

Источник - Сергей Звездин: персональный блог (<http://blog.zvezdin.com/ru/237>)

## Как использовать GestureService и добавить обработчики жестов из кода

---

В [Windows Phone 7 Toolkit](#) доступен объект для обработки жестов – GestureService. Как известно, этот сервис позволяет обрабатывать стандартные для Windows Phone 7 жесты. Однако, многие примеры использования GestureService сводятся к применению его в XAML. В этой небольшой заметке я покажу как использовать GestureService из кода.

Для начала работы нам необходимо подключить Windows Phone 7 Toolkit к проекту. Для этого нужно добавить ссылку на сборку *Microsoft.Phone.Controls.Toolkit.dll*.

### Как использовать GestureService из кода

Для использования GestureService из кода, нам нужно получить доступ к объекту GestureListener, который будет ассоциирован с некоторой областью на экране. В рамках этой области (*source*) будут отслеживаться наши жесты.

```
1 | var listener = GestureService.GetGestureListener(source);
```

Объект GestureListener готов к работе, теперь можно обрабатывать его жесты.

### Как добавить обработчик жестов

Для обработки конкретного жеста следует подписаться на соответствующее событие объекта GestureListener:

```
1 | listener.Tap += OnTapHandler;
```

Безусловно можно подписаться и на другие жесты, используя аналогичный подход.

### Где разместить код обработки жестов

Подписываться на события жестов можно в любой момент после создания соответствующих элементов управления. Например, это можно делать в конструкторе страницы (конечно, более правильно это делать во ViewModel, но сейчас мы не будем усложнять пример). Например, следующий код создает новый элемент управления (Rectangle) на странице и добавляет для него

## обработчик жестов.

```
01 public partial class Page1 : PhoneApplicationPage
02 {
03     Rectangle _source;
04
05     public MainPage()
06     {
07         InitializeComponent();
08         _source = new Rectangle {Height = 200, Width = 200, Fill = new
SolidColorBrush(Colors.Green)};
09         ContentPanel.Children.Add(_source);
10
11         var listener = GestureService.GetGestureListener(_source);
12         listener.Tap += OnTapHandler;
13     }
14
15     private void OnTapHandler(object sender, GestureEventArgs e)
16     {
17         _source.Fill = new SolidColorBrush(Colors.Yellow);
18     }
19 }
```