

크로스-플랫폼 브라우저 플러그인 프레임워크 - FireBreath

AJ님께 소식듣고 정리해서 포스팅하려고 했는데 [Ajaxian](#)에서 벌써 소개했더군요. [FireBreath](#)는 윈도, 맥, 리눅스와 같은 각기 다른 플랫폼에서 파이어폭스, 크롬, 사파리, IE 브라우저 단위로 플러그인을 개발하는 수고를 덜어주는 프레임워크입니다. 브라우저 플러그인은 순수 웹 기술만으로 구현하는 것이 어려운 경우 이를 대신할 수 있는 소프트웨어를 연동하여 고차원 서비스를 브라우저를 통하여 제공하기 위한 것입니다. 예를 들면 쿼타임, 플래시, 실버라이트 등이 있겠네요.

NPAPI browsers on windows, mac, and linux:

- Gecko/Firefox
- Google Chrome
- Apple Safari

ActiveX Control hosts:

- Microsoft Internet Explorer 6, 7, and 8

그리고 아래와 같은 자바스크립트와 연동하기 위한 API도 제공하고 있어

여러모로 수고를 덜 수 있습니다. ActiveX로 인한 크로스-브라우저 이슈가 대두되고 있는 한국 웹문화에 도움될만한 대안이로군요!

JSAPI (Javascript API) class structure

The JSAPI object is the core object of the scripting architecture. All objects providing functionality that is exposed directly to Javascript extend FB::JSAPI

HasProperty, GetProperty, and SetProperty can be overridden to provide a fully dynamic set of properties on an object (i.e. it can change at runtime)

HasMethod and Invoke can be overridden to provide a fully dynamic set of methods on an object

If you only need a static interface on your object, the simplest way to provide functionality is with Handler functions.

Handler functions can be of three types:

- Method handlers
- Property Getters
- Property Setters

These must then be registered with the JSAPI in the constructor of the object



```
MyJSAPIObject::MyJSAPIObject(void)
{
    registerMethod( "toString", (CallMethodPtr)&MyJSAPIObject::callFindURL );
    registerProperty( "valid", (GetPropPtr)&MyJSAPIObject::getSomeProperty,
        (SetPropPtr)&MyJSAPIObject::setSomeProperty);
}
```

From any method inside MyJSAPIObject, you can fire an event by calling the FireEvent method:

```
this->FireEvent("onload", vector_list<FB::variant>("String param", 4, 23.3));
```

All event handlers that have been attached to this object from Javascript will be fired with the 3 specified parameters.