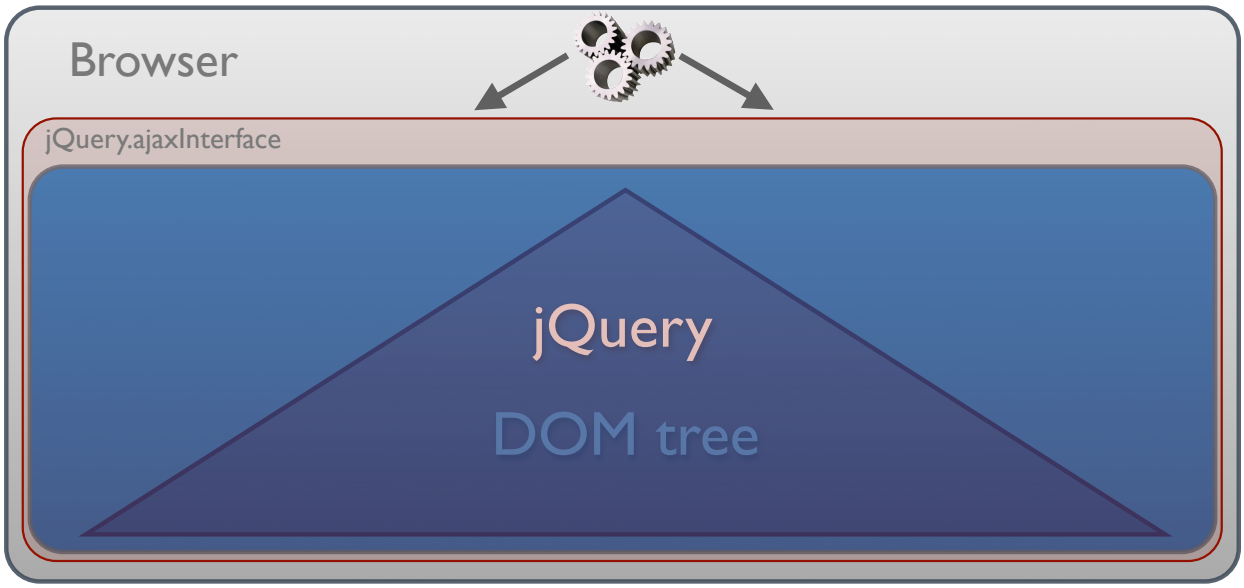
jQuery.ajaxInterface V0.9

"you can cause any frontend modifications without writing a single line of javascript code!"

Idea:
With the help of the jQuery.ajaxInterface you get a standardized way of realizing any frontend adjustments by simply generating an XML document the way you would generate a normal HTML page.

jQuery.ajaxInterface is abstracting the jQuery JavaScript code into an easy readable, editable and first of all generatable XML file.
You can replace any frontend content and meta information with the help of a small XML file (e.g. a template) without writing even one single line of JavaScript code!



Summary:
jQuery.ajaxInterface is a XML wrapper for jQuery.
It's a powerful tool for quickly turning an entire website into an AJAX application and at the same time putting the power of jQuery into the hands of backend developers and template editors.

Roadmap:
The next version is going to add support for jQuery plugins like jQuery UI.

Example:
This example adds new content to the page and changes the page title:

```
<?xml version="1.0"?>
<jquery>

   <title>new page title</title>
   <css>
      <file action="add" src="css/style.css" />
   </css>

   <setSelector value="body">
      <html action="append">
         <![CDATA[
            <h1>new content!</h1>
         ]]>
      </html>
      <setSelector value="body h1">
         <animation action="bind" event="click" name="shake" speed="1500">
            <animation event="click" name="fadeOut" speed="1000" />
         </animation>
       </setSelector>
   </setSelector>

   <response>
     true
   </response>
</jquery>
```

The only thing you need to do in frontend is to add the library and load the links and submit the forms through the jQuery.ajaxInterface, which is like a tiny layer over the original jQuery AJAX functionality and has exactly the same API, including synchronous and asynchronous requests over POST and GET.

```
<script src="jquery.js"></script>
<script src="jquery.ajaxinterface.js"></script>

// Turn all links into ajaxInterface requests
$('a').click(function(){
     $.ajaxInterface({url: this.href});
     return false;
});
```

Why?
So, you still wonder what you need it for?
Compared to a normal HTML file you have a time axis for a number of animations. You can run them one after the other, or bind them to events right through this one file. You can even include other libraries, call JavaScript functions and even start nested ajaxInterface calls. You have the option to add an optional parameter (e.g. "ajax=true") to

every ajaxInterface request. In this case you gain the opportunity to separate the type of the request in backend and deliver either the common HTML file or just the altering part with the ajaxInterface XML document.

Documentation

The biggest difference between a normal AJAX request and an jQuery.alaxInterface request is surely its invisible parsing and execution of the XML commands without any developer involvement. The only result you will get in your frontend application code is the "<response />" part of the XML document.

jQuery.ajaxInterface **request API**:

```
// load via ajaxInterface
jQuery.ajaxInterface({
     type: ["get"]/["post"],
     url: [URL],
     async: [false]/[true],
     data: "request=ajax"
});
```

XML document:

You can place any of the following XML code blocks inside of the "jquery" root node. The "<response/>" block is used to push data to the calling JavaScript function in frontend. All the other blocks will be executed and removed by the ajaxInterface. The result is a new document with the root node called "<response/>" .

Important: jQuery is famous for its method chaining and its use of callback functions. You can use this functionality by simply nesting the blocks into each other. This function is obviously not available for every statement (e.g. title). Just follow this guide since every one of them is marked. On the other hand everyone of the following blocks can be nested into one of those.
A possible use case can be:
- Set the page title.
- Run a number of animations.
- Change the title again once the last animation was completed.

Basic document structure:

```
<?xml version="1.0"?>
<jquery>
 All the ajaxInterface commands should be nested as children of the "jquery" root node.
```

Changing the document title:

```
+<title>
| new page title
+</title>
```

Adding and removing CSS files:

```
+<css>
| CSS meta manipulations (adding and removing css files)
|--+ <file action="" src="" />
|     action: (String) 'add':       Adds a new CSS file to the DOM.
|                      'remove':  Removes a CSS file from the document.
|     src:      (String):               Path to the CSS file
| (in case you want to add GET parameters, the path needs got be URL encoded)
+</css>
```

Adding JavaScript files:

There is no way to remove the executed JavaScript from a document.

```
+<script>
| JavaScript meta manipulations (adding new javascript files)
|--+ <file src="" />
|     src:      (String) :  Path to the JavaScript file (in case you want to add GET
|                           parameters the path needs got be URL encoded)
|--+ <function name="" value='' event=""/>
|     name:   (String):        Name of any global JS function to be called.
|     value:  (JSON) optional: JSON to be used as a parameter for the function call.
|                               Be careful to set the value of this attribute in
|                               single quote, since JSON expects you to user
|                               double quote for its values.
|    event:  (String) optional: Name of the event the function call should be bound
|                               to.
+</script>
```

Adding, replacing, removing and altering DOM elements:

setSelector is the starting point of any kind of body odifications.
You can set an new selector at any point of your XML document.
All the following modifications are going to be performed on the selected DOM elements.
You can either set a new selector by simply adding this tag or by nesting all the following commands inside of it (to increase the readability of the document).

```
+<setSelector value="">
|    selector: (String) All the following modifications will be performed
|                            on the DOM elements selected by this selector (CSS or
XPath).
|--+<html action="">
|  | This tag is used for any HTML modifications.
|  | (if you want to add, replace or remove DOM elements).
|  | action: (String) 'append':  This command will append the following code
|  |                             to the selected elements
|  |                             (Put your code into CDATA inside of the HTML tag).
|  |                  'replace': This command will replace the content of the selected
|  |                              elements.
|  |                  'remove':  This command is removing the selected elements from
|  |                              the DOM.
|  +</html>
|
|--+<id value="" />
|    Setting an new ID for the selected element
|    (make sure that the selector applies for one element only!).
|    value:  (String): new ID for the selected element.
|
|--+<styles action="" value="" />
```

```
|   | This tag is performing inline style modifications on the selected elements.
|   | action: (String) 'add':     This command will add the styles.
|   |                  'replace': This code will replace the existing inline
|   |                             styles with the new ones.
|   |                  'remove':  This code is removing any of the inline style infor
|   |                              mations.
|   | value:  (String): new ID for the selected element.
|  +</styles>
|
|--+<class action="" value="" />
|    action: (String) 'add':     This command adds the listed class name(s) to the DOM
|                                 element.
|                     'replace': This command is replacing any of the existing class
|                                 names with new ones.
|                     'remove':  This command will remove the named classes from the
|                                 elements.
|                                Leaving this tag empty will remove all of the
|                                 existing classes from the element.
|     value:  (String): Class name(s) for the selected elements.
|
|--+<animation event="" action="" name="" speed="" top="" bottom="" left="" right=""
|   |           width="" height="" opacity="" value=''>
|   | This tag will perform custom jQuery animations on the selected elements
|   | and can bind and unbind them on any of the following events.
|   | Nesting is possible: If you want any additional page modifications to be
|   |                      performed after the execution of this animation,
|   |                      you simply have to nest them into this tag
|   |                      (e.g. another animation).
|   | event: (String) (optional): Leaving this attribute empty will cause your
|   |                  animation to be executed without any event being fired.
|   |                  All the events are bound using the jQuery "live" function making
|   |                  sure that they are inherited to any of the new elements matching
|   |                  the given selector.
|   |                  Possible values: blur, focus, focusin, focusout, load, resize,
|   |                  scroll, unload, click, dblclick, mousedown, mouseup, mousemove,
|   |                  mouseover, mouseout, mouseenter, mouseleave, change,
|   |                  select, submit, keydown, keypress, keyup, error
|   | action: (String) (optional) 'bind':  Bind the animation to the named event.
|   |                                 'unbind': Unbind the named event.
|   |
|   | name: (String): This is the name of the animation. Possible animations:
|   |                 delay: This command delays the following animations for the
|   |                         given time.
|   |                        Possible parameters: speed (int: ms)
|   |                 hide: This command is hiding the selected elements.
|   |                        Possible parameters: speed (int: ms)
|   |                 show: This command if showing the selected elements.
|   |                        Possible parameters: speed (int: ms)
|   |                 fadeIn: This command is fading in the selected elements.
|   |                        Possible parameters: speed (int: ms)
|   |                 fadeOut: This command is fading out the selected elements.
|   |                        Possible parameters: speed (int: ms)
|   |                 fadeTo: This command is fading the selected elements to a
|   |                          certain opacity.
|   |                        Possible parameters: speed (int: ms), opacity (float: 0-1)
|   |                 slideUp: This command is hiding the selected elements with
|   |                           a sliding motion (up).
|   |                        Possible parameters: speed (int: ms), opacity (float: 0-1)
|   |                 slideDown: This command is hiding the selected elements with
|   |                             a sliding motion (down).
|   |                        Possible parameters: speed (int: ms), opacity (float: 0-1)
|   |                 moveTo: This command moves an element to a certain position
|   |                          (absolute or relative)
|   |                        Possible parameters: speed (int: ms),
```

```
|   |                                          top  (int: absolute top position,
|   |                                          String: "+=px" relative top position),
|   |                                          bottom  (int: absolute bottom pos,
|   |                                          String: "+=px" relative bottom pos),
|   |                                          left     (int: absolute left position,
|   |                                          String: "+=px" relative left pos),
|   |                                          right    (int: absolute right pos,
|   |                                          String: "+=px" relative right pos)
|   |                  resizeTo: This command resizes an element to a certain size
|   |                              (absolute or relative)
|   |                        Possible parameters: speed (int: ms),
|   |                                          width  (int: absolute width,
|   |                                          String: "+=px" relative width change),
|   |                                          height  (int: absolute height,
|   |                                         String: "+=px" relative height change),
|   |                  shake: Shake is an animation known from jQuery UI.
|   |                         This command will shake an element five times during
|   |                          the given time.
|   |                        Possible parameters: speed (int: ms)
|   |                  setCSS: This command is setting the CSS of the matched
|   |                           elements and runs the next animation.
|   |                          This is not an animations and therefor doesn't have
|   |                           a speed parameter.
|   |                        Possible parameters: value (String: JSON) CSS parameter
|   |                                              to be set.
|   |                                               Be careful to set the value of
|   |                                              this attribute in single quote,
|   |                                              since JSON expects you to user
|   |                                              double quote for its values.
|   |                  animation: This is the generic jQuery animation function.
|   |                        Possible parameters: speed (int: ms)
|   |                                              value (String: JSON) CSS parameter
|   |                                               to be animated during the given time.
|   |                                                    Be careful to set the value of
|   |                                                   this attribute in single quote,
|   |                                                   since JSON expects you to user
|   |                                                   double quote for its values.
|   |
|   +</animation>
|
|--+<ajaxInterface src="" async="" event="" />
|     src  (String): Path to the ajaxInterface XML document
|                         (in case you want to add GET parameters, the path needs
|                          got be URL encoded)
|     async (String) true: The request will be performed asynchronously.
|                    false: The request will be performed synchronously.
|     event (String) (optional): Leaving this attribute empty will cause your
|                                 animation to be executed without any event being fired.
|                                 All the events are bound using the jQuery "live"
|                                 function making sure that they are inherited to any of
|                                 the new elements matching the given selector.
|                     Possible values: blur, focus, focusin, focusout, load, resize,
|                     scroll, unload, click, dblclick, mousedown, mouseup, mousemove,
|                     mouseover, mouseout, mouseenter, mouseleave, change,
|                     select, submit, keydown, keypress, keyup, error
+</setSelector>
```

EXAMPLES:

```
<?xml version="1.0"?>
<jquery>
   ALL THE EXAMPLES GO INHERE!
      <response /> - This is the response part and will be forwarded into the calling function. You can use it to pass the backend status to your program. The inner part is being forwarded as a standalone xml document.
</jquery>
```

Examples HEADER:

- Replacing the page title:

jQuery:

```
$("title").text("new page title");
```

jQuery.ajaxInterface:

```
<title>parsing successful</title>
```

- Adding a CSS file to the DOM tree:

jQuery:

```
$("head").append('<link rel="stylesheet" href="path/to/cssfile.css" />');
```

jQuery.ajaxInterface:

```
Be careful to set the value of this attribute in single quote,
since JSON expects you to user double quote for its values.
<css>
  <file action="add" src="path/to/cssfile.css" />
</css>
```

- Removing a CSS file from the DOM tree:

jQuery:

```
$("head link[href='path/to/cssfile.css']").remove();
```

jQuery.ajaxInterface:

```
Be careful to set the value of this attribute in single quote,
since JSON expects you to user double quote for its values.
<css>
  <file action="remove" src="path/to/cssfile.css"/>
</css>
```

- Adding a JavaScript file to the DOM tree:

jQuery:

```
$("body").append('<script type="text/javascript"
src="path/to/javaScriptFile.js"></script>');
```

jQuery.ajaxInterface:

```
Be careful to set the value of this attribute in single quote,
since JSON expects you to user double quote for its values.
<script>
  <file  src="path/to/cssfile.css"/>
</script>
```

- Removing a JavaScript file from the DOM tree:
There is no way to remove the executed JavaScript from a document.

- Calling a global JavaScript function with parameters:
jQuery:
```
$("body").click(function(){
     exampleFunction({"param": "true"})
});
```

jQuery.ajaxInterface:
```
<script>
  <function name="exampleFunction" value='{"param": "true"}' event="click"/>
</script>
```

Examples DOM:

- Setting a selector:
jQuery.ajaxInterface:
```
<setSelector value="body"/>
```
This will set a new selector for all of the following modifications until another selector is set.

- Adding a new element to the DOM tree:
jQuery:
```
$("body").append("<h1>new content!</h1>");
```

jQuery.ajaxInterface:
```
<setSelector value="body">
  <html action="append">
    <![CDATA[
      <h1>new content!</h1>
    ]]>
  </html>
</setSelector>
```

- Removing an element from the DOM tree:
jQuery:
```
$("h1").remove();
```

jQuery.ajaxInterface:
```
<setSelector value="h1" />
<html action="remove" />
```

- Replacing an element in the DOM tree:
jQuery:
```
$("h1").html("another content");
```

jQuery.ajaxInterface:
```
<setSelector value="h1" />
<html action="append">
  <![CDATA[
    another content
  ]]>
</html>
```

- Adding or replacing an ID of a DOM element:

jQuery:

```
$("h1").attr("id","ID1");
```

jQuery.ajaxInterface:

```
<setSelector value="h1" />
<id value="ID1" />
```

- Adding a CSS class to a DOM element:

jQuery:

```
$("h1").addClass("newClass");
```

jQuery.ajaxInterface:

```
<setSelector value="h1" />
<class action="add" value="newClass" />
```

- Removing a CSS class from a DOM element:

jQuery:

```
$("h1").removeClass("newClass");
```

jQuery.ajaxInterface:

```
<setSelector value="h1" />
<class action="remove" value="newClass" />
```

- Replacing a CSS class on a DOM element:

jQuery:

```
$("h1").removeClass().addClass("newClass");
```

jQuery.ajaxInterface:

```
<setSelector value="h1" />
<class action="replace" value="newClass" />
```

- Adding CSS styles to a DOM element:

jQuery:

```
$("h1").css({border: "1px solid red", width: "100px", color: "red"});
```

jQuery.ajaxInterface:

```
<setSelector value="h1" />
<styles action="add" value='{"border": "1px solid blue", "width": "100px", "color":
"red"}' />
```

- Replacing CSS styles on a DOM element:

jQuery:

```
$("h1").removeAttr("style").css({border: "1px solid red", width: "100px", color:
"red"});
```

jQuery.ajaxInterface:

```
<setSelector value="h1" />
<styles action="replace" value='{"border": "1px solid blue", "width": "100px",
"color": "red"}' />
```

- Removing CSS styles from a DOM element:
jQuery:
```
$("h1").removeAttr("style");
```

jQuery.ajaxInterface:
```
<setSelector value="h1" />
<styles action="remove" />
```

Examples ANIMATIONS:
jQuery.ajaxInterface is also bringing an abstraction for jQuery events and animations.
There are two possible types of animations: the first ones are running immediately on the element, the second ones have to be fired by an event.
A possible XML document for running two animations after each other after a click event has been fired could look something like this:

```
<?xml version="1.0" ?>
<jquery>
  <setSelector value="h1">
     <animation name="click" name="delay" speed="750">
        <animation name="moveTo" top="+=200" left="+=200" speed="1500">
          <animation name="fadeOut" speed="1500" />
        </animation>
     </animation>
  </setSelector>
</jquery>
```

All the nested animations will be performed on the selected elements and have therefore to be placed after or inside of a `<setSelector>` tag.
There are two ways to place a multiple number of animations. Placing them one after the other would be equal to a normal method chain, however placing them inside of each other would execute them in a call-back manner.

- Animating an element with method chaining:
jQuery:
```
$("h1").animate({top: "+=200", left: "+=200"}, 1500).animate({top: "+=200", left:
"+=200"}, 1500);
```

jQuery.ajaxInterface:
```
<setSelector value="h1">
  <animation name="moveTo" top="+=200" left="+=200" speed="1500" />
  <animation name="resizeTo" width="300" height="300" speed="1500" />
</setSelector>
```

- Animating an element with callbacks:
jQuery:
```
$("h1").animate({top: "+=200", left: "+=200"}, 1500, function(){
  $(this).animate({width: 300, height: 300}, 1500);
});
```

jQuery.ajaxInterface:
```
<setSelector value="h1">
  <animation name="moveTo" top="+=200" left="+=200" speed="1500">
```

```
    <animation name="resizeTo" width="300" height="300" speed="1500" />
  </animation>
</setSelector>
```

Now we want to run the animation after an event was fired.

- Animating an element with click event:
jQuery:
```
$("h1").click(function(){
  $(this).animate({top: "+=200", left: "+=200"}, 1500, function(){
    $(this).animate({width: 300, height: 300}, 1500);
  });
});
```

jQuery.ajaxInterface:
The only thing we need to add is the event parameter.
```
<setSelector value="h1">
  <animation event="click" name="moveTo" top="+=200" left="+=200" speed="1500">
    <animation name="resizeTo" width="300" height="300" speed="1500" />
  </animation>
</setSelector>
```

- Animating an element with multiple events:
We just have to add an other animations block with an other event.
```
<setSelector value="h1">
  <animation event="mouseover" name="moveTo" top="+=200" left="+=200" speed="1500">
    <animation name="resizeTo" width="300" height="300" speed="1500" />
  </animation>
  <animation event="click" name="fadeOut" speed="1500" />
</setSelector>
```

Possible events:
blur, focus, focusin, focusout, load, resize, scroll,
unload, click, dblclick, mousedown, mouseup, mousemove,
mouseover, mouseout, mouseenter, mouseleave, change,
select, submit, keydown, keypress, keyup, error

Possible animations:
hide, show, fadeIn, fadeOut, fadeTo, moveTo, resizeTo,
shake
Shake is an animation known from jQuery UI. It shakes your element five times during the given time.

An example jQuery.ajaxInterface XML document might look something like this:

```
<?xml version="1.0"?>
<jquery>

     <title>new page title</title>
     <css>
           <file action="add" src="css/style.css" />
     </css>
    <setSelector value="body">
          <html action="append">
               <![CDATA[
                    <div id="newBlock">
                         <h1>new content!</h1>
                    </div>
               ]]>
          </html>
     <setSelector value="#newBlock">
          <animation action="bind" event="click" name="shake" speed="1500">
               <animation action="unbind" event="click" name="shake" />
               <animation name="setCSS" value='{"border": "10px solid green"}'>
                    <title>new page title after animation</title>
                         <css>
                              <file action="add" src="css/otherStyle.css" />
                         </css>
                         <html action="append">
                              <![CDATA[
                                   <h2>newer content!</h2>
                         ]]>
                         </html>
                         <setSelector value="#newBlock h1" />
                        <id value="idForH1" />
                        <styles action="add" value="color: red;" />
                        <script>
                              <file src="example.js" />
                              <function name="exampleFunction" value='{"param":
"true"}' event="click"/>
                        </script>
                        <ajaxInterface src="test2.xml" async="true" event="click">
                         <setSelector value="body">
                                   <html action="append">
                                        <![CDATA[
                                             <h3>ajax ready!</h3>
                                        ]]>
                                   </html>
                              </setSelector>
                        </ajaxInterface>
                   <setSelector value="h2" />
                   <animation event="click" name="fadeOut" speed="1000" />
                    </animation>
               </animation>
          </setSelector>
     </setSelector>

     <response>
     true
     </response>
</jquery>
```