgium.json');
+$data = json_decode($contents);
+
+$points = $data[0]->geometry->coordinates[0];
+
+$result1 = rdp_simplify($points, 0.001);
+$result2 = rdp_simplify($points, 0.01);
+
+var_dump(count($points), count($result1), count($result2));
+
+?>
+--EXPECT--
+int(1146)
+int(1029)
+int(261)
```

Add Stub and Config Entry for LineString

```
diff --git config.m4 config.m4
index 2f36d47..6c41bb9 100644
--- config.m4
+++ config.m4
@@ -14,5 +14,5 @@ PHP_ARG_ENABLE([geospatial],
 if test "$PHP_GEOSPATIAL" != "no"; then
   AC_DEFINE(HAVE_GEOSPATIAL, 1, [ Have geospatial support ])

-  PHP_NEW_EXTENSION(geospatial, geospatial.c lib/rdp.c lib/geo_array.c, $ext_shared)
+  PHP_NEW_EXTENSION(geospatial, geospatial.c lib/rdp.c lib/geo_array.c
geojson/linestring.c, $ext_shared)
 fi

diff --git geojson/linestring.c geojson/linestring.c
new file mode 100644
index 0000000..0cf4bae
--- /dev/null
+++ geojson/linestring.c
@@ -0,0 +1,8 @@
+#include "php.h"
+
+#include "php_geospatial.h"
+#include "geospatial_arginfo.h"
+
+ZEND_METHOD(Geospatial_GeoJSON_LineString, __construct)
+{
+}

diff --git geospatial.stub.php geospatial.stub.php
index 1453fda..4e84d15 100644
--- geospatial.stub.php
+++ geospatial.stub.php
@@ -1,7 +1,19 @@
 <?php
-
 /**
  * @generate-class-entries
  */

+namespace {
 function rdp_simplify(array $coordinates, float $epsilon) : array {}
+}
+
+namespace Geospatial\GeoJSON {
+
+final class LineString
+{
+    private readonly array $points;
+
+    public function __construct(array $points) {}
+}
+
+}

diff --git geospatial_arginfo.h geospatial_arginfo.h
```

Register Class

```
diff --git geospatial.c geospatial.c
index 0253b53..366d4a8 100644
--- geospatial.c
+++ geospatial.c
@@ -11,6 +11,17 @@
 #include "lib/rdp.h"
 #include "geospatial_arginfo.h"

+zend_class_entry *geospatial_geojson_linestring_ce;
+
+static void geospatial_register_classes(void)
+{
+     geospatial_geojson_linestring_ce = register_class_Geospatial_GeoJSON_LineString();
+}
+
+PHP_MINIT_FUNCTION(geospatial)
+{
+     geospatial_register_classes();
+}

 /* {{{ PHP_RINIT_FUNCTION */
 PHP_RINIT_FUNCTION(geospatial)
@@ -38,7 +49,7 @@ zend_module_entry geospatial_module_entry = {
      STANDARD_MODULE_HEADER,
      "geospatial",                       /* Extension name */
      ext_functions,                      /* zend_function_entry */
-     NULL,                               /* PHP_MINIT - Module initialization */
+     PHP_MINIT(geospatial),              /* PHP_MINIT - Module initialization */
      NULL,                               /* PHP_MSHUTDOWN - Module shutdown */
      PHP_RINIT(geospatial),              /* PHP_RINIT - Request initialization */
      NULL,                               /* PHP_RSHUTDOWN - Request shutdown */
```

Satisfy ZPP Warning

```
diff --git geojson/linestring.c geojson/linestring.c
index 0cf4bae..f3e7bf6 100644
--- geojson/linestring.c
+++ geojson/linestring.c
@@ -5,4 +5,9 @@

 ZEND_METHOD(Geospatial_GeoJSON_LineString, __construct)
 {
+     zval *points;
+
+     ZEND_PARSE_PARAMETERS_START(1,1)
+           Z_PARAM_ARRAY(points)
+     ZEND_PARSE_PARAMETERS_END();
 }
```

Implement Constructor

```
diff --git geojson/linestring.c geojson/linestring.c
index f3e7bf6..3222890 100644
--- geojson/linestring.c
+++ geojson/linestring.c
@@ -10,4 +10,6 @@ ZEND_METHOD(Geospatial_GeoJSON_LineString, __construct)
      ZEND_PARSE_PARAMETERS_START(1,1)
            Z_PARAM_ARRAY(points)
      ZEND_PARSE_PARAMETERS_END();
+
+     zend_update_property(Z_OBJCE_P(ZEND_THIS), Z_OBJ_P(ZEND_THIS), "points",
strlen("points"), points);
 }
```

Implement LineString::getCoordinates

```
diff --git geojson/linestring.c geojson/linestring.c
index 3222890..0539003 100644
--- geojson/linestring.c
+++ geojson/linestring.c
@@ -13,3 +13,14 @@ ZEND_METHOD(Geospatial_GeoJSON_LineString, __construct)

      zend_update_property(Z_OBJCE_P(ZEND_THIS), Z_OBJ_P(ZEND_THIS), "points",
strlen("points"), points);
 }
+
+ZEND_METHOD(Geospatial_GeoJSON_LineString, getCoordinates)
+{
+     ZEND_PARSE_PARAMETERS_NONE();
+
+     RETURN_ZVAL(
+           zend_read_property(Z_OBJCE_P(ZEND_THIS), Z_OBJ_P(ZEND_THIS), "points",
strlen("points"), false, NULL),
+           true,
+           false
+     );
+}

diff --git geospatial.stub.php geospatial.stub.php
index 4e84d15..22adaac 100644
--- geospatial.stub.php
+++ geospatial.stub.php
@@ -14,6 +14,8 @@ final class LineString
      private readonly array $points;

      public function __construct(array $points) {}
+
+     public function getCoordinates() : array {}
 }

 }

diff --git geospatial_arginfo.h geospatial_arginfo.h
```

Add Tests

```
diff --git tests/geojson-linestring-constructor.phpt tests/geojson-linestring-
constructor.phpt
new file mode 100644
index 0000000..9188862
--- /dev/null
+++ tests/geojson-linestring-constructor.phpt
@@ -0,0 +1,32 @@
+--TEST--
+Test for LineString constructor
+--FILE--
+<?php
+$ls = new \Geospatial\GeoJSON\LineString(
+     [
+           [   0, 52.5 ],
+           [ 7.2, 60.3 ]
+     ]
+);
+var_dump($ls);
+?>
+--EXPECTF--

diff --git tests/geojson-linestring-getCoordinates.phpt tests/geojson-linestring-
getCoordinates.phpt
new file mode 100644
index 0000000..8fa713b
--- /dev/null
+++ tests/geojson-linestring-getCoordinates.phpt
@@ -0,0 +1,29 @@
+--TEST--
+Test for LineString::getCoordinates()
+--FILE--
+<?php
+$ls = new \Geospatial\GeoJSON\LineString(
+     [
+           [   0, 52.5 ],
+           [ 7.2, 60.3 ]
+     ]
+);
+var_dump($ls->getCoordinates());
+?>
+--EXPECT--
```

Implement Validation and Test

```
diff --git geojson/linestring.c geojson/linestring.c
index 0539003..a161fa8 100644
--- geojson/linestring.c
+++ geojson/linestring.c
@@ -3,6 +3,63 @@
 #include "php_geospatial.h"
 #include "geospatial_arginfo.h"

+static bool valid_linestring(int argument_nr, zval *points)
+{
+     HashTable *ht = HASH_OF(points);
+     zval *element, *longitude, *latitude;
+     size_t element_num = 0;
+
+     if (zend_hash_num_elements(ht) < 2) {
+           zend_argument_value_error(argument_nr, "must have at least two elements");
+           return false;
+     }
+
+     ZEND_HASH_FOREACH_VAL(ht, element) {
+           if (Z_TYPE_P(element) != IS_ARRAY) {
+                 zend_argument_value_error(argument_nr, "array element #%zd is not an
array", element_num);
+                 return false;
+           }
+
+           if (zend_hash_num_elements(HASH_OF(element)) != 2) {
+                 zend_argument_value_error(argument_nr, "array element #%zd does not
contain a two element coordinate pair", element_num);
+                 return false;
+           }
+
+           longitude = zend_hash_index_find(HASH_OF(element), 0);
+           if (!longitude || (Z_TYPE_P(longitude) != IS_LONG && Z_TYPE_P(longitude) !=
IS_DOUBLE)) {
+                 zend_argument_value_error(argument_nr, "array element #%zd does not
contain a two element coordinate pair", element_num);
+                 return false;
+           }
+           if (Z_TYPE_P(longitude) == IS_LONG && (Z_LVAL_P(longitude) < -180 ||
Z_LVAL_P(longitude) > 180)) {
+                 zend_argument_value_error(argument_nr, "array element #%zd longitude
(%ld) is out of range [-180, 180]", element_num, Z_LVAL_P(longitude));
+                 return false;
+           }
+           if (Z_TYPE_P(longitude) == IS_DOUBLE && (Z_DVAL_P(longitude) < -180 ||
Z_DVAL_P(longitude) > 180)) {
+                 zend_argument_value_error(argument_nr, "array element #%zd longitude
(%f) is out of range [-180, 180]", element_num, Z_DVAL_P(longitude));
+                 return false;
+           }
+
+
+           latitude = zend_hash_index_find(HASH_OF(element), 1);
+           if (!latitude || (Z_TYPE_P(latitude) != IS_LONG && Z_TYPE_P(latitude) !=
IS_DOUBLE)) {
+                 zend_argument_value_error(argument_nr, "array element #%zd does not
contain a two element coordinate pair", element_num);
+                 return false;
+           }
+           if (Z_TYPE_P(latitude) == IS_LONG && (Z_LVAL_P(latitude) < -90 ||
Z_LVAL_P(latitude) > 90)) {
+                 zend_argument_value_error(argument_nr, "array element #%zd latitude
(%ld) is out of range [-90, 90]", element_num, Z_LVAL_P(latitude));
+                 return false;
```

```
+           }
+           if (Z_TYPE_P(latitude) == IS_DOUBLE && (Z_DVAL_P(latitude) < -90 ||
Z_DVAL_P(latitude) > 90)) {
+                   zend_argument_value_error(argument_nr, "array element #%zd latitude (%f)
is out of range [-90, 90]", element_num, Z_DVAL_P(latitude));
+                   return false;
+           }
+
+           ++element_num;
+     } ZEND_HASH_FOREACH_END();
+
+     return true;
+}
+
 ZEND_METHOD(Geospatial_GeoJSON_LineString, __construct)
 {
      zval *points;
@@ -11,6 +68,10 @@ ZEND_METHOD(Geospatial_GeoJSON_LineString, __construct)
           Z_PARAM_ARRAY(points)
      ZEND_PARSE_PARAMETERS_END();

+     if (!valid_linestring(1, points)) {
+           RETURN_THROWS();
+     }
+
      zend_update_property(Z_OBJCE_P(ZEND_THIS), Z_OBJ_P(ZEND_THIS), "points",
strlen("points"), points);
 }

diff --git tests/geojson-linestring-contructor-errors.phpt tests/geojson-linestring-
contructor-errors.phpt
new file mode 100644
index 0000000..4d5f76e
--- /dev/null
+++ tests/geojson-linestring-contructor-errors.phpt
@@ -0,0 +1,41 @@
+--TEST--
+Test for LineString constructor errors
+--FILE--
+<?php
+$tests = [
+     [],
+     [ [ 0, 52.5 ] ],
+     // [ [ 0, 52.5 ], [ 7.1, 59.6 ] ], correct
+     [ [ 52.5 ], [ 7.1, 59.6 ] ],
+     [ [ 0, "52.5" ], [ 7.1, 59.6 ] ],
+     [ [ 0, 52.5 ], [ "7.1", 59.6 ] ],
+     [ [ false, 52.5 ], [ 7.1, 59.6 ] ],
+     [ [ 0, 52.5 ], [ new stdClass, 59.6 ] ],
+
+     [ [ -190, 52.5 ], [ 7.1, 59.6 ] ],
+     [ [ +190, 52.5 ], [ 7.1, 59.6 ] ],
+     [ [ 0, 52.5 ], [ 7.1, -99.6 ] ],
+     [ [ 0, 52.5 ], [ 7.1, +99.6 ] ],
+];
+
+foreach ($tests as $test) {
+     try {
+           $ls = new Geospatial\GeoJSON\LineString($test);
+     } catch (ValueError $e) {
+           echo get_class($e), ': ', $e->getMessage(), "\n";
+     }
+}
+
+?>
+--EXPECTF--
```

Implement LineString::simplify

```
diff --git geojson/linestring.c geojson/linestring.c
index a161fa8..dd0739d 100644
--- geojson/linestring.c
+++ geojson/linestring.c
@@ -3,6 +3,8 @@
 #include "php_geospatial.h"
 #include "geospatial_arginfo.h"

+#include "../lib/rdp.h"
+
 static bool valid_linestring(int argument_nr, zval *points)
 {
      HashTable *ht = HASH_OF(points);
@@ -85,3 +87,49 @@ ZEND_METHOD(Geospatial_GeoJSON_LineString, getCoordinates)
            false
      );
 }
+
+ZEND_METHOD(Geospatial_GeoJSON_LineString, simplify)
+{
+     double epsilon;
+     geo_array *points;
+     int i;
+
+     ZEND_PARSE_PARAMETERS_START(1,1)
+           Z_PARAM_DOUBLE(epsilon)
+     ZEND_PARSE_PARAMETERS_END();
+
+     /* Prepare return value */
+     array_init(return_value);
+
+     /* Convert PHP variables into algorithm data structures */
+     points = geo_hashtable_to_array(
+           zend_read_property(Z_OBJCE_P(ZEND_THIS), Z_OBJ_P(ZEND_THIS), "points",
strlen("points"), false, NULL)
+     );
+     if (!points) {
+           return;
+     }
+
+     /* Run algorithm */
+     rdp_simplify(points, epsilon, 0, points->count -1);
+
+     /* Prepare and return result */
+     for (i = 0; i < points->count; i++)
+     {
+           zval pair;
+
+           if (!points->status[i]) {
+                continue;
+           }
+
+           array_init(&pair);
+           add_next_index_double(&pair, points->x[i]);
+           add_next_index_double(&pair, points->y[i]);
+
+           add_next_index_zval(return_value, &pair);
+     }
+
+     /* Clean Up */
+     geo_array_dtor(points);
+
+}
+
```

```
diff --git geospatial.stub.php geospatial.stub.php
index 22adaac..261de83 100644
--- geospatial.stub.php
+++ geospatial.stub.php
@@ -16,6 +16,7 @@ final class LineString
      public function __construct(array $points) {}

      public function getCoordinates() : array {}
+     public function simplify(float $epsilon) : array {}
 }

 }

diff --git geospatial_arginfo.h geospatial_arginfo.h

diff --git tests/geojson-linestring-simplify.phpt tests/geojson-linestring-simplify.phpt
new file mode 100644
index 0000000..5670e03
--- /dev/null
+++ tests/geojson-linestring-simplify.phpt
@@ -0,0 +1,21 @@
+--TEST--
+Test for LineString::simplify
+--FILE--
+<?php
+$contents = file_get_contents(__DIR__ . '/geojson-belgium.json');
+$data = json_decode($contents);
+
+$points = $data[0]->geometry->coordinates[0];
+
+$ls = new Geospatial\GeoJson\LineString($points);
+
+$result1 = $ls->simplify(0.001);
+$result2 = $ls->simplify(0.01);
+
+var_dump(count($points), count($result1), count($result2));
+
+?>
+--EXPECT--
+int(1146)
+int(1029)
+int(261)
```

Convert to Returning New LineString instead of Array

```
diff --git geojson/linestring.c geojson/linestring.c
index dd0739d..d0900aa 100644
--- geojson/linestring.c
+++ geojson/linestring.c
@@ -5,6 +5,8 @@

 #include "../lib/rdp.h"

+extern zend_class_entry *geospatial_geojson_linestring_ce;
+
 static bool valid_linestring(int argument_nr, zval *points)
 {
      HashTable *ht = HASH_OF(points);
@@ -92,6 +94,7 @@ ZEND_METHOD(Geospatial_GeoJSON_LineString, simplify)
 {
      double epsilon;
      geo_array *points;
+     zval simplified_points;
      int i;

      ZEND_PARSE_PARAMETERS_START(1,1)
@@ -99,7 +102,7 @@ ZEND_METHOD(Geospatial_GeoJSON_LineString, simplify)
      ZEND_PARSE_PARAMETERS_END();

      /* Prepare return value */
-     array_init(return_value);
+     array_init(&simplified_points);

      /* Convert PHP variables into algorithm data structures */
      points = geo_hashtable_to_array(
@@ -125,11 +128,15 @@ ZEND_METHOD(Geospatial_GeoJSON_LineString, simplify)
             add_next_index_double(&pair, points->x[i]);
             add_next_index_double(&pair, points->y[i]);

-            add_next_index_zval(return_value, &pair);
+            add_next_index_zval(&simplified_points, &pair);
      }

      /* Clean Up */
      geo_array_dtor(points);

+     /* Return values in new object */
+     object_init_ex(return_value, geospatial_geojson_linestring_ce);
+     zend_update_property(Z_OBJCE_P(return_value), Z_OBJ_P(return_value), "points",
strlen("points"), &simplified_points);
+     zval_ptr_dtor(&simplified_points);
 }

diff --git geospatial.stub.php geospatial.stub.php
index 261de83..861e44a 100644
--- geospatial.stub.php
+++ geospatial.stub.php
@@ -16,7 +16,7 @@ final class LineString
      public function __construct(array $points) {}

      public function getCoordinates() : array {}
-     public function simplify(float $epsilon) : array {}
+     public function simplify(float $epsilon) : LineString {}
 }

 }
diff --git geospatial_arginfo.h geospatial_arginfo.h
```

```
diff --git tests/geojson-linestring-simplify.phpt tests/geojson-linestring-simplify.phpt
index 5670e03..d8a32a3 100644
--- tests/geojson-linestring-simplify.phpt
+++ tests/geojson-linestring-simplify.phpt
@@ -12,10 +12,18 @@ $ls = new Geospatial\GeoJson\LineString($points);
 $result1 = $ls->simplify(0.001);
 $result2 = $ls->simplify(0.01);

-var_dump(count($points), count($result1), count($result2));
+var_dump(get_class($result1), get_class($result2));
+
+var_dump(
+    count($points),
+    count($result1->getCoordinates()),
+    count($result2->getCoordinates())
+);

 ?>
 --EXPECT--
```