AJAX-чат на prototype.js

RAD с помощью библиотеки prototype.js на примере AJAX-чата

мастер-класс

Александр Шуркаев
GooDoo IT / Newmedia Stars
www.prototypejs.ruwww.prototypejs.ru,
htmlcoder.visions.ru
alshur@ya.ru

Почему чат на AJAX’е?

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Передача малого количества данных
- Быстрое обновление данных
- Нет необходимости в наличии Java-машины и дополнительного ПО
- Не используются дополнительные порты
- AJAX — это модно :-)

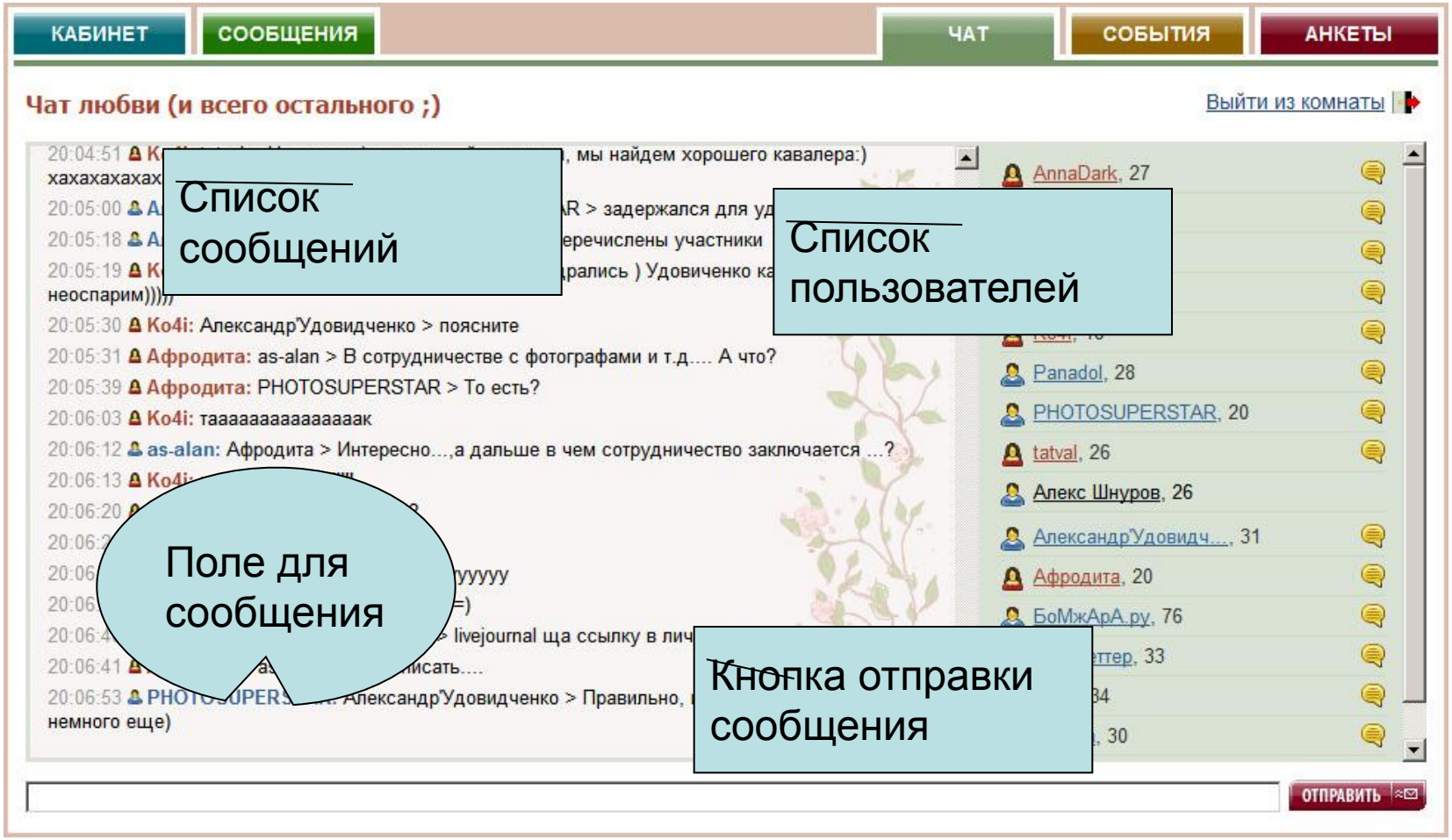
Почему prototype.js?

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Удобная и кроссбраузерная работа с AJAX’ом, DOM’ом
- Готовые расширения JavaScript-объектов
- ООП подход
- Самый популярный framework

Базовый интерфейс чата

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ



Передача данных в AJAX-чате

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Что такое JSON и чем он хорош?**
  - Передача сериализованных объектов и массивов
  - Native-поддержка в prototype.js
  - JSON мало «весит»
- **Вообще, есть и другие способы**
  - XML
  - HTML-куски
  - Текст
  - HTTP-заголовки (реализация веб-сервисов)

AJAX-запросы к серверу

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Список новых сообщений
- Список всех пользователей
- Отправка сообщения

AJAX-запросы к серверу

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Список новых сообщений**

Запрос (GET)	Ответ (JSON)
/ajax/chat/data.html?room_id=1&stamp=23&rand=0.04169033924254095	{"stamp": 24,"messages": [{"time": 1176209243, "text": "Превед!","user_id": 5005}, {"time": 1176209999,"text": "Ха-ха", "user_id": 10}]}

AJAX-запросы к серверу

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Список всех пользователей**
  - Получение дополнительных пользователей, нужных для отрисовки списка сообщений

Запрос (POST)	Ответ (JSON)
/ajax/chat/users.html?room_id=1&extra_user_ids=5005,15&rand=0.12987317360431827	[{"video_count": 0, "is_extra_user": 1, "is_contact": 0, "sex": 1, "age": 34, "login": "petya", "username": "PPP", "is_friend": 0, "is_ignored": 0, "smallavatar": "/images/0000/g/o/petya/photo/album1/12345.27192_21-sm.jpg", "id": 5005}, {"video_count": 0, "is_extra_user": 1, "is_contact": 1, "sex": 0, "age": 18, "login": "ko4i", "username": "Ko4i-МОЗГ", "is_friend": 0, "is_ignored": 0, "smallavatar": "/images/0000/k/o/ko4i/photo/album1/12345.76161_3221-sm.gif", "id": 10}]

AJAX-запросы к серверу

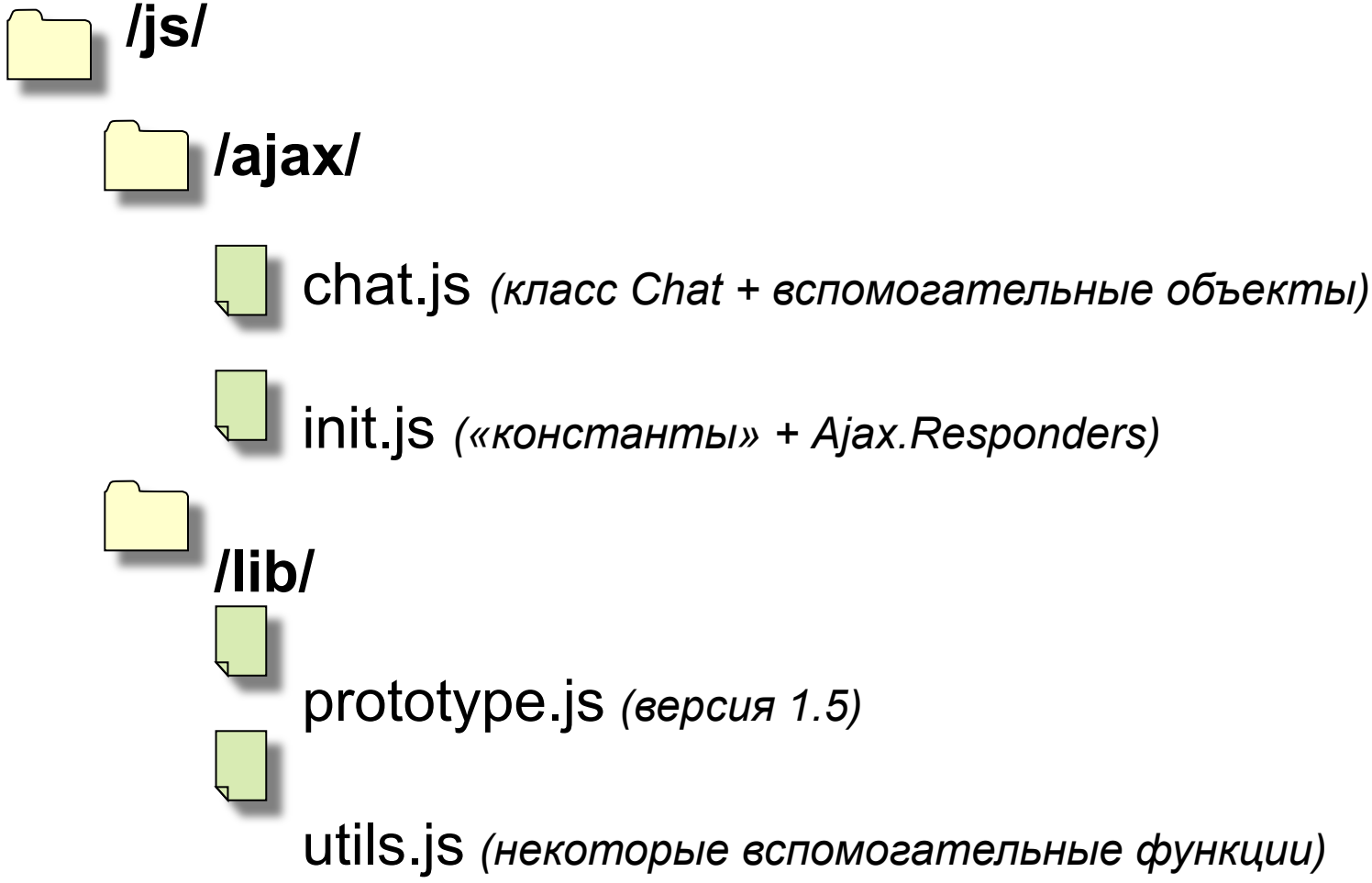
РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Отправка сообщения**
  - Объединение отправки и показа новых сообщений
  - Буферизация сообщений

Запрос (POST)	Ответ (JSON)
/ajax/chat/msg.html?msg=Hi!&room_id=1&stamp=554&rand=0.9728927922458538	{"stamp": 555, "messages": [{"time": 1176221022, "text": “Hi!", "user_id": 99}]}

Структура JavaScript-модулей

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ



JS-классы и объекты чата

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Ключевые используемые prototype.js-классы**
  - PeriodicalExecuter
  - Ajax.Request
  - Template
- **Класс Chat**
- **Вспомогательные объекты**
  - ChatHelpers
  - ChatAjaxHelpers
  - ChatStrings

JS-классы и объекты чата

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **PeriodicalExecuter**

```
startListTimer: function(type){
   if (typeof this[type + '_list_timer'] == 'undefined')
   this[type + '_list_timer'] = new
   PeriodicalExecuter(this[('get-' + type +
   '-list').camelize()].bind(this), this.options[type +
   '_updater_frequency']);
   else this[type + '_list_timer'].registerCallback();
},
stopListTimer: function(type){
   if (typeof this[type + '_list_timer'] != 'undefined')
   this[type + '_list_timer'].stop();
}
```

JS-классы и объекты чата

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Ajax.Request**

```
getMsgsList: function(){
   this.stopListTimer('msgs');
   new Ajax.Request(
     this.options.msgs_url,
     {
          method: 'get',
          parameters: 'room_id=' + this.room_id + '&stamp=' +
   this.stamp + ChatAjaxHelpers.addExtraAjaxParams(),
          onComplete: this.getMsgsListDone.bind(this),
          onFailure: ChatAjaxHelpers.handleAjaxFailure.bind(this,
   this.options.frontend_base_url)
     });
}
```

JS-классы и объекты чата

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Template**

```
msg_default_tpl: new Template('<p class="#{css_class}"><span
    class="time">#{time}</span> <a href="#{profile_link}"
    target="_blank" title="Посмотреть профиль (в новом
    окне)">#{user_ico}</a><span
    class="#{username_css_class}">#{username}:</span>
    #{text}</p>'),
msg_system_tpl: new Template('<p class="system-msg"><span
    class="time">#{time}</span> #{text}</p>'),
msgListItemTemplate: function(vars, type){
    var type = type || 'default';
    return ChatHelpers['msg_' + type + '_tpl'].evaluate(vars);
}
```

JS-классы и объекты чата

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Класс Chat**

```
var Chat = new Class.create();

Chat.prototype = {

  initialize: function(room_id, msg_form, msg_input,
  msg_submit, msgs_list_div, users_list_div, user_login,
  options){

    /* ... */

  }

};
```

JS-классы и объекты чата

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Вспомогательные объекты**
  - ChatHelpers
  - ChatAjaxHelpers
  - ChatStrings

```
var ChatStrings = {
  NA: 'N/A',
  TO: ' > '
}
```

Практические приемы

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- Инициализация настроек класса (options)
- Приватные данные в классе
- Борьба с кэшированием AJAX-запросов
- windows-1251 вместо UTF

Tip: инициализация настроек

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
this.options = {
    msgs_url: '/ajax/chat/data.html',
    msg_url: '/ajax/chat/msg.html',
    users_url: '/ajax/chat/users.html',
    frontend_base_url: '/chat/?room_id=0',
    msgs_updater_frequency: 5,
    users_updater_frequency: 20,
    max_msgs_in_list: 100,
    max_chars_in_username: 20
};
Object.extend(this.options, options || {});
```

Tip: приватные данные

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
var _users_open_cache = [];
this.isInUsersOpenCache = function(user_id){
   return (_users_open_cache.indexOf(user_id) >= 0);
}
this.usersOpenCacheAdd = function(user_id){
   if (!this.isInUsersOpenCache(user_id))
   _users_open_cache.push(user_id);
}
this.usersOpenCacheDelete = function(user_id){
   _users_open_cache =
   _users_open_cache.without(user_id);
}
```

Tip: борьба с кэшированием

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
var ChatAjaxHelpers = {
  /* ... */
  addExtraAjaxParams: function(){
    return '&rand=' + Math.random();
  },
  /* ... */
}
```

Tip: windows-1251 вместо UTF

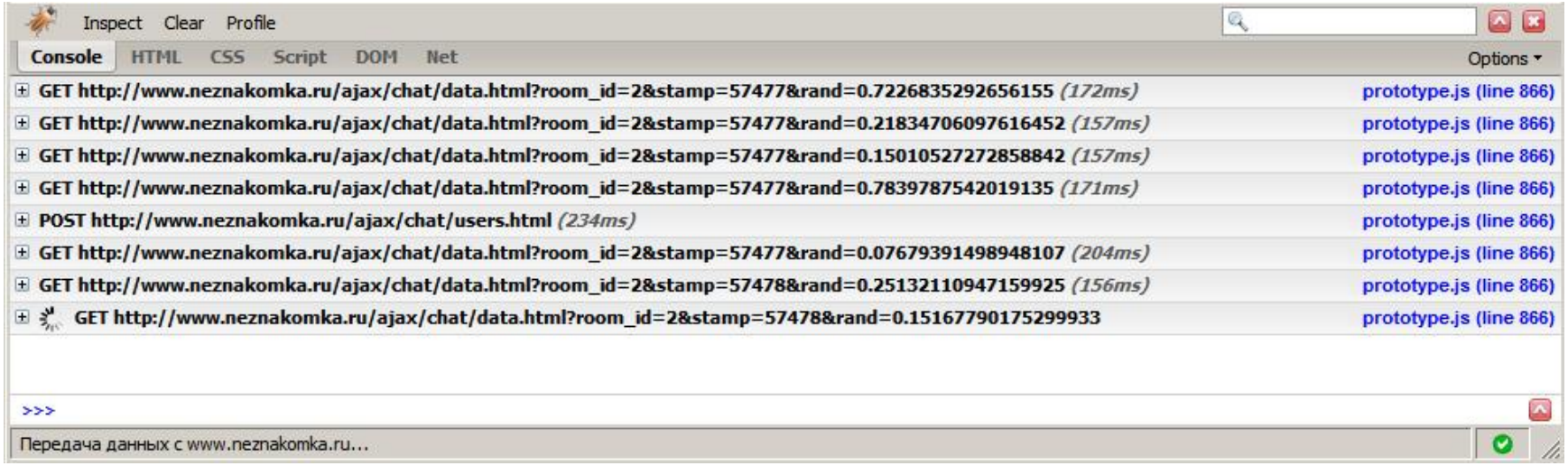
РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
window.encodeURIComponent = function(str){
    var trans = [];
    for (var i = 0x410; i <= 0x44F; i++) trans[i] = i - 0x350; // А-Яа-я
    trans[0x401] = 0xA8; // Ё
    trans[0x451] = 0xB8; // ё
    var ret = [];
    for (var i = 0; i < str.length; i++){
      var n = str.charCodeAt(i);
      if (typeof trans[n] != 'undefined') n = trans[n];
      if (n <= 0xFF) ret.push(n);
    }
    return escape(String.fromCharCode.apply(null, ret)).replace(/\+/g,
    '%2B'); // +
}
window.decodeURIComponent = function(str){
    return unescape(str);
}
```

Отладка чата

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Обработка исключений и AJAX-ошибок**
  - Объект Ajax.Responders
  - Методы ChatAjaxHelpers.handleAjaxFailure и ChatAjaxHelpers.evalJsonResponse
- **Firebug! Firebug! Firebug!**



Ajax.Responders

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
var ajaxGlobalHandlers = {
   onException: function(t, e){
     if (typeof DEBUG != 'undefined' && DEBUG){
          var error_info = $H(e);
          var s = '';
          error_info.each(function(pair){
               s += pair.key + ' = "' + pair.value + '"\n';
          });
          alert(s);
     }
   }
};
Ajax.Responders.register(ajaxGlobalHandlers);
```

ChatAjaxHelpers.handleAjaxFailure

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
var ChatAjaxHelpers = {
  /* ... */
  handleAjaxFailure: function(frontend_base_url,
  request){
    if (typeof DEBUG != 'undefined' && DEBUG)
  alert('Error: ' + request.status + ' -- ' +
  request.statusText);
    else if (request.status == 412) location.reload(); //
  Precondition Failed
    else location.href = frontend_base_url;
  }
}
```

ChatAjaxHelpers.evalJsonResponse

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

```
var ChatAjaxHelpers = {
  /* ... */
  evalJsonResponse: function(json_str){
    var data = null;
    try{
        data = eval('(' + json_str + ')');
    }catch(e){
        if (typeof DEBUG != 'undefined' && DEBUG) throw
  {message: 'JSON data is invalid!'};
    }
    return data;
  }
}
```

Вопросы

Что интересного на сервере?

РОССИЙСКИЕ ИНТЕРНЕТ ТЕХНОЛОГИИ

- **Крутится Perl-демон**
  - Клиент делает запросы по HTTP, через AJAX
  - Демон «узнает» клиента по сессионной куке
  - В ответ на любой из запросов демон отдает текстовый ответ в формате JSON
- **JavaScript-модули gzip’уются**
  - Не стоит особо беспокоиться о большом размере модулей