

Allowing Wake Lock to Serve Browser Process Clients

The goal of the below changes is to:

- Allow for creating PowerSaveBlockers with arbitrary (type, reason, description) parameters in the WakeLockContext mojo interface while keeping the implementation as simple as possible
- Allow for replacing browser process usage of PowerSaveBlocker with usage of the Wake Lock mojo interface
- Continue to support having one PowerSaveBlocker per WebContents in which all connections from frames associated with that WebContents are coalesced

The way in which the below changes accomplish the goal is to:

- Move to a 1:1 mapping between PowerSaveBlocker and WakeLockServiceImpl
- Allow WakeLockService to serve an arbitrary number of clients, whose requests to block/unblock wake lock are coalesced in a similar fashion as WakeLockServiceContext does currently

New Mojo Interfaces

- wake_lock_context_provider.mojom
 - GetWakeLockContextForID(id)
 - GetUnassociatedWakeLockContext()
- wake_lock_context.mojom
 - GetWakeLock(WakeLockServiceRequest&, reason, description, type)
- wake_lock_service.mojom
 - RequestWakeLock()
 - CancelWakeLock()
 - AddClient(WakeLockServiceRequest&)

Wake Lock Impl Changes

- WakeLockServiceImpl now maintains a PowerSaveBlocker internally as well as a BindingSet for inbound requests.
 - The bulk of WakeLockServiceContext moves into this class

- We'll need to use the Context feature of BindingSet in order to track whether there is a wake lock request outstanding on a per-client basis, as WakeLockServiceImpl is currently doing
- WakeLockServiceContext simply holds onto its context ID and passes that to a WakeLockServiceImpl that it creates in GetWakeLock().
- There is a special null ID that is used by GetUnassociatedWakeLockContext().

//content Changes

- RenderFrameHostDelegate
 - Has ConnectToBlinkWakeLock(WakeLockServiceRequest& request) interface
 - No longer has GetWakeLockServiceContext() method
- WebContentsImpl
 - Has a WakeLockServicePtr blink_wake_lock_ ivar
 - Implements ConnectToBlinkWakeLock(request) as follows:
 - Initializes blink_wake_lock_ if necessary via wake_lock_context_host_
 - Calls blink_wake_lock_->AddConnection(request)
- RenderFrameHostImpl
 - Binds RenderFrameHostDelegate::ConnectToBlinkWakeLock() as the factory function for the WakeLockService mojo interface
- WebContents
 - Has a device::mojom::WakeLockContext* GetWakeLockContext() method

Browser Process Usage

With Access to a WebContents

- Call web_contents->GetWakeLockContext()->GetWakeLock(request, reason, description, type)

Without Access to a WebContents

- Connect to the Device Service and obtain a WakeLockContextProviderPtr
- Call GetUnassociatedWakeLockContext() on the WakeLockContextProvider
- Get the WakeLock from that unassociated context