jQuery UI Widgets

Basics, Application, Pros&Cons

Nils Glück

Schedule (15-20 min.)

- When it makes sense to use jQuery UI widgets
- How you do it + what it does for you
- Simple example
- Pros & cons

When it makes sense

The customer says:

„We have this very complex, very important environment (CMS, framework) running. **Keep it running.**“

„We just need some client-sided solution **as an add-on**“

„We want a **flexible yet stable** solution. Please keep it simple, and be sure to document it well“

Implications: Stand-alone + conventional + API

✌ jQuery + DOM node = scope ✌

- In the early days, the DOM was simple hypertext:

```
<b id="myelement">so boring</b>
```

- Nowadays, DOM nodes have their own jQuery scope:

```
<b id="myelement" data-helloworld="example ;)">spectacular</b>
```

- jQuery can deal with that:

```
> $('#myelement').data()
<· Object {helloworld: "example ;)"}
```

- jQuery UI works **literally on DOM nodes**

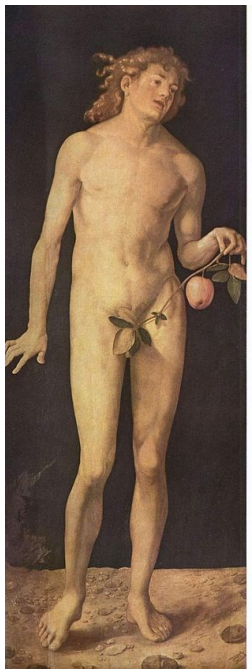
The basic idea

- Let‘s create something totally new, a **plugin of its own**
- That thing should work for **many scenarios** (it should be able to have **many instances**)

- Let‘s play God in jQuery.

```
> $("ul.planets li#earth #el1").human(
{ id: "Adam", sex: "male" }
);

> $("ul.planets li#earth #el2").human(
{ id: "Eve", sex: "female" }
);
```



The widget prototype

- Let‘s create our plugin prototype (= widget prototype).
- The manual says: „All of the functionality that automatically gets added to your plugin comes from **a base widget prototype**, which is defined as jQuery.Widget.prototype.“
- Your plugin will be **based** on this prototype
- Your plugin will show up in the DOM as **instances based on your plugin**

Object concept

jQuery

As a function:
Base Widget prototype
(„what all creations need“)

CODE

An instance of this:
Prototype of your widget
(„what your particular creation is“)

OPTIONS

An instance of this:
Your widget instances in a DOM
(„what your creatures are“)

Let‘s have a party

Before we can „have“ it, we must conceptualize it in a **widget prototype**:

```
$.widget("webworker.party",
{ /* ... */ }
);
```

- What happens at a party?
- What does a party do?
- What objects does a party deal with?
- What options should a party have?

Let‘s party in many instances

- Partys are all different, though they have one common prototype
- Music can be different
- People can be different
- Some partys are existing but never really start

Simple prototype, complex instances

- Why not load options inside an external JSON source?
- http://www.dw.de/flashes/html5/mit17/index.html?http://www.dw.de/flashes/html5/mit17/content/data.json&de
- Change one option -> make it Russian: http://www.dw.de/flashes/html5/mit17/index.html?http://www.dw.de/flashes/html5/mit17/content/data.json&ru
- Change another -> Make it English: http://www.dw.de/flashes/html5/mit17/index.html?http://www.dw.de/flashes/html5/mit17/content/data.json&en

PR☺S & C☹NS

- Simple: Prototype + options = instance
- It teaches you abstraction and soft code
- 100% OOP + 100% API + 100% jQuery
- Super-fast development ($$ Ka-Ching $$)

- Overhead
- Not really for OOP beginners
- Boring

Resources

- jQuery CDN: https://code.jquery.com/
- jQuery UI custom CDN (UI+Core required): http://jqueryui.com/download/
- The widget manual: http://learn.jquery.com/jquery-ui/widget-factory/
- The widget prototype API: http://api.jqueryui.com/jquery.widget/

About me

- Freelance web developer
  (JavaScript + JavaScript frameworks as well as front-end in general; also some PHP+DB stuff)
- @home: Cologne, Germany
- Keen on **statistics**, **data modeling**, **visualization**
- M.Sc. Market Research
- https://www.xing.com/profile/Nils_Glueck2
- https://github.com/NilsOle
- <info@nilsole.net>