Re: WebBluetooth API discussion R3

From : Ben Tian <btian@mozilla.com> Fri, Apr 18, 2014 03:31 PM
Subject : Re: WebBluetooth API discussion R3
To : Evelyn Hung <ehung@mozilla.com>, Arthur Chen <achen@mozilla.com>, Ian Liu <iliu@mozilla.com>, Gene Lian <clian@mozilla.com>, Alive Kuo <akuo@mozilla.com>
Cc : Jamin Liu <jaLiu@mozilla.com>, Eric Chou <echou@mozilla.com>, Gina Yeh <gyeh@mozilla.com>, Shawn Huang <shuang@mozilla.com>, Jocelyn Liu <joliu@mozilla.com>

Hi all,

Please check the latest WebBluetooth API draft on https://wiki.mozilla.org/B2G/Bluetooth/WebBluetooth-v2. The API has been revised based on Promise. Also the following is BT team's response to R3 suggestions. Let us know if you have any question.

--

6) About the memory life cycle of DOMRequest & Promise. [API REVISED]
BT team will investigate it and make sure the API is OK for *DOMRequest* or *Promise implementation*.

> We revise adapter.startDiscovery() since original approach (BluetoothStartDiscoveryRequest extends DOMRequest + *ondevicefound* event handler) is not suitable for Promise. Please refer to [4] for the revision and sample code. The new approach directly returns a object that contains *ondevicefound* event handler. Applications can receive *ondevicefound* event each time a remote bluetooth device is discovered.
>
> [4] https://wiki.mozilla.org/B2G/Bluetooth/WebBluetooth-v2/BluetoothAdapter#startDiscovery.28.29

2) Should we put properties and methods into *'detail'* **of BluetoothPairingEvent ?**

Evelyn: It seems like a convention to put properties and methods into 'detail' object. **(Evelyn)** However, it's not defined in spec., let leave this one to BT team.BT team will investigate it and make the decision.

> NO. We decide NOT to wrap properties and methods in *detail* since most existing webidl uses *detail* as a long attribute rather than a general object. For example, both UIEvent [1] and TimeEvent [2] use *detail* as long attribute, while CustomEvent [3] is the only event using it as an any attribute. Therefore we keep our original event structure and not to put them into *detail*.
>
> [1] http://dxr.mozilla.org/mozilla-central/source/dom/webidl/UIEvent.webidl?from=UIEvent.webidl#19
> [2] http://dxr.mozilla.org/mozilla-central/source/dom/webidl/TimeEvent.webidl?from=TimeEvent.webidl#17
> [3] http://dxr.mozilla.org/mozilla-central/source/dom/webidl/CustomEvent.webidl?from=CustomEvent.webidl&case=true#

--
Ben

From: "Evelyn Hung" <ehung@mozilla.com>
To: "Jamin Liu" <jaLiu@mozilla.com>, "Ben Tian" <btian@mozilla.com>
Cc: "Eric Chou" <echou@mozilla.com>, "Gina Yeh" <gyeh@mozilla.com>, "Shawn Huang" <shuang@mozilla.com>, "Arthur Chen" <achen@mozilla.com>, "Gene Lian" <glian@mozilla.com>, "Jocelyn Liu" <joliu@mozilla.com>, "Ian Liu" <iliu@mozilla.com>, "Alive Kuo" <akuo@mozilla.com>
Sent: Tuesday, April 15, 2014 7:00:39 PM
Subject: Re: WebBluetooth API discussion R3

Thank Jamin and Ben for preparing, hosting and follow-up this meeting, and thank BT team for cooking the draft.
I think we were very productive in these meetings because of your rich materials :)

Thanks,
Evelyn

> Hi all,
>
> Here is the meeting minutes of our WebBluetooth API discussion R3.
> Thank you for your attendance.

1) Which type should the *passkey* be ? *unsigned long* or *DOMString* ?

Gene: Why *passkey* is *unsigned long* at here [1], but it's *DOMString* at here [2].

→ Ben & Shawn: They means different thing. It's a naming problem.
[1] can only be a number between 0 ~ 999999.
[2] could be number or ASCII string, it combines *setPasskey()* and *setPinCode()* .

```
interface BluetoothPairingHandle
{
readonly attribute unsigned long? passkey;

DOMRequest setPasskey(DOMString aPasskey);
DOMRequest setPairingConfirmation(boolean aConfirm);
};
```

Gene & Evelyn: Should we use *DOMString* for [1] even it's a unsigned number ?
→ Yes

→ ***Action Item:*** BT team would modify the wiki page.
readonly attribute unsigned long? passkey; → *readonly attribute DOMString? passkey;*

2) Should we put properties and methods into *'detail'* of [BluetoothPairingEvent](#) ?

Evelyn: It seems like a convention to put properties and methods into 'detail' object. **(Evelyn)**
However, it's not defined in spec., let leave this one to BT team.

Option 1: ('detail' version)

```
interface BluetoothPairingEvent : Event
{
readonly attribute BluetoothPairingHandle detail;
};

interface BluetoothPairingHandle
{
readonly attribute BluetoothDevice device;
  readonly attribute DOMString? passkey;
DOMRequest setPasskey(DOMString aPasskey);
DOMRequest setPairingConfirmation(boolean aConfirm);
};
```

Option 2: (current version)

```
interface BluetoothPairingEvent : Event
{
readonly attribute BluetoothDevice device;
  readonly attribute DOMString? passkey;

DOMRequest setPasskey(DOMString aPasskey);
DOMRequest setPairingConfirmation(boolean aConfirm);
};
```

→ ***Action Item:*** BT team will investigate it and make the decision.
A brief report will be sent by email recently.

3) Should the properties be null if it invalid ?

Evelyn: Since *passkey* is useless in *onpairingconsentreq*, it should be null in that case.
→ **Yes.**
e.g. *passkey* would be null if it's invalid.

```
interface BluetoothPairingHandle
{
  readonly attribute DOMString? passkey; → Is nullable
. . .
};
```

Gina: Does it means we should assign null to every invalid properties?
→ **No**. (We assign null only if it's a optional proverty.)
e.g. *name* would be default value if it's invalid.

```
interface BluetoothAdapter: EventTarget
{
readonly attribute BluetoothAdapterState state;
readonly attribute DOMString address;
readonly attribute DOMString name; → Not nullable
readonly attribute boolean discoverable;
```

> readonly attribute boolean discovering;
> . . .
>
> };
>
> → *(Remain Unchanged)*
>
> **5) It could be a problem if every privilege app can call stopDiscovery()**
> (Evelyn)
>
> Eric: The same situation would happen on BT Enable/Disable too.
> If there is any plan to restrict BT Enable/Disable to certain app in future, the start/stopDiscovery would follow the same rule.
> → *(Remain Unchanged)*
>
> **6) About the memory live cycle of DOMRequest**
>
> > Please refet to Ben's PDF https://wiki.mozilla.org/images/5/55/Webbluetooth-response-0314.pdf
> > *20) If we put devicefound listener into req of startDiscovery(), we have to hold that req object in some place for unregister purpose. (Evelyn)*
> > *21) When does DOM req end? When the onsuccess is fired or when it exits its live cycle? (Eric)*
> > *YES, gaia has to hold the reference to the DOM request. The DOM request exists until both gaia and gecko release references to it. So as long as gaia and gecko holds the DOM request's references, gaia can always receive the ondevicedfound event fired by gecko.*
>
> → ***Action Item:*** **BT team will investigate it and make sure the API is OK for *DOMReuest* or *Promise implementation*.**
>
> Regards,
> Jamin
>
> On 2014年04月03日 18:48, Ben Tian wrote:
>
> > **The following is a new meeting request:**
> >
> > **Subject:** WebBluetooth API discussion R3

Organizer: "Ben Tian" <btian@mozilla.com>

Location: "TPE-4H Eden" <tpe-4h@mozilla.com>

Resources: "TPE-4H Eden" <tpe-4h@mozilla.com>

Time: Thursday, April 10, 2014, 2:00:00 PM - 4:00:00 PM GMT +08:00 Taipei

Invitees: echou@mozilla.com; gyeh@mozilla.com; shuang@mozilla.com; jaliu@mozilla.com; joliu@mozilla.com; achen@mozilla.com; ehung@mozilla.com; glian@mozilla.com; iliu@mozilla.com; akuo@mozilla.com

~~*~*~*~*~*~*~*~*

Hi all,

Let's have round 3 WebBluetooth API discussion next Thu. Today is a good sign that we start having agreements. Hope we can finish it next week.

I'll revise WebBluetooth API draft based on today's conclusion and send out by Thu. Meanwhile please check out [1] for our reply on prior questions/suggestions and let me know for any feedback. Thanks.

[1] https://wiki.mozilla.org/images/5/55/Webbluetooth-response-0314.pdf

Ben

--
Jamin Liu
Software Engineer | mozilla Taiwan
Tel:+886-2-87861100 ext.219