

## Die Tutorialschritte:

1. Installation des Ext JS Frameworks
2. Einführung der nötigen API-Elemente
3. Implementierung eines Beispiels

## 1. Installation

Datei *tutorial.zip* von <http://www.me73.com/downloads/tutorial.zip> herunterladen und den Inhalt in ein beliebiges Verzeichnis entpacken.

In dem *ext* Ordner befinden sich die notwendigen Dateien des Frameworks, dieser Ordner muss später in den Beispielen referenziert werden.

## 2. Die ExtJS-API

Für eine detailliertere Referenz bitte <http://extjs.com/deploy/dev/docs/> anschauen.

### **Ext**

Ein globales (Singleton-) Objekt des Typs Ext. Das Objekt hält die Hauptfunktionalität des Frameworks.

Methoden von Bedeutung sind:

*void onReady(Function fn, Object scope, boolean override):*

Dieser Event-Handler wird ausgelöst sobald das Dokument fertig geladen wurde.

*Element get(Mixed el):* Erstellt / gibt das gefragte Element zurück.

### **Ext.Window**

Eine Fensterklasse für Fensterinstanzen.

Für das Tutorial relevante Methoden sind:

**Window(Object config):** Konstruktor mit Konfiguration als Parameter.

**show(Element target):** Zeigt das Fenster. Aktiviert und bringt das Fenster in den Vordergrund, falls es inaktiv ist.

**hide():** Versteckt und deaktiviert ein Fenster.

### **Ext.TabPanel**

Ein Panel, welches mehrere Tabs halten kann.

## 1. Implementierung

Im Rahmen dieses Tutorials werden wir ein einfaches Fenster mit Tabs mit Hilfe des Frameworks erstellen.

Als erstes stellt man nochmal sicher wo sich der vorher entpackte *ext* Ordner befindet. Nun fangen wir mit der HTML-Datei an.

---

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Hello World Window</title>
    <link rel="stylesheet" type="text/css" href="ext/resources/css/ext-all.css" />
    <script type="text/javascript" src="ext/ext.js"></script>
    <script type="text/javascript" src="helloworld.js"></script>
  </head>
```

---

Der Header bietet noch keine Überraschungen, man muss jedoch sicherstellen, dass der **ext** Ordner hier richtig verlinkt wird. In diesem Fall liegt die HTML Datei im selben Verzeichnis wie der **ext** Ordner. Das angeben der *ext-all.css* ist notwendig, da diese auch für die Funktionalität der Fenster und Tabs mitverantwortlich ist. **ext.js** ist das eigentlich Framework. **helloworld.js** ist die Skriptdatei, die wir als nächstes schreiben werden.

---

```
  <body>
    <h1>Hello World Window!</h1><br />
    <input type="button" id="windowButton" value="Hello Window" /><br /><br />
    <div id="window" class="x-hidden">
      <div class="x-window-header">Hello World</div>
      <div id="tabs">
        <div class="x-tab" title="Hello World 1">
          <p>Hello</p>
        </div>
        <div class="x-tab" title="Hello World 2">
          <p>World!</p>
        </div>
      </div>
    </div>
  </body>
</html>
```

---

Der Body-Teil erzeugt einen Button, dessen id später vom Framework gebraucht wird, in diesem Fall **windowButton** genannt. Das **window** div ist notwendig für das erstellen des Fensters, dies ist der Container dafür. Hier werden auch gleich die beiden Tabs bereits erstellt, die ebenfalls in divs gehalten werden.

Wir haben den HTML Rahmen nun fertig, jetzt fehlt noch das Javascript. Dazu erstellen wir eine neue Datei, in meinem Fall habe ich sie **helloworld.js** genannt. In dieser erstellen wir als erstes:

---

```
var win;

function closeWindow()
{
    win.hide();
}
```

Die globale Variable **win** wird später das Fensterobjekt halten. Die Funktion **closeWindow** versteckt das Fenster.

---

```
function createWindow()
{
    if (!win)
    {
        var tabPanel = new Ext.TabPanel({
            el: 'tabs',
            autoTabs:true,
            activeTab:0,
            deferredRender:false,
            border:false
        });
        var submitButton = {text:'Submit', disabled:true};
        var closeButton = {text: 'Close', handler: closeWindow};
        win = new Ext.Window({
            el:'window',
            layout:'fit',
            width:500,
            height:300,
            closeAction:'hide',
            plain: true,
            items: tabPanel,
            buttons: [submitButton, closeButton]
        });
    }
    win.show(this);
}
```

---

Die Funktion **createWindow** erstellt bzw. aktiviert ein Fenster. Wenn das Fenster noch nicht erstellt worden ist, werden ein **TabPanel** und zwei **Buttons** angelegt. Diese und weitere Eigenschaften des Fensters werden dann in den Konstruktor von **Window** übergeben. Als Event Handler für das **closeButton** Objekt wird die vorher definierte **closeWindow** Funktion angegeben.

---

```
function main()
{
    var button = Ext.get('windowButton');
    button.on('click', createWindow);
}

Ext.onReady(main);
```

---

Schließlich wird noch die **main** Funktion definiert, die als Event Handler für das Laden des Dokuments bereitsteht. Der vorher in HTML definierte Button **windowButton** wird hier als **Element** Objekt instanziiert und dessen **click**-Event mit Funktion **createWindow** abgearbeitet.