Obviel Documentation

Release 1.0

Obviel Developers

May 19, 2013

CONTENTS

Obviel is a client-side web framework. Obviel helps you build dynamic, rich client-side web applications in JavaScript.

The core principle of Obviel: **For a JavaScript *object* you render a *view* on an *element*: Ob-vi-el**

Obviel's core is a simple extension to jQuery. It adds a `render` function that is used to render objects on DOM elements:

```
$(".foo").render(obj);
```

`obj` can be any object with a simple string property called `iface`:

```
var obj = {
  iface: 'animal'
  name: 'Elephant'
};
```

You tell Obviel how to render an object of a certain iface by declaring a *view*:

```
obviel.view({
  iface: 'animal',
  obvt: '<div>Animal name: {name}</div>'
});
```

You can use plain JavaScript code, templates, or a combination to construct a view.

That's Obviel in a nutshell. Obviel's `manual` tells you a lot more about Obviel's core.

On top of that Obviel adds a lot of features, such as templating, i18n support, form generation and validation, and routing. Obviel stays close to HTML but lets you build sophisticated components when you need to.

CHAPTER

ONE

CLIENT-SIDE TEMPLATING

Obviel has a `client-side templating language` built-in. This template language has easy-to-use but extensive `i18n support` so you can offer your web application in multiple languages. It also integrates well with Obviel core, supporting sub-view rendering and easy event handlers.

CHAPTER

TWO

INTERNATIONALIZATION SUPPORT (I18N)

Obviel lets you create a web UI that supports more than one language. It uses `a standard gettext-based approach for i18n`. Obviel comes with a server-side toolchain that can be used to extract translatable text from both JavaScript code as well as templates, and to prepare translations for publication on the web.

CHAPTER

THREE

FORM LIBRARY

Obviel comes with a client-side `form library`. Describe your form using JSON structures, either on the client or on the server. Get and post your form data as JSON. Use composite and repeating fields with ease!

CHAPTER

FOUR

ROUTING WITH TRAJECT

Obviel contains an easy to use `routing library` called Traject, which can be used both to resolve paths to objects as well as construct paths for objects. With Traject you can construct dynamic single-page user interfaces that work with hyperlinks and the browser back button.

CHAPTER

FIVE

VERSATILE BUT UNOBTRUSIVE

Obviel is powerful. Obviel is also *unobtrusive*. You only need to learn a small, powerful core API to unlock the power of Obviel. Obviel puts minimal demands on your JavaScript objects. They can be plain JavaScript objects on the client. The objects could also be coming from a web server.

CHAPTER

SIX

RESTFUL

Obviel doesn't just say REST is cool and then go through the motions. An Obviel-based app can start with a single URL and find out about the rest of your application's URLs by following *hyperlinks* in objects. You can also swap between hyperlink and sub-object transparently. And that's RESTful.

CHAPTER

SEVEN

LOOSE COUPLING

Obviel lets you build dynamic, *loosely coupled* web applications. Your JavaScript code will be partitioned into small, loosely coupled cooperating components. Obviel also supports loose coupling between the web server and the client application. You can change your server code and your client UI will adjust itself without changing a line of JavaScript. You can also modify JavaScript code without worrying too much about the server.

CHAPTER

EIGHT

LEARNING MORE

Look at `demo code`.

Read the `Obviel documentation`.

Check out the `Obviel API`.

Questions? Comments? Join the `Obviel community`!

Want to contribute to Obviel? Read our `notes on development`.

CHAPTER

NINE

DOWNLOAD OBVIEL

Download Obviel. See the `changelog` for what's new!