Module: pdfjsLib

Classes

PDFDataRangeTransport
PDFDocumentProxy
PDFPageProxy
RenderTask

Members

(inner, constant) build :string

Type:

- string

Source: display/api.js, line 3229

(inner, constant) PDFWorker :any

Type:

- any

Source: display/api.js, line 1900

(inner, constant) version :string

Type:

- string

Source: display/api.js, line 3226

Methods

(inner) getDocument(src) → {PDFDocumentLoadingTask}

This is the main entry point for loading a PDF and interacting with it. NOTE: If a URL is used to fetch the PDF data a standard Fetch API call (or XHR as fallback) is used, which means it must follow same origin rules, e.g. no cross-domain requests without CORS.

Parameters:

Name	Type	Description
src	string | URL | TypedArray | PDFDataRangeTransport | DocumentInitParameters	Can be a URL where a PDF file is located, a typed array (Uint8Array) already populated with data, or a parameter object.

Source: display/api.js, line 221

Returns:

Type
PDFDocumentLoadingTask

Type Definitions

DocumentInitParameters

Document initialization / loading parameters object.

Type:

- Object

Properties:

Name	Type	Attributes	Description
url	string | URL	<optional>	The URL of the PDF.
data	TypedArray | Array.<number> | string	<optional>	Binary PDF data. Use typed arrays (Uint8Array) to improve the memory usage. If PDF data is BASE64-encoded, use `atob()` to convert it to a binary string first.
httpHeaders	Object	<optional>	Basic authentication headers.
withCredentials	boolean	<optional>	Indicates whether or not cross-site Access-Control requests should be made using credentials such as cookies or authorization headers. The default is `false`.
password	string	<optional>	For decrypting password-protected PDFs.
initialData	TypedArray	<optional>	A typed array with the first portion or all of the pdf data. Used by the extension since some data is already loaded before the switch to range requests.
length	number	<optional>	The PDF file length. It's used for progress reports and range requests operations.
range	PDFDataRangeTransport	<optional>	Allows for using a custom range transport implementation.
rangeChunkSize	number	<optional>	Specify maximum number of bytes fetched per range request. The default value is DEFAULT_RANGE_CHUNK_SIZE.
worker	PDFWorker	<optional>	The worker that will be used for loading and parsing the PDF data.

Name	Type	Attributes	Description
verbosity	number	<optional>	Controls the logging level; the constants from VerbosityLevel should be used.
docBaseUrl	string	<optional>	The base URL of the document, used when attempting to recover valid absolute URLs for annotations, and outline items, that (incorrectly) only specify relative URLs.
cMapUrl	string	<optional>	The URL where the predefined Adobe CMaps are located. Include the trailing slash.
cMapPacked	boolean	<optional>	Specifies if the Adobe CMaps are binary packed or not.
CMapReaderFactory	Object	<optional>	The factory that will be used when reading built-in CMap files. Providing a custom factory is useful for environments without Fetch API or `XMLHttpRequest` support, such as Node.js. The default value is {DOMCMapReaderFactory}.
useSystemFonts	boolean	<optional>	When `true`, fonts that aren't embedded in the PDF document will fallback to a system font. The default value is `true` in web environments and `false` in Node.js.
standardFontDataUrl	string	<optional>	The URL where the standard font files are located. Include the trailing slash.
StandardFontDataFactory	Object	<optional>	The factory that will be used when reading the standard font files. Providing a custom factory is useful for environments without Fetch API or `XMLHttpRequest` support, such as Node.js. The default value is {DOMStandardFontDataFactory}.
useWorkerFetch	boolean	<optional>	Enable using the Fetch API in the worker-thread when reading CMap and standard font files. When `true`, the `CMapReaderFactory` and `StandardFontDataFactory` options are ignored. The default value is `true` in web environments and `false` in Node.js.

Name	Type	Attributes	Description
stopAtErrors	boolean	<optional>	Reject certain promises, e.g. `getOperatorList`, `getTextContent`, and `RenderTask`, when the associated PDF data cannot be successfully parsed, instead of attempting to recover whatever possible of the data. The default value is `false`.
maxImageSize	number	<optional>	The maximum allowed image size in total pixels, i.e. width * height. Images above this value will not be rendered. Use -1 for no limit, which is also the default value.
isEvalSupported	boolean	<optional>	Determines if we can evaluate strings as JavaScript. Primarily used to improve performance of font rendering, and when parsing PDF functions. The default value is `true`.
disableFontFace	boolean	<optional>	By default fonts are converted to OpenType fonts and loaded via the Font Loading API or `@font-face` rules. If disabled, fonts will be rendered using a built-in font renderer that constructs the glyphs with primitive path commands. The default value is `false` in web environments and `true` in Node.js.
fontExtraProperties	boolean	<optional>	Include additional properties, which are unused during rendering of PDF documents, when exporting the parsed font data from the worker-thread. This may be useful for debugging purposes (and backwards compatibility), but note that it will lead to increased memory usage. The default value is `false`.
enableXfa	boolean	<optional>	Render Xfa forms if any. The default value is `false`.
ownerDocument	HTMLDocument	<optional>	Specify an explicit document context to create elements with and to load resources, such as fonts, into. Defaults to the current document.
disableRange	boolean	<optional>	Disable range request loading of PDF files. When enabled, and if the server supports partial content requests, then the PDF will be fetched in chunks. The default value is `false`.

Name	Type	Attributes	Description
`disableStream`	boolean	<optional>	Disable streaming of PDF file data. By default PDF.js attempts to load PDF files in chunks. The default value is `false`.
`disableAutoFetch`	boolean	<optional>	Disable pre-fetching of PDF file data. When range requests are enabled PDF.js will automatically keep fetching more data even if it isn't needed to display the current page. The default value is `false`. NOTE: It is also necessary to disable streaming, see above, in order for disabling of pre-fetching to work correctly.
`pdfBug`	boolean	<optional>	Enables special hooks for debugging PDF.js (see `web/debugger.js`). The default value is `false`.

Source: display/api.js, line 117

`GetAnnotationsParameters`

Page annotation parameters.

Type:

- Object

Properties:

Name	Type	Description
`intent`	string	Determines the annotations that will be fetched, can be either 'display' (viewable annotations) or 'print' (printable annotations). If the parameter is omitted, all annotations are fetched.

Source: display/api.js, line 1124

`getTextContentParameters`

Page getTextContent parameters.

Type:

- Object

Properties:

Name	Type	Attributes	Description
`normalizeWhitespace`	boolean		Replaces all occurrences of whitespace with standard spaces (0x20). The default value is `false`.

Name	Type	Attributes	Description
`disableCombineTextItems`	boolean		Do not attempt to combine same line TextItem's. The default value is `false`.
`includeMarkedContent`	boolean	<optional>	When true include marked content items in the items array of TextContent. The default is `false`.

Source: display/api.js, line 1067

`GetViewportParameters`

Page getViewport parameters.

Type:

- Object

Properties:

Name	Type	Attributes	Description
`scale`	number		The desired scale of the viewport.
`rotation`	number	<optional>	The desired rotation, in degrees, of the viewport. If omitted it defaults to the page rotation.
`offsetX`	number	<optional>	The horizontal, i.e. x-axis, offset. The default value is `0`.
`offsetY`	number	<optional>	The vertical, i.e. y-axis, offset. The default value is `0`.
`dontFlip`	boolean	<optional>	If true, the y-axis will not be flipped. The default value is `false`.

Source: display/api.js, line 1052

`MarkInfo`

Properties correspond to Table 321 of the PDF 32000-1:2008 spec.

Type:

- Object

Properties:

Name	Type	Description
`Marked`	boolean	
`UserProperties`	boolean	
`Suspects`	boolean	

Source: display/api.js, line 917

OutlineNode

Type:

- Object

Properties:

Name	Type	Description
title	string	
bold	boolean	
italic	boolean	
color	Uint8ClampedArray	The color in RGB format to use for display purposes.
dest	string | Array.<any> | null	
url	string | null	
unsafeUrl	string | undefined	
newWindow	boolean | undefined	
count	number | undefined	
items	Array.<OutlineNode>	

> Source: display/api.js, line 865

PDFDocumentLoadingTask

The loading task controls the operations required to load a PDF document (such as network requests) and provides a way to listen for completion, after which individual pages can be rendered.

Type:

- Object

Properties:

Name	Type	Attributes	Description
docId	string		Unique identifier for the document loading task.
destroyed	boolean		Whether the loading task is destroyed or not.
onPassword	function	<optional>	Callback to request a password if a wrong or no password was provided. The callback receives two parameters: a function that should be called with the new password, and a reason (see PasswordResponses).

Name	Type	Attributes	Description
onProgress	function	<optional>	Callback to be able to monitor the loading progress of the PDF file (necessary to implement e.g. a loading bar). The callback receives an {Object} with the properties `loaded` ({number}) and `total` ({number}) that indicate how many bytes are loaded.
onUnsupportedFeature	function	<optional>	Callback for when an unsupported feature is used in the PDF document. The callback receives an UNSUPPORTED_FEATURES argument.
promise	Promise. <PDFDocumentProxy>		Promise for document loading task completion.
destroy	function		Abort all network requests and destroy the worker. Returns a promise that is resolved when destruction is completed.

Source: display/api.js, line 508

PDFDocumentStats

Type:

- Object

Properties:

Name	Type	Description
streamTypes	Object. <string, boolean>	Used stream types in the document (an item is set to true if specific stream ID was used in the document).
fontTypes	Object. <string, boolean>	Used font types in the document (an item is set to true if specific font ID was used in the document).

Source: display/api.js, line 951

PDFOperatorList

PDF page operator list.

Type:

- Object

Properties:

Name	Type	Description
`fnArray`	Array.<number>	Array containing the operator functions.
`argsArray`	Array.<any>	Array containing the arguments of the functions.

Source: display/api.js, line 1185

`PDFWorkerParameters`

Type:

- Object

Properties:

Name	Type	Attributes	Description
`name`	string	<optional>	The name of the worker.
`port`	Object	<optional>	The `workerPort` object.
`verbosity`	number	<optional>	Controls the logging level; the constants from VerbosityLevel should be used.

Source: display/api.js, line 1891

`RefProxy`

Type:

- Object

Properties:

Name	Type	Description
`num`	number	
`gen`	number	

Source: display/api.js, line 759

`RenderParameters`

Page render parameters.

Type:

- Object

Properties:

Name	Type	Attributes	Description
`canvasContext`	Object		A 2D context of a DOM Canvas object.
`viewport`	PageViewport		Rendering viewport obtained by calling the `PDFPageProxy.getViewport` method.
`intent`	string	<optional>	Rendering intent, can be 'display' or 'print'. The default value is 'display'.

Name	Type	Attributes	Description
renderInteractiveForms	boolean	<optional>	Whether or not interactive form elements are rendered in the display layer. If so, we do not render them on the canvas as well. The default value is `false`.
transform	Array.<any>	<optional>	Additional transform, applied just before viewport transform.
imageLayer	Object	<optional>	An object that has `beginLayout`, `endLayout` and `appendImage` functions.
canvasFactory	Object	<optional>	The factory instance that will be used when creating canvases. The default value is {new DOMCanvasFactory()}.
background	Object | string	<optional>	Background to use for the canvas. Any valid `canvas.fillStyle` can be used: a `DOMString` parsed as CSS value, a `CanvasGradient` object (a linear or radial gradient) or a `CanvasPattern` object (a repetitive image). The default value is 'rgb(255,255,255)'.
includeAnnotationStorage	boolean	<optional>	Render stored interactive form element data, from the AnnotationStorage-instance, onto the canvas itself; useful e.g. for printing. The default value is `false`.
optionalContentConfigPromise	Promise.<OptionalContentConfig>	<optional>	A promise that should resolve with an OptionalContentConfig created from `PDFDocumentProxy.getOptionalContentConfig`. If `null`, the configuration will be fetched automatically with the default visibility states set.

Source: display/api.js, line 1133

StructTreeContent

Structure tree content.

Type:

- Object

Properties:

Name	Type	Description
type	string	either "content" for page and stream structure elements or "object" for object references.
id	string	unique id that will map to the text layer.

Source: display/api.js, line 1176

StructTreeNode

Structure tree node. The root node will have a role "Root".

Type:

- Object

Properties:

Name	Type	Description
children	Array.<(StructTreeNode|StructTreeContent)>	Array of StructTreeNode and StructTreeContent objects.
role	string	element's role, already mapped if a role map exists in the PDF.

Source: display/api.js, line 1166

TextContent

Page text content.

Type:

- Object

Properties:

Name	Type	Description
items	Array.<(TextItem|TextMarkedContent)>	Array of TextItem and TextMarkedContent objects. TextMarkedContent items are included when includeMarkedContent is true.
styles	Object.<string, TextStyle>	TextStyle objects, indexed by font name.

Source: display/api.js, line 1079

TextItem

Page text content part.

Type:

- Object

Properties:

Name	Type	Description
str	string	Text content.
dir	string	Text direction: 'ttb', 'ltr' or 'rtl'.
transform	Array.<any>	Transformation matrix.
width	number	Width in device space.
height	number	Height in device space.
fontName	string	Font name used by PDF.js for converted font.
hasEOL	boolean	Indicating if the text content is followed by a line-break.

Source: display/api.js, line 1090

TextMarkedContent

Page text marked content part.

Type:

- Object

Properties:

Name	Type	Description
type	string	Either 'beginMarkedContent', 'beginMarkedContentProps', or 'endMarkedContent'.
id	string	The marked content identifier. Only used for type 'beginMarkedContentProps'.

Source: display/api.js, line 1104

TextStyle

Text style.

Type:

- Object

Properties:

Name	Type	Description
ascent	number	Font ascent.
descent	number	Font descent.
vertical	boolean	Whether or not the text is in vertical mode.
fontFamily	string	The possible font family.

Source: display/api.js, line 1114

TypedArray

Type:

- Int8Array | Uint8Array | Uint8ClampedArray | Int16Array | Uint16Array | Int32Array | Uint32Array | Float32Array | Float64Array

Source: display/api.js, line 109
