Sketch of navigator.bluetooth.getDevices()

Public document; may move to https://github.com/WebBluetoothCG/web-bluetooth/.

Use cases

- User selects an Eddystone-URI beacon from their Eddystone picker. That opens a web page. The web page uses getDevices() to retrieve the right device without needing to show a redundant requestDevice() dialog.
- User selects a device from the requestDevice() dialog to pair with a web page. User closes and then re-opens the web page. Web page uses getDevices() to retrieve the device so the user doesn't need to pick it again.

Interface

```
partial interface Bluetooth {

  Promise<readonly Map<DOMString, BluetoothDevice>> getDevices();

// or:

  Promise<BluetoothDevice> getDevice(DOMString identifier);

  Promise<sequence<BluetoothDevice>> getDevices();

}
```

getDevices() MUST:

1. Let *devices* be a new array.
2. For each key *device* in the current origin's allowed devices map, get the BluetoothDevice representing *device*, and add the result to *devices*.
3. If getDevices() returns a sequence, return Promise.all(*devices*). If it returns a Map, return Promise.all(devices).then(devices => Map([[device.id, device] for device in devices])).

We need to keep the site updated with changes to the result of getDevices(). Ideally, I think I'd like to use Object.observe to do that, but Object.observe(Map) may not work without changes to ECMAScript, which is a big risk. If we can't or don't want to use Object.observe, we should have a `deviceschanged` event to alert the site that their pairings have changed.

getDevice(id) MUST:

1. Let *device* be the device in the current origin's allowed devices map whose device id is *id*.
2. If no such device was found, return a Promise rejected with a NotFoundError.
3. Otherwise get the BluetoothDevice representing *device* and return the result.

Bikeshed

- getDevices
- getPairedDevices
- getAllowedDevices