

# TulaDev.NET

## *jQuery как путь к RIA*

*Rich User Interface на основе  
jQuery + jQueryUI + MVC + HTML5*

Андрей Кулешов  
Codemasters International  
Senior C# Developer, Team Lead

[akuleshov@codereign.net](mailto:akuleshov@codereign.net) |  
<http://www.codemastersintl.com/Blogs/Andrey-Kuleshov>

# Что будет:

- Понятия Rich User Interface и RIA
- jQuery как средство работы с UI
- Введение в библиотеку. Понятие селекторов
- Обработка событий. Анимация в jQuery
- jQuery UI как библиотека пользовательских элементов на стороне клиента
- Ajax в jQuery. Взаимодействие с ASP.NET MVC
- Расширяемость jQuery. Обзор существующих плагинов

# Rich User Interface

- Elaborate, having complex formatting, multimedia, or depth of interaction (Продуманный, имеющий сложное форматирование, мультимедиа или глубину взаимодействия с пользователем)
- Изначально начал появляться на десктопных приложениях
- Начал активно внедряться в Web с появлением Flash технологии
- Однако сейчас наступил период, когда возможностей самих браузеров становится достаточно



- JQuery отделяет поведение от структуры HTML

Функция `$` (aka `jQuery()` function) возвращает:

- а) JavaScript –объект, содержащий массив DOM-элементов
- б) в том порядке, в каком они были найдены в документе
- в) Соответствующих переданному **селектору**
- г) этот объект известен как `wrapper` или `wrapped set`

```
$("div.someClass").show();
```

Большинство методов объекта-обертки возвращают ту же самую группу элементов => возможно объединение в цепь вызовов

```
$("div.someClass").show().addClass("SomeOtherClass");
```

# Преимущества jQuery

- Не пытается создать свой новый язык или вариант языка, но использует комбинацию CSS, HTML и классического JavaScript
- Легковесный (минимизированная сжатая версия – 29 кб)
- Легкий для изучения

```
$('#something').hide().css('background', 'red').show();
```

- Большое количество готовых плагинов
- Легко расширяемый
- Поддерживается Microsoft (IntelliSense, CDN)
- Большое количество обучающих материалов
- Помогает писать *Unobstrutive JavaScript*

# Событие

- В классическом JavaScript проблематично запустить какой-то код после загрузки документа – нет единой событийной модели

```
$(document).ready(function () {  
    $("div.someClass").show();  
});  
  
$(function () {  
    $("div.someClass").show();  
});
```

# Селекторы



- В основе jQuery лежит selector engine (Sizzle)
- Может быть использован для выборки элементов по:
  - имени (идентификатору);
  - атрибутам;
  - позиции;
- Многократно перегруженный оператор `$()` используется как для осуществления выборки, так и для создания новых

```
var div = $("<div id='helloDiv'>Hello!</div>");
```

# Разнообразие селекторов

```
$("p")
```

Все параграфы

```
$("p a.someClass")
```

Все ссылки, находящиеся внутри параграфов и имеющие класс someClass

```
$("ul.someList > li > a")
```

Все ссылки внутри элементов списка, если сам список имеет класс someClass

```
$("a[href*='http://tuladev.net']")
```

Все ссылки на TulaDev.NET

```
$("span[class^='some']")
```

Все span-элементы, имеющие класс, начинающийся на «some»

```
$("span[class]")
```

Все span-элементы, имеющие хоть какой-нибудь класс

# Демо

## Простые селекторы





# События в jQuery

- `.click(handler)`, `.dblclick(handler)`, `.mouseenter(handler)` – упрощение синтаксиса
- `.bind('eventname', handler)` - `.unbind('eventname', handler)`
  - через него реализуются все одномоментные связывания
  - события обрабатываются лишь для тех элементов, которые уже присутствуют в DOM на момент вызова
- `.live('eventname', handler)` – `.die('eventname', handler)`
  - работают для всех элементов, включая динамически добавленные
  - используют «всплывание» события до уровня корня дерева элементов
- `.delegate(selector, eventType, handler)`
  - используют «всплывание» до уровня элемента, на котором были вызваны

# Демо jQuery events



# Анимация в jQuery

- .fadeIn, .fadeOut
- .animate

```
$('#book').animate({  
  opacity: 0.25,  
  left: '+=50',  
  height: 'toggle'  
}, 5000, function() {  
  // Animation complete.  
});
```

# Демо Анимация



# jQuery UI

- обширная библиотека контролов
- кроссбраузерность (почти всегда)
- только client-side

# Демо jQuery UI

<http://jqueryUI.com/demos>



# jQuery + Ajax + Mvc

- HTML vs JSON
- Способы получения контента с сервера
- PartialView vs JsonResult

```
{
  "firstName": "Иван",
  "lastName": "Иванов",
  "address": {
    "streetAddress": "Московское ш., 101, кв.101",
    "city": "Ленинград",
    "postalCode": 101101
  },
  "phoneNumbers": [
    "812 123-1234",
    "916 123-4567"
  ]
}
```

# Демо Анимация





Вопросы?  
Внимательно слушаю 😊

Андрей Кулешов

Codemasters International

Senior C# Developer, Team Lead

[akuleshov@codereign.net](mailto:akuleshov@codereign.net) |

<http://www.codemastersintl.com/Blogs/Andrey-Kuleshov>

# ИСТОЧНИКИ

- jQuery documentation  
<http://docs.jquery.com/>
- jQuery UI  
<http://jqueryui.com/demos/>
- What You, as an ASP.NET Developer, Need to Know About jQuery *Gill Cleeren*  
<http://msteched.com>

Андрей Кулешов

Codemasters International

Senior C# Developer, Team Lead

[akuleshov@codereign.net](mailto:akuleshov@codereign.net) |

<http://www.codemastersintl.com/Blogs/Andrey-Kuleshov>