

УДК 004

## ИСПОЛЬЗОВАНИЕ SIGNALR ДЛЯ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ

*М.В. Смалёха*

*Научный руководитель: Е.А. Кочегурова, к.т.н., профессор, каф. АиКС, ИК, ТПУ  
Томский политехнический университет, 634050, Россия, г. Томск, пр. Ленина, 30  
E-mail: maksim\_smal@mail.ru*

*This article describes the capabilities of the technology SignalR. SignalR is a powerful library for creating interactive web-real-time applications for MVC.*

**Keywords:** web-application, SignalR, Javascript, MVC.

**Ключевые слова:** web-приложение, SignalR, Javascript, MVC.

При современном уровне развития интернет технологий корпоративные веб-порталы, состоящие из нескольких модулей, охватывающих всю деятельность компании, набирают всё большую популярность. Веб-приложения имеют несравнимые преимущества в процессе разработки, развёртывания и поддержки пользователей, по сравнению с десктопными решениями. Одним из таких модулей является модуль коммуникаций, который имеет немаловажное значение в трудовой деятельности сотрудников, поскольку коммуникации играют важную роль с точки зрения ведения проектов. Основным компонентом модуля коммуникаций является чат. Соответственно, возникает проблема, как обеспечить интерактивность веб-страницы, позволяющую отсылать данные на сервер, и получать данные с сервера в асинхронном режиме.

ASP.NET SignalR – это библиотека для ASP.NET разработчиков, которая служит для разработки компонентов, работающих в реальном времени. В зависимости от поддержки клиента библиотека SignalR использует все транспортные технологии: Web Sockets, Server-Sent Event, Long Polling, Polling. Технология Web Sockets является наиболее предпочтительной технологией для SignalR, поэтому она используется первой. [1]

Для работы с SignalR на сервере создаётся хаб (Hub). Хаб является более высокоуровневым средством передачи данных, так как он построен на базе Connection API, и это позволяет клиенту и серверу вызывать методы друг друга напрямую, как будто это локальные методы. [2]

Каждое подключение на сервере имеет свой ConnectionId – значение типа Guid, с которым можно в дальнейшем работать, как с конкретным клиентом. SignalR позволяет очень гибко манипулировать подключениями клиентов: работать с конкретным клиентом, объединять клиентов в группы, работать только с вызвавшим клиентом, работать со всеми, кроме вызвавшего, и т. д. Также можно переопределять стандартные методы библиотеки OnConnected() и OnDisconnected(bool), что позволяет иметь список текущих подключений. Ниже приведен пример такого переопределения.

```
namespace MvcApplication3.Hubs
{
    [HubName("MyHub")]
    public class MyHub : Hub
    {
        Static int Number = 0;
        public void Hello()
        {
            var id = Context.ConnectionId;
            Clients.All.hello("Hello World");
        }
    }
}
```

```
public override Task OnConnected()    {
    Number++;
    return base.OnConnected();
}
public override Task OnDisconnected(bool stopCalled)    {
    Number--;
    return base.OnDisconnected(stopCalled);
}
}
```

Когда серверный код вызывает метод клиента, то через активный транспортный протокол отправляется пакет данных, который содержит имя метода и его параметры (если параметр является объектом, он сериализуется в JSON). Затем клиент ведет поиск метода в своем коде по присланному имени вызываемого метода, и если находит, то метод выполняется с десериализованными параметрами из присланного пакета. [3]

Все это возможности делают технологию SignalR наиболее подходящей для разработки чата. В отличие от Web Sockets можно определять неограниченное количество функций, как на сервере, так и на клиенте, что делает SignalR более гибким.

```
$(function () {
    var hub = $.connection.MyHub;
    hub.client.Hello = function(hello){
        alert("hello");
    }
    $.connection.hub.start().done(function()    {
        hub2.server.hello();
    });
})
```

Вместе с JavaScript, SignalR становится мощным инструментом для разработки интерактивных web-приложений. Однако стоит отметить, что при реализации числа большого функциональных частей проекта реализованных с помощью SignalR происходит разделение кода, поскольку в проектах MVC логика взаимодействия находится в контроллерах. Поэтому SignalR стоит использовать только тогда, когда он действительно необходим, а в обычных ситуациях стоит отдавать предпочтение AJAX.

В результате работы был реализован модуль чата, который поддерживает личное общение двух пользователей и групповой общение. Реализована гибкая система уведомлений о новых сообщениях, которая появляется и исчезает на всех страницах пользователя (дата обращения: 11.02.15).

### Список литературы

1. ASP.NET SignalR [Электронный ресурс]: Официальный сайт. URL: <http://www.asp.net/signalr> (дата обращения: 11.02.15).
2. Что такое SignalR [Электронный ресурс]: Блог о разработке ASP.NET MVC. URL: [http://www.codehint.ru/articles/2014-10-29\\_introduction\\_signalr](http://www.codehint.ru/articles/2014-10-29_introduction_signalr) (дата обращения: 11.02.15).
3. Использование SignalR в ASP.NET MVC [Электронный ресурс]: Блог о разработке веб-порталов. URL: <http://www.waveaccess.ru/blog/2014/dec/06/signalr/> (дата обращения: 11.02.15).