SAFE$_{Wapp}$: 자바스크립트 웹 앱을 위한 Web API 오용 검출기

배성경[1] 조형헌[2] 류석영[1]

[1]**카이스트** [2]삼성전자

“JavaScript”

“동적인 언어”

```
function f(x) {
    return x;
}
```

```
function f(x) {
        return x;
    }
f(42);
```

```
function f(x) {
        return x;
    }
f(42);

f("문자열");
```

```
function f(x) {
    return x;
}
f(42);
f({ 임의의: "객체" });
f("문자열");
```

```
function f(x) {
    return x;
}

f(42);
f(f);
f({ 임의의: "객체" });
f("문자열");
```

```
* Bug Detector *
example.js:11:1~11:22: [Warning] Calling a function 'f' with the
parameter 'x' of varying types (Number, String, Object).
```

```
function f(x) {
    return x;
}

f(42);
f(f);
f({ 임의의: "객체" });
f("문자열");
```

```
function f(x) {
    return x;
}
```

```
function f(x) {
    return x;
}

f();
```

```
function f(x) {
    return x;
}

f();

    f("인자", "개수", "상관없음");
```

```
* Bug Detector *
example.js:10:1~10:4: [Warning] Too few arguments to function 'f'.
example.js:11:1~11:22: [Warning] Too many arguments to function 'f'.
```

```
function f(x) {
    return x;
}

f();

    f("인자", "개수", "상관없음");
```

하지만 API 함수를 쓴다면
어떨까?

.add()

Categories: Traversing > Miscellaneous Traversing

.add(selector)

Returns: jQuery

Description: *Add elements to the set of matched elements.*

.add(selector)

version added: 1.0

selector
Type: Selector
A string representing a selector expression to find additional elements to add to the set of matched elements.

.add(elements)

version added: 1.0

elements
Type: Elements
One or more elements to add to the set of matched elements.

.add(html)

version added: 1.0

html
Type: htmlString
An HTML fragment to add to the set of matched elements.

NAMESPACE

Prototype

View source on GitHub →

Description

The `Prototype` namespace provides fundamental information about the Prototype library you're using, as well as a central repository for default iterators or functions.

We say "namespace," because the `Prototype` object is not intended for instantiation, nor for mixing in other objects. It's really just... a namespace.

Your version of Prototype

Your scripts can check against a particular version of Prototype by examining `Prototype.Version`, which is a version `String` (e.g. "<%= PROTOTYPE_VERSION %>"). The famous script.aculo.us library does this at load time to ensure it's being used with a reasonably recent version of Prototype, for instance.

Browser features

Prototype also provides a (nascent) repository of browser feature information, which it then uses here and there in its source code. The idea is, first, to make Prototype's source code more readable; and second, to centralize whatever scripting trickery might be necessary to detect the browser feature, in order to ease maintenance.

Default iterators and functions

Numerous methods in Prototype objects (most notably the `Enumerable` module) let the user pass in a custom iterator, but make it optional by defaulting to an "identity function" (an iterator that just returns its argument, untouched). This is the `Prototype.K` function, which you'll see referred to in many places.

Many methods also take it easy by protecting themselves against missing methods here and there, reverting to empty functions when a supposedly available method is missing. Such a function simply ignores its potential arguments, and does nothing whatsoever (which is, oddly enough, blazing fast). The quintessential empty function sits, unsurprisingly, at `Prototype.emptyFunction` (note the lowercase first letter).

Namespaces {}

Prototype.Selector

A namespace that acts as a wrapper around the choosen selector engine (Sizzle by default).

Prototype.Browser

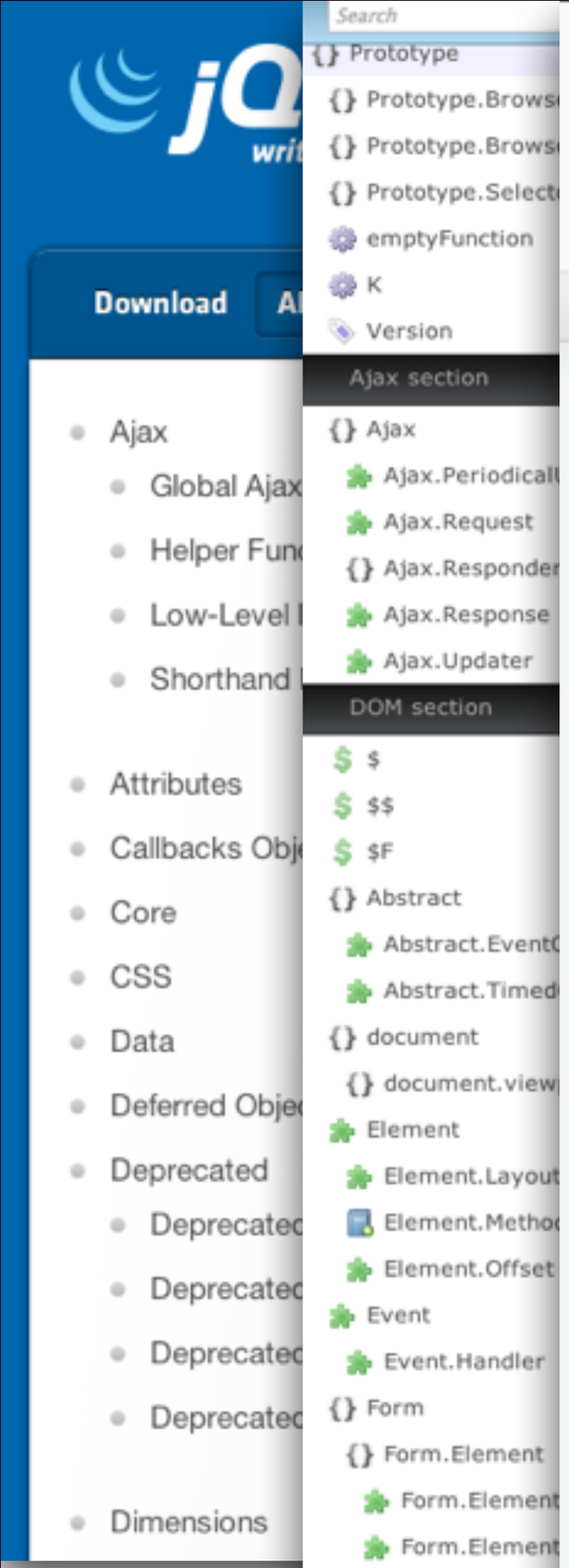
A collection of Boolean values indicating the browser which is currently in use. Available properties are `IE`, `Opera`, `WebKit`, `MobileSafari` and `Gecko`.

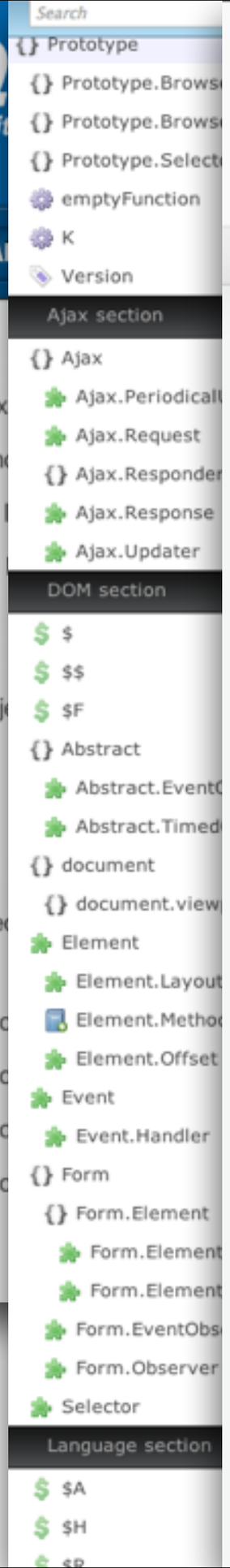
Prototype.BrowserFeatures

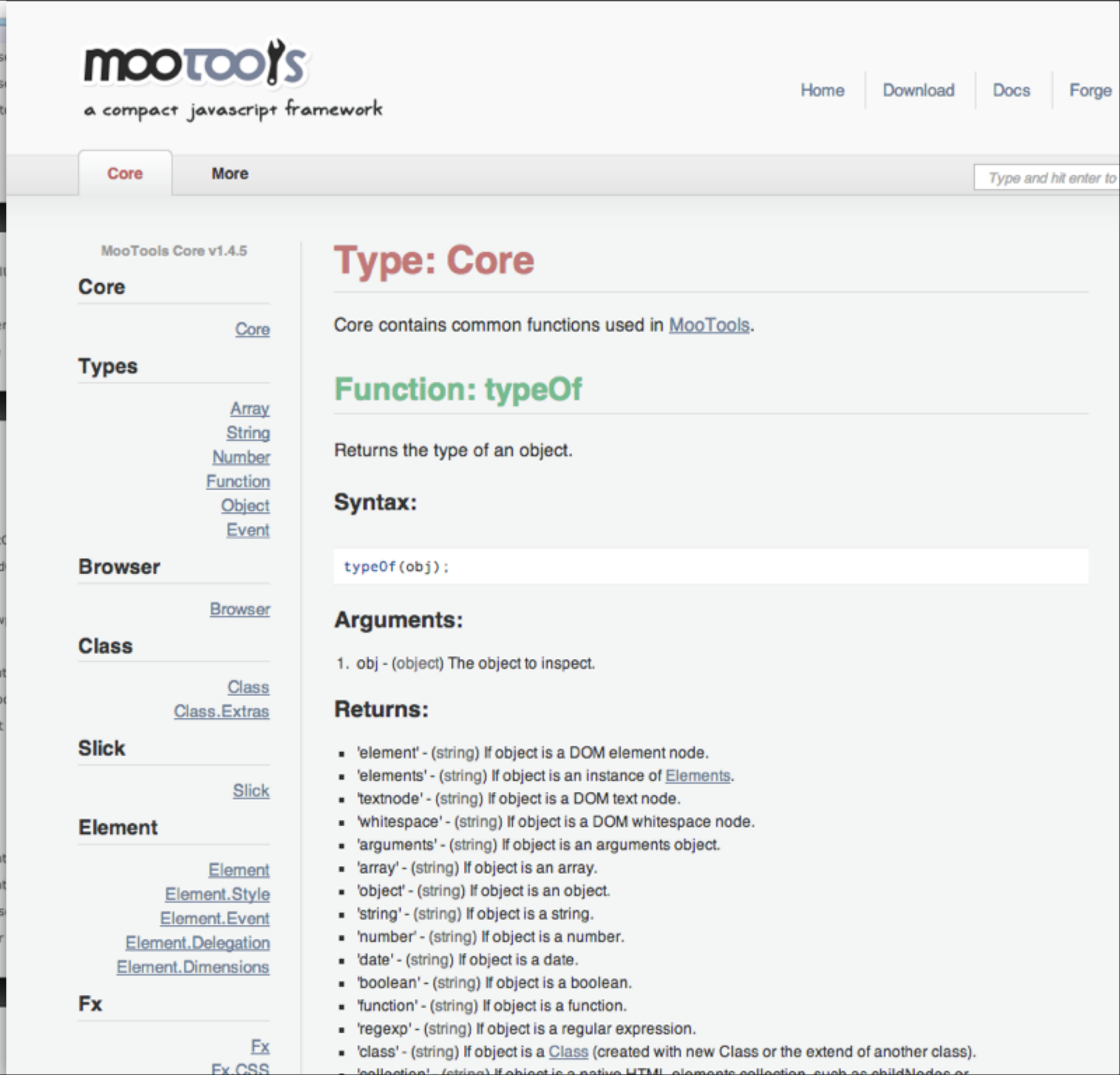
A collection of Boolean values indicating the presence of specific browser features.

Class methods `emptyFunction` `K`

Class properties `Version`


Download
Ajax
Global Ajax
Helper Func
Low-Level
Shorthand
Attributes
Callbacks Obj
Core
CSS
Data
Deferred Obje
Deprecated
Deprecated
Deprecated
Deprecated
Deprecated
Dimensions


Search
Prototype
Prototype.Brows
Prototype.Brows
Prototype.Select
emptyFunction
K
Version
Ajax section
Ajax
Ajax.Periodicall
Ajax.Request
Ajax.Responder
Ajax.Response
Ajax.Updater
DOM section
$
$$
$F
Abstract
Abstract.Event
Abstract.Timed
document
document.view
Element
Element.Layout
Element.Metho
Element.Offset
Event
Event.Handler
Form
Form.Element
Form.Element
Form.Element
Form.EventObs
Form.Observer
Selector
Language section
$A
$H
$R


mooTools
a compact javascript framework
Home
Download
Docs
Forge
Core
More
Type and hit enter to
MooTools Core v1.4.5
Core
Core
Types
Array
String
Number
Function
Object
Event
Browser
Browser
Class
Class
Class.Extras
Slick
Slick
Element
Element
Element.Style
Element.Event
Element.Delegation
Element.Dimensions
Fx
Fx
Fx.CSS
Type: Core
Core contains common functions used in MooTools.
Function: typeOf
Returns the type of an object.
Syntax:
typeOf(obj);
Arguments:
1. obj - (object) The object to inspect.
Returns:
'element' - (string) If object is a DOM element node.
'elements' - (string) If object is an instance of Elements.
'textnode' - (string) If object is a DOM text node.
'whitespace' - (string) If object is a DOM whitespace node.
'arguments' - (string) If object is an arguments object.
'array' - (string) If object is an array.
'object' - (string) If object is an object.
'string' - (string) If object is a string.
'number' - (string) If object is a number.
'date' - (string) If object is a date.
'boolean' - (string) If object is a boolean.
'function' - (string) If object is a function.
'regexp' - (string) If object is a regular expression.
'class' - (string) If object is a Class (created with new Class or the extend of another class).

Web IDL

Web IDL

Web Interface Description Language

Samsung Web API Guide

http://img-developer.samsung.com/onlinedocs/samsung_webapi_guide/html/index.html

```
webapis.calendar.getDefaultCalendar("HELLO");
```

Method Signature

```
Calendar getDefaultCalendar(CalendarType type);
```

```
webapis.calendar.getDefaultCalendar("HELLO");
```

WebIDL Specification

```
enum CalendarType { "EVENT", "TASK" };
```

Method Signature

```
Calendar getDefaultCalendar(CalendarType type);
```

```
webapis.calendar.getDefaultCalendar("HELLO");
```

WebIDL Specification

```
enum CalendarType { "EVENT", "TASK" };
```

Method Signature

```
Calendar getDefaultCalendar(CalendarType type);
```

```
webapis.calendar.getDefaultCalendar("HELLO");
```

```
* Bug Detector *
example.js:1:1~1:45: [WebAPIError] Argument #1 of the function
webapis.calendar.getDefaultCalendar is wrong; the expected type
is CalendarType.
```

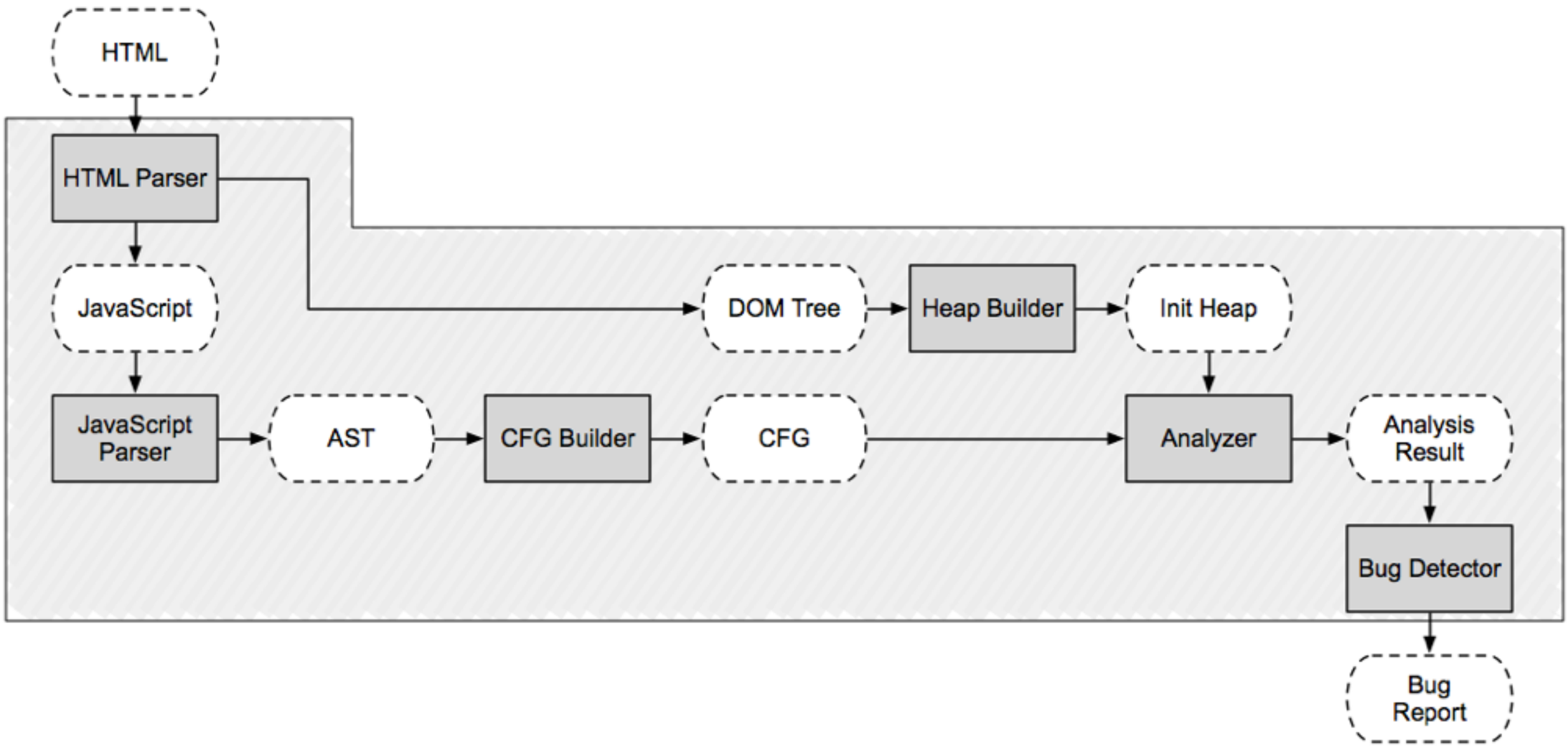
Web API 사용 결함 검출

- Web API에서 지원하지 않는 API 호출하는 경우
- API의 인자 수가 명세와 맞지 않는 경우
- 비동기로 동작하는 API의 error callback을 생략하는 경우
- 예외사항을 발생시킬 수 있는 API를 try-catch로 감싸지 않은 경우

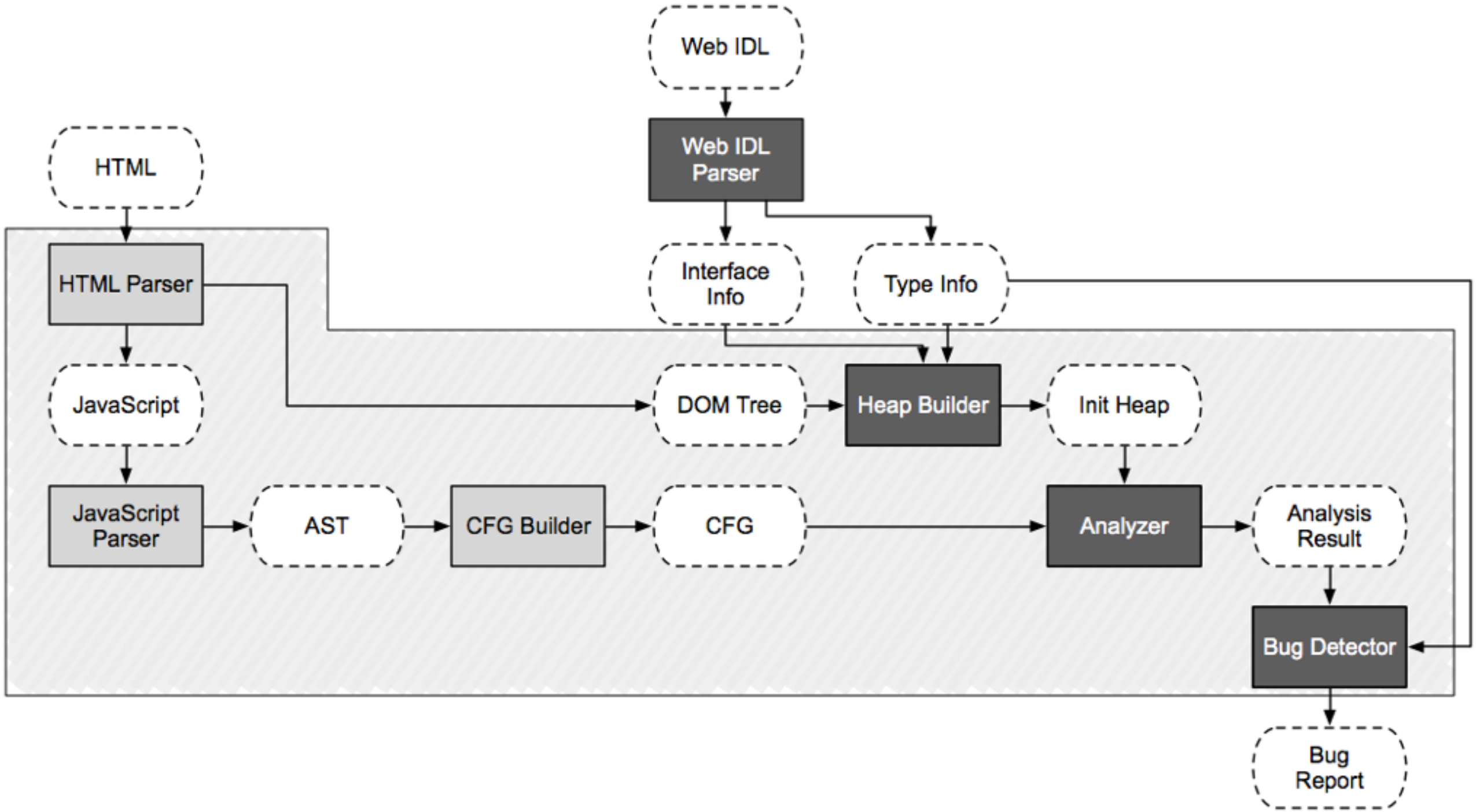
Web API 사용 결함 검출

- API의 인자로 명세와 다른 자료형의 값이 사용된 경우
- Callback으로 사용된 함수 내부에서 인자에 없는 property를 사용한 경우
- Dictionary 객체를 명세와 다르게 생성한 경우

SAFE


HTML
HTML Parser
JavaScript
JavaScript Parser
AST
CFG Builder
CFG
DOM Tree
Heap Builder
Init Heap
Analyzer
Analysis Result
Bug Detector
Bug Report

SAFEwapp


Web IDL
Web IDL Parser
HTML
HTML Parser
JavaScript
JavaScript Parser
AST
CFG Builder
CFG
Interface Info
Type Info
DOM Tree
Heap Builder
Init Heap
Analyzer
Analysis Result
Bug Detector
Bug Report

포스터 세션에서 만나요