Data Binding

Data-binding — позволяет реализовать постоянную синхронизацию JS моделей с представлением, избежать массового дублирования кода, отвечающего за его обновления.

Data-binding – приобретает особую значимость когда идет речь о связи model-view.

[***1: Object.observe && DOM.onChange***](http://frontender.info/native_javascript_data_binding/#1objectobserveampampdomonchange)

Object.observe() – часть стандарта ECMAScript 7, используется для асинхронного обзора изменений джаваскрипт объектов, позволяет observer-у получать упорядоченную по времени последовательность записей, описывающих набор изменений, случившихся с наблюдаемым объектом.

События, о которых observer будет оповещен:

- добавление нового свойства;

- обновление значение существующего свойства;

- удаление свойства;

- реконфигурация свойства (изменение свойств, которые могут быть указаны в Object.defineProperty :

enumerable: false,

configurable: false,

writable: false,

value: 'static'

)

Для того чтоб добавить observer на объект:

Object.observe(todoModel, observer); // observer – коллбек-функция, которая вызовется, когда наблюдаемый объект как-то изменится.

function observer(changes){

changes.forEach(function(change, i){

console.log('what property changed? ' + change.name);

console.log('how did it change? ' + change.type);

console.log('whats the current value? ' + change.object[change.name]);

console.log(change); // all changes

});

}

Можно подписаться только на определенный вид события:

Object.observe(todoModel, observer, ['delete']);

Чтоб отписаться:

Object.unobserve(todoModel, observer);

***2 Обрабатывать событие change***

// <input id="foo">

$("#foo").val("");

function bindDomElemToObjProp(domElem, obj, propertyName) {

$(domElem).change(function() {

obj[propertyName] = $(domElem).val();

alert("user.name теперь "+user.name);

});

}

user = {}

bindDomElemToObjProp($("#foo"), user, 'name');

// Введите в поле ввода 'Обама'

user.name // Обама.

## Недостаток - использование *.change*не сработает для изменений, которые не вызывают событие *change*, например, программное изменение DOM.

## *3 Изменение get и set c помощью Object.defineProperty*

//<input id="foo">

Object.defineProperty(user, 'name', {

get: function() { return document.getElementById("foo").value },

set: function(newValue) { document.getElementById("foo").value = newValue; },

configurable: true // Для того, чтобы можно было переопределить это позднее

});

Или более универсально:

function bindModelInput(obj, property, domElem) {

Object.defineProperty(obj, property, {

get: function() { return domElem.value; },

set: function(newValue) { domElem.value = newValue;},

configurable: true

});

}

Links

<http://frontender.info/native_javascript_data_binding/>

<http://www.html5rocks.com/en/tutorials/es7/observe/>

<http://engineering.paiza.io/entry/2015/03/12/145216>