

XRSession supported features

The Problem

Many features have some sort of “sentinel value” you can query to determine if an optional feature was enabled. (ie: `requestReferenceSpace('bounded-floor')` to check if you got the ‘bounded-floor’ feature.)

But not every feature has a natural associated query and even in cases where we do the results can be awkward to use (need to wait for a promise, sometimes null is a valid result, etc).

The anchors API is an example of this, where the ``trackedAnchors`` set-like may be empty if anchors are not supported OR if there are no anchors tracked.

Proposed solution

```
interface XRSupportedFeatures {  
    readonly setlike<DOMString>;  
};
```

```
partial interface XRSession {  
    [SameObject] readonly attribute XRSupportedFeatures features;  
};
```

Proposed solution cont.

```
if (xrSession.features.has('some-feature')) {  
  // Use some-feature  
} else {  
  console.warn('some-feature is not supported by this session.');
```

Potential issue

- Features passed into a session are not currently required to be a string.
- Maybe we should start requiring that? (It's functionally the case today.)
- DOM Overlay shows a good path forward for handling the more complex cases:

```
navigator.xr.requestSession('immersive-ar', {  
  optionalFeatures: ['dom-overlay'],  
  domOverlay: { root: uiElement }  
});
```