WebBluetooth API draft

Firefox OS Bluetooth Team -
Eric Chou, Gina Yeh, Ben Tian,
Shawn Huang, and Jamin Liu
WebBluetooth API

- Allow to turn on/off BT, discover remote devices, and pair/unpair to remote devices
  “BEFORE CONNECTING”

- Use cases
  - Turn on/off BT
  - Discover remote devices
  - Pair/unpair to remote devices
Major Changes

• Turn on/off BT
  • mozSettings => BluetoothAdapter.enable()/disable()

• Discover remote devices
  • ondevicefound event handler => BluetoothDiscoveryHandle.ondevicefound

• Pair to remote devices
  • System messages => Event handlers
Interfaces
3 Main Interfaces

- **BluetoothManager**
  - Manage the adapters

- **BluetoothAdapter**
  - Represent local device

- **BluetoothDevice**
  - Represent remote device
Other Interfaces

- **BluetoothClassOfDevice**
  - Define capabilities of remote device

- **Events**
  - `BluetoothAdapterEvent = (isDefault, BluetoothAdapter?, address?)`
  - `BluetoothAttributeEvent = (attribute, value)`
  - `BluetoothDeviceEvent = (BluetoothDevice?, address?)`
  - `BluetoothPairingEvent = (BluetoothDevice, BluetoothPairingHandle)`
    - `BluetoothPairingHandle = (passkey?, setPasskey, setPairingConfirmation)`
Interfaces:

BluetoothManager

Interfaces:

BluetoothDevice

Use cases
Use case:

Turn on/off BT
## Turn on BT

<table>
<thead>
<tr>
<th>Gaia Application</th>
<th>Gecko</th>
<th>Bluetooth Backend</th>
<th>Remote Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>get default adapter</td>
<td>adapter = navigator.mozBluetooth.defaultAdapter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BT was off</td>
<td>adapter.enable()</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>adapter.onattributechanged(state=enabling)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BT is turned on</td>
<td>adapter.enable()</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>promise.onResolve</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>adapter.onattributechanged(name=“adapter1”)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>adapter.onattributechanged(state=enabled)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Random order
Turn off BT

**Gaia Application**
- BT was on: `adapter.disable()`
- BT is turned off: `adapter.disable()`

**Gecko**
- `adapter.onattributechanged(state=disabling)`
- `adapter.onattributechanged(state=disabled)`

**Bluetooth Backend**
- `promise.onReject`

**Remote Device**
- `promise.onResolve`

*random order*
BluetoothAdapterState

device boots

adapter.enable()

promise.onResolve

adapter.enable() / disable()

promise.onReject

enabling

enabled

disabling

adapter.disable()

promise.onReject

adapter.disable() / enable() / disable()
Use case:

Set properties
Set adapter name

Gaia Application

adapter.name was “bluetooth”

adapter.setName(“WebBluetooth”)

adapter.onattributechanged(name=“WebBluetooth”)

adapter.name becomes “WebBluetooth”

promise.onResolve

adapter.setName(“WebBluetooth”)

setting identical value gets error

promise.onReject

Gecko

Bluetooth Backend

Remote Device
Set adapter discoverable

adapter was undiscoverable

adapter.setDiscoverable(true)

adapter.onattributechanged(discoverable=true)

adapter becomes discoverable

promise.onResolve

setting identical value gets error

adapter.setDiscoverable(true)

promise.onReject
Use case:

Discover remote devices
Discover remote devices

<table>
<thead>
<tr>
<th>Gaia Application</th>
<th>Gecko</th>
<th>Bluetooth Backend</th>
<th>Remote Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapter.startDiscovery()</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>adapter.onAttributechanged(discovering=true)</td>
<td>promise.onResolve(handle)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>handle.onDevicefound(device=remote device 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>handle.onDevicefound(device=remote device 2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>handle.onDevicefound(device=remote device 3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>adapter.onAttributechanged(discovering=false)</td>
</tr>
</tbody>
</table>
Use case:

Pair to remote devices
Pair – display passkey

Gaia Application

start pairing: `adapter.pair(address)`

display handle.passkey to user: `adapter.pairingReqs.onDisplayPasskeyReq(device, handle)`

Gecko

device.onattributechanged(paired=true)

Bluetooth Backend

adapter.onDevicePaired(device=remote device)

Remote Device

promise.onResolve
Pair – enter passkey

Gaia Application

- start pairing: `adapter.pair(address)`
- user enters passkey:
  - `adapter.pairingReqs.onPairingConfirmationReq(device, handle)`
  - `promise2 = handle.setPasskey("123456")`
  - `promise2.onResolve`

Gecko

Bluetooth Backend

Remote Device

- `device.onattributechanged(paired=true)`
- `adapter.ondevicepaired(device=remote device)`
- `promise.onResolve`
Pair - confirmation

**Start pairing**

Gaia Application:
- `adapter.pair(address)`

Gecko:
- `adapter.pairingReqs.onPairingConfirmationReq(device, handle)`

Bluetooth Backend:
- `promise2 = handle.setPairingConfirmation(true)`
- `promise2.onResolve`

Remote Device:
- `device.onattributechanged(paired=true)`
- `adapter.onDevicePaired(device=remote device)`
- `promise.onResolve`

**User confirms handle.passkey**

Gaia Application:
- `adapter.pairingReqs.onPairingConfirmationReq(device, handle)`
- `promise2 = handle.setPairingConfirmation(true)`
- `promise2.onResolve`

Gecko:
- `device.onattributechanged(paired=true)`

Bluetooth Backend:
- `adapter.onDevicePaired(device=remote device)`
- `promise.onResolve`

Remote Device:
- `promise.onResolve`
Pair – consent

**Gaia Application**

- **start pairing**
  - `adapter.pair(address)`

- **device**
  - `onattributechanged(paired=true)`

- **adapter**
  - `ondevicepaired(device=remote device)`

- **complete pairing**
  - `promise.onResolve`

**Gecko**

**Bluetooth Backend**

**Remote Device**

- `adapter.pairingReqs.onpairingconsentreq(device, handle)`
Bluetooth ON/OFF (From “state” perspective)

- Disabled
  - `enable() / reg.error` 
  - `reg.onerror`
  - `req.success` 
- Enabling 
  - `enable() / reg.error` 
  - `reg.onerror` 
  - `req.onsuccess`
- Enabled
  - `reg.onerror`
  - `req.onsuccess`
  - `disable()`
- Disabling
  - `reg.onerror`
  - `req.onsuccess`

If event “attributechanged” would be fired up whenever a new state is reached.
In addition, it must be fired before `reg.onsuccess` in case `enabling -> enabled` and `disabling -> disabled`.