

V8 Maybe APIs guidelines in Blink

bashi@chromium.org

2015-12-09

To address exception handling issues described in [this document](#), we are going to use [V8 Maybe APIs](#) and add empty checks. The contract of Maybe APIs is that there is an exception if and only if these APIs fail. Here are some guidelines for those who want to interact with V8 APIs in Blink.

- Use V8 Maybe APIs if you write new code. Don't use deprecated non-Maybe APIs (they are annotated with `V8_DEPRECATED_SOON`). In some cases, we can make sure that v8 APIs never fail except for fatal errors (e.g. stack overflow) but we want to have the same policy which we can apply everywhere in Blink. One exception is inspector related code, as there is an ongoing effort to move it out.
- Blink functions which directly return a result of V8 APIs should return a Maybe handle. This way we can make sure that call sites check empty handles.
- A Blink function should not return an empty Maybe handle without an exception. This aligns V8's behavior.
- If you just need to detect exceptions and don't need extra work, you don't need to use `v8::TryCatch`. You can detect exceptions by checking return values of `MaybeLocal::ToLocal()` or `Maybe::IsNothing()`.
- `toV8()` can return empty handles. You must check `IsEmpty()` before using return value.
- `v8String()` and `v8AtomicString()` never return empty handles. It immediately crashes when a fatal error occurs. Use `v8::String::NewFromUtf8()` directly if you need to handle exceptions.