Zone.js's support for standard apis

Zone.js patched most standard APIs such as DOM event listeners, XMLHttpRequest in Browser, EventEmitter and fs API in Node.js so they can be in zone.

In this document, all patched API are listed.

For non-standard APIs, please see NON-STANDARD-APIS.md

Patch Mechanisms

There are several patch mechanisms

- wrap: makes callbacks run in zones, and makes applications able to receive onInvoke and onIntercept callbacks
- Task: just like in the JavaScript VM, applications can receive onScheduleTask, onInvokeTask, onCancelTask and onHasTask callbacks
    1. MacroTask
    2. MicroTask
    3. EventTask

Some APIs which should be treated as Tasks, but are currently still patched in the wrap way. These will be patched as Tasks soon.

Browser

Web APIs

API	Patch Mechanism	Others
setTimeout/clearTimeout	MacroTask	app can get handlerId, interval, args, isPeriodic(false) through task.data
setImmediate/clearImmediate	MacroTask	same with setTimeout
setInterval/clearInterval	MacroTask	isPeriodic is true, so setInterval will not trigger onHasTask callback
requestAnimationFrame/cancelAnimationFrame	MacroTask	
mozRequestAnimationFrame/mozCancelAnimationFrame	MacroTask	
webkitRequestAnimationFrame/webkitCancelAnimationFrame	MacroTask	
alert	wrap	
prompt	wrap	
confirm	wrap	
Promise	MicroTask	

EventTarget	EventTask	see below Event Target for more details
HTMLElement on properties	EventTask	see below on properties for more details
XMLHttpRequest.send/abort	MacroTask	
XMLHttpRequest on properties	EventTask	
IDBIndex on properties	EventTask	
IDBRequest on properties	EventTask	
IDBOpenDBRequest on properties	EventTask	
IDBDatabaseRequest on properties	EventTask	
IDBTransaction on properties	EventTask	
IDBCursor on properties	EventTask	
WebSocket on properties	EventTask	
MutationObserver	wrap	
WebkitMutationObserver	wrap	
FileReader	wrap	
registerElement	wrap	

EventTarget

- For browsers supporting EventTarget, Zone.js just patches EventTarget, so everything that inherits from EventTarget will also be patched.
- For browsers that do not support EventTarget, Zone.js will patch the following APIs in the IDL that inherit from EventTarget

ApplicationCache	EventSource	FileReader	InputMethodContext
MediaController	MessagePort	Node	Performance
SVGElementInstance	SharedWorker	TextTrack	TextTrackCue
TextTrackList	WebKitNamedFlow	Window	Worker
WorkerGlobalScope	XMLHttpRequest	XMLHttpRequestEventTarget	XMLHttpRequestUpload
IDBRequest	IDBOpenDBRequest	IDBDatabase	IDBTransaction
IDBCursor	DBIndex	WebSocket	

The following 'on' properties, such as onclick, onreadystatechange, are patched in Zone.js as EventTasks

copy	cut	paste	abort

blur	focus	canplay	canplaythrough
change	click	contextmenu	dblclick
drag	dragend	dragenter	dragleave
dragover	dragstart	drop	durationchange
emptied	ended	input	invalid
keydown	keypress	keyup	load
loadeddata	loadedmetadata	loadstart	message
mousedown	mouseenter	mouseleave	mousemove
mouseout	mouseover	mouseup	pause
play	playing	progress	ratechange
reset	scroll	seeked	seeking
select	show	stalled	submit
suspend	timeupdate	volumechange	waiting
mozfullscreenchange	mozfullscreenerror	mozpointerlockchange	mozpointerlockerror
error	webglcontextrestored	webglcontextlost	webglcontextcreationerror

NodeJS

API	Patch Mechanism	Others
setTimeout/clearTimeout	MacroTask	app can get handlerId, interval, args, isPeriodic(false) through task.data
setImmediate/clearImmediate	MacroTask	same with setTimeout
setInterval/clearInterval	MacroTask	isPeriodic is true, so setInterval will not trigger onHasTask callback
process.nextTick	Microtask	isPeriodic is true, so setInterval will not trigger onHasTask callback
Promise	MicroTask	
EventEmitter	EventTask	All APIs inherit EventEmitter are patched as EventTask
crypto	MacroTask	
fs	MacroTask	all async methods are patched

EventEmitter, addEventListener, prependEventListener and 'on' will be patched once as EventTasks, and removeEventListener and removeAllListeners will remove those EventTasks

Electron

Zone.js does not patch the Electron API, although in Electron both browser APIs and node APIs are patched, so if you want to include Zone.js in Electron, please use dist/zone-mix.js

ZoneAwareError

ZoneAwareError replaces global Error, and adds zone information to stack trace. ZoneAwareError also handles 'this' issue. This type of issue would happen when creating an error without `new` : `this` would be `undefined` in strict mode, and `global` in non-strict mode. It could cause some very difficult to detect issues.

```
const error = Error();
```

ZoneAwareError makes sure that `this` is ZoneAwareError even without new.

ZoneAwarePromise

ZoneAwarePromise wraps the global Promise and makes it run in zones as a MicroTask. It also passes promise A+ tests.

UnpatchedEvents

Sometimes we don't want some `event` to be patched by `zone.js` , we can instruct zone.js to leave these `event` to be unpatched by following settings.

```
// disable on properties
var targets = [window, Document, HTMLBodyElement, HTMLElement];
__Zone_ignore_on_properties = [];
targets.forEach(function (target) {
  __Zone_ignore_on_properties.push({
    target: target,
    ignoreProperties: ['scroll']
  });
});

// disable addEventListener
global['__zone_symbol__UNPATCHED_EVENTS'] = ['scroll'];
```