Using Ajax with X-Objects

This brief tutorial will show you some best practices for using Ajax URLs with X-Objects and how to simplify the implementation process.

Step 1: Creating your Ajax Controller

First we'll create a new Controller called **a_controller** in the file **/app/controllers/a.php** and extend our standard controller class.

For this example, in our Controller, we have 2 new example methods, one echoes back JSON to the client, and the other echoes back xHTML:

```
class a_controller extends xo_controller {
	// add a widget, and echo back the results as JSON
	public function add(){
		header('Content-Type: application/json');
		// use controller's magic method $this->req to access $_REQUEST variable
		echo (string) new add_widget_component(json_decode($this->req->json));
	}
	// display a widget, given its Id
	public function display(){
		header('Content-Type: text/html; charset=utf-8');
		/** use controller's method $this->uri() to get a piece of the URI
		 *  In this case, since the url is /a/display/<id> we use the third component
		 */
		echo (string) new widget_display_component($this->uri->part(3));
	}
}
```

Step 2: Create Web Components

Now we'll need two new web components, **add_widget_component** and **widget_display_component** to do the work for us.

```
/** use of xo_json_object gives us an easy way to collate results in an array and
 *   display back as JSON
 */
class add_widget_component extends xo_json_object {
	public function __construct($json){
		$widget = new widget();
		$widget->set_from_json($json);	// inherited from business_object
		$this->json_result['result'] = $widget->save()?'success':'error';
		$this->json_result['error'] = $widget->save_error;
	}
}
```

```
/** xo_displayable_object gives us a quick way to bind a view to an object */
class widget_display_component extends xo_displayable_object {
        public function __construct($widget_id){
                $this->object = new widget("id='$widget_id'");
                $this->view = 'my-widget-view';
        }
}
```

Step 3: Create View for Display

Now in /app/views we need to add a new view for the above display component, as 'my-widget-view.php':

```
<div class="my-widget-view id-<?=$this->business_object->id?>">
        <span><?=$this->business_object->name?></span>
</div>
```

Step 4: Add client Ajax calls

Now somewhere in your JavaScript or jQuery code, you'll want to call each method:

```
// call to add widget:
$.ajax({
        type: 'POST',
        url: '/a/add',
        data: 'json='+JSON.stringify({
                name: $('#name').val()
        }),
        success: function(json){
                if (json.result =='success'){
                        // do something
                } else {
                        // do something else
                }
        }
});

// call to display widget
$.ajax({
        url: '/a/display/'+$('#widget_id').val(),
        success: function(xhtml){
                $('.receptacle').html(xhtml);
        }
});
```