class FB JSAPIAuto FireJSEvent


```
void FB::JSAPIAuto::FireJSEvent ( const std::string &     eventName,
                                  const FB::VariantMap &  members,
                                  const FB::VariantList & arguments
                                )
```
virtual

Fires an event into javascript asynchronously using a W3C-compliant event parameter.

This fires an event to all handlers attached to the given event in javascript. With a W3C-compliant event parameter

IE:

```
document.getElementById("plugin").attachEvent("onload", function() { alert("loaded!"); });
```

Firefox/Safari/Chrome/Opera:

```
// Note that the convention used by these browsers is that "on" is implied
document.getElementById("plugin").addEventListener("load", function() { alert("loaded!");
}, false);;/.
```

You can then fire the event – from any thread – from the **JSAPI** object like so:

```
FireEvent("onload", FB::variant_list_of("param1")(2)(3.0));
```

Also note that registerEvent must be called from the constructor to register the event.

```
registerEvent("onload");
```

Parameters

- **eventName** Name of the event. This event must start with "on"
- **members**
- **arguments** The arguments that should be sent to each attached event handler

See Also

- **registerEvent**

Reimplemented from **FB::JSAPIImpl**.

Definition at line **426** of file **JSAPIAuto.cpp**.

References **FB::variant::cast()**, **FB::JSAPIImpl::FireJSEvent()**, and **FB::variant::is_of_type()**.

```
427 {
428  JSAPIImpl::FireJSEvent(eventName, members, arguments);
429  FB::variant evt(getAttribute(eventName));
430  if (evt.is_of_type<FB::JSObjectPtr>()) {
431  VariantList args;
432  args.push_back(FB::CreateEvent(shared_from_this(), eventName, members, arguments));
433  try {
434  evt.cast<JSObjectPtr>()->InvokeAsync("", args);
435  } catch (...) {
436  // Apparently either this isn't really an event handler or something failed.
437  }
438  }
439 }
```

FB::JSObjectPtr
boost::shared_ptr< FB::JSObject > JSObjectPtr
Defines an alias representing a JSObject shared_ptr (you should never use a JSObject* directly) ...
Definition: APITypes.h:109

FB::variant
Accepts any datatype, used in all interactions with javascript. Provides tools for getting back out t...
Definition: variant.h:198

FB::JSAPIImpl::FireJSEvent
virtual void FireJSEvent(const std::string &eventName, const FB::VariantMap &members, const FB::VariantList &arguments)
Fires an event into javascript asynchronously using a W3C-compliant event parameter.
Definition: JSAPIImpl.cpp:185

FB::VariantList
std::vector< variant > VariantList
Defines an alias representing list of variants.
Definition: APITypes.h:64

FB::JSAPIAuto::getAttribute
virtual FB::variant getAttribute(const std::string &name)
Returns the attribute with the given name, empty variant if none.
Definition: JSAPIAuto.cpp:406

Here is the call graph for this function: