DEV NOTES / RESEARCH / OBSERVATION

Source:

https://developer.mozilla.org/enUS/docs/Web/API/HTML_Drag_and_Drop_API

HTML drag-and-drop uses the DOM event model and *drag events* inherited from mouse events. A typical *drag operation* begins when a user selects a *draggable* element, drags the element to a *droppable* element, and then releases the dragged element.

During drag operations, several event types are fired, and some events might fire many times, such as the drag and dragover events.

Each drag event type has an associated global event handler:

drag	ondrag	…a *dragged item* (element or text selection) is dragged.
dragend	ondragend	…a drag operation ends (such as releasing a mouse button or hitting the Esc key; see Finishing a Drag.)
dragenter	ondragenter	…a dragged item enters a valid drop target. (See Specifying Drop Targets.)
dragexit	ondragexit	…an element is no longer the drag operation's immediate selection target.
dragleave	ondragleave	…a dragged item leaves a valid drop target.
dragover	ondragover	…a dragged item is being dragged over a valid drop target, every few hundred milliseconds.
dragstart	ondragstart	…the user starts dragging an item. (See Starting a Drag Operation.)
drop	ondrop	…an item is dropped on a valid drop target. (See Performing a Drop.)

INTERFACES:

The HTML Drag and Drop interfaces are DragEvent, DataTransfer, DataTransferItem and DataTransferItemList.

The DragEvent interface has a constructor and one dataTransfer property, which is a DataTransfer object.

DataTransfer objects include the drag event's state, such as the type of drag being done (like copy or move), the drag's data (one or more items), and the MIME type of each *drag item*. DataTransfer objects also have methods to add or remove items to the drag's data.

The DragEvent and DataTransfer interfaces should be the only ones needed to add HTML Drag and Drop capabilities to an application. (Firefox supports some Gecko-specific extensions to the DataTransfer object, but those extensions will only work on Firefox.)

Each DataTransfer object contains an items property, which is a list of DataTransferItem objects. A DataTransferItem object represents a single *drag item*, each with a kind property (either string or file) and a type property for the data item's MIME type. The DataTransferItem object also has methods to get the drag item's data.

The DataTransferItemList object is a list of DataTransferItem objects. The list object has methods to add a drag item to the list, remove a drag item from the list, and clear the list of all drag items.

A key difference between the DataTransfer and DataTransferItem interfaces is that the former uses the synchronous getData() method to access a drag item's data, but the latter instead uses the asynchronous getAsString() method.