

---

# **HED JavaScript**

***Release 3.10.0***

**HED Working Group**

**Oct 06, 2023**



**CONTENTS:**

<b>1</b>	<b>bids</b>	<b>3</b>
1.1	Schema . . . . .	3
1.2	tsvParse . . . . .	3
1.3	types . . . . .	3
<b>2</b>	<b>validator</b>	<b>7</b>
2.1	event . . . . .	7
2.1.1	hed3 . . . . .	7
2.1.2	validator . . . . .	9
<b>3</b>	<b>Indices and tables</b>	<b>11</b>
	<b>Index</b>	<b>13</b>





\_static/images/croppedWideLogo.png

### Links

- [PDF docs](#)
- [Source code](#)

Note: this is a work in progress. More information is coming.



Bids functionality

## 1.1 Schema

Intentionally Blank

## 1.2 tsvParse

**parseTSV**(*contents*)

Parse a TSV file.

**Arguments**

- **contents** (*string*) – The contents of a TSV file.

**Returns**

**Object** – The parsed contents of the TSV file.

## 1.3 types

**class BidsData**()

---

**Note:** Deprecated: Will be removed in v4.0.0.

---

Base class for BIDS data.

**BidsData.definitions**

---

**Note:** Deprecated: Will be removed in v4.0.0.

---

**type:** Map.<string, ParsedHedGroup>

A Mapping from definition names to their associated ParsedHedGroup objects.

**BidsData.hedIssues**

---

**Note:** Deprecated: Will be removed in v4.0.0.

---

**type:** Array.<Issue>

A list of HED validation issues. This will be converted to BidsIssue objects later on.

**BidsData.parsedStringMapping**

---

**Note:** Deprecated: Will be removed in v4.0.0.

---

**type:** Map.<string, ParsedHedString>

A mapping from unparsed HED strings to ParsedHedString objects.

**class BidsFile**(*name*, *file*)

A BIDS file.

**BidsFile.file**

**type:** object

The file object representing this file data. This is used to generate BidsIssue objects.

**BidsFile.name**

**type:** string

The name of this file.

**class BidsJsonFile**()

A BIDS JSON file.

**BidsJsonFile.BidsJsonFile****BidsJsonFile.jsonData**

**type:** object

This file's JSON data.

**class BidsTsvFile**(*name*, *tsvData*, *file*, *potentialSidecars*, *mergedDictionary*)

A BIDS TSV file.

Constructor.

**Arguments**

- **name** (*string*) – The name of the TSV file.
- **tsvData** (*Object/string*) – This file's TSV data.
- **file** (*object*) – The file object representing this file.
- **potentialSidecars** (*Array.<string>*) – The list of potential JSON sidecars.
- **mergedDictionary** (*object*) – The merged sidecar data.

**BidsTsvFile.BidsTsvFile**

Constructor.



**BidsTsvFile.hedColumnHedStrings**

**type:** Array.<string>

HED strings in the “HED” column of the TSV data.

**BidsTsvFile.mergedSidecar**

**type:** BidsSidecar

The pseudo-sidecar object representing the merged sidecar data.

**BidsTsvFile.parsedTsv**

**type:** Object

This file’s parsed TSV data.

**BidsTsvFile.potentialSidecars**

**type:** Array.<string>

The list of potential JSON sidecars.

**BidsTsvFile.sidecarHedData**

**type:** Map.<string, (string|Object.<string, string>)>

The extracted HED data for the merged pseudo-sidecar.

**class BidsEventFile**(*name*, *potentialSidecars*, *mergedDictionary*, *tsvData*, *file*)

A BIDS events.tsv file.

Constructor.

#### Arguments

- **name** (*string*) – The name of the event TSV file.
- **potentialSidecars** (*Array.<string>*) – The list of potential JSON sidecars.
- **mergedDictionary** (*object*) – The merged sidecar data.
- **tsvData** (*Object|string*) – This file’s TSV data.
- **file** (*object*) – The file object representing this file.

**BidsEventFile.BidsEventFile**

Constructor.

**class BidsTabularFile**(*name*, *potentialSidecars*, *mergedDictionary*, *tsvData*, *file*)

A BIDS TSV file other than an events.tsv file.

Constructor.

#### Arguments

- **name** (*string*) – The name of the TSV file.
- **potentialSidecars** (*Array.<string>*) – The list of potential JSON sidecars.
- **mergedDictionary** (*object*) – The merged sidecar data.
- **tsvData** (*Object|string*) – This file’s TSV data.
- **file** (*object*) – The file object representing this file.

**BidsTabularFile.BidsTabularFile**

Constructor.

**class** BidsSidecar(*name*, *sidecarData*, *file*)

Constructor.

**Arguments**

- **name** (*string*) – The name of the sidecar file.
- **sidecarData** (*Object*) – The raw JSON data.
- **file** (*Object*) – The file object representing this file.

## VALIDATOR

### 2.1 event

#### 2.1.1 hed3

**class Hed3Validator**(*parsedString, hedSchemas, definitions, options*)

Hed3Validator class

Constructor.

##### Arguments

- **parsedString** (*ParsedHedString*) – The parsed HED string to be validated.
- **hedSchemas** (*Schemas*) – The collection of HED schemas.
- **definitions** (*Map.<string, ParsedHedGroup>*) – The parsed definitions.
- **options** (*Object.<string, boolean>*) – The validation options.

**Hed3Validator.Hed3Validator**

Constructor.

**Hed3Validator.definitions**

**type:** *Map.<string, ParsedHedGroup>*

The parsed definitions.

**Hed3Validator.checkDefinitionGroupSyntax**(*tagGroup*)

Check the syntax of HED 3 definitions.

##### Arguments

- **tagGroup** (*ParsedHedGroup*) – The tag group.

**Hed3Validator.checkDefinitionStringSyntax**()

Check full-string Definition syntax.

**Hed3Validator.checkForInvalidTopLevelTagGroupTags**()

Check for tags marked with the `topLevelTagGroup` attribute that are not in top-level tag groups.

**Hed3Validator.checkForInvalidTopLevelTags**()

Check for invalid top-level tags.

`Hed3Validator.checkForMissingDefinitions(tag, defShortTag="Def")`

Check for missing HED 3 definitions.

**Arguments**

- **tag** (*ParsedHedTag*) – The HED tag.
- **defShortTag** (*string*) – The short tag to check for.

`Hed3Validator.checkIfTagIsValid(tag, previousTag)`

Check if an individual HED tag is in the schema or is an allowed extension.

`Hed3Validator.checkIfTagUnitClassUnitsAreValid(tag)`

Check that the unit is valid for the tag's unit class.

**Arguments**

- **tag** (*ParsedHed3Tag*) – A HED tag.

`Hed3Validator.checkPlaceholderStringSyntax()`

Check full-string placeholder syntax.

`Hed3Validator.checkPlaceholderTagSyntax(tag)`

Check basic placeholder tag syntax.

**Arguments**

- **tag** (*ParsedHedTag*) – A HED tag.

`Hed3Validator.checkTemporalSyntax(tagGroup)`

Check the syntax of HED 3 onsets and offsets.

**Arguments**

- **tagGroup** (*ParsedHedGroup*) – The tag group.

`Hed3Validator.checkValueTagSyntax(tag)`

Check the syntax of tag values.

**Arguments**

- **tag** (*ParsedHed3Tag*) – A HED tag.

`Hed3Validator.validateFullParsedHedString()`

Validate the full parsed HED string.

`Hed3Validator.validateHedTagGroup(parsedTagGroup)`

Validate a HED tag group.

`Hed3Validator.validateIndividualHedTag(tag, previousTag)`

Validate an individual HED tag.

`Hed3Validator.validateTopLevelTagGroups()`

Validate the top-level HED tag groups in a parsed HED string.

`Hed3Validator.validateTopLevelTags()`

Validate the top-level HED tags in a parsed HED string.

`Hed3Validator.validateUnits(tag)`

Validate a unit and strip it from the value.

**Arguments**

- **tag** (*ParsedHed3Tag*) – A HED tag.

#### Returns

`Hed3Validator.validateValue(value, isNumeric)`

Determine if a stripped value is valid.

#### Arguments

- **value** (*string*) – The stripped value.
- **isNumeric** (*boolean*) – Whether the tag is numeric.

## 2.1.2 validator

**class HedValidator**(*parsedString, hedSchemas, options*)

HedValidator class

Constructor.

#### Arguments

- **parsedString** (*ParsedHedString*) – The parsed HED string to be validated.
- **hedSchemas** (*Schemas*) – The collection of HED schemas.
- **options** (*Object.<String, boolean>*) – The validation options.

`HedValidator.HedValidator`

Constructor.

`HedValidator.hedSchemas`

**type:** Schemas

The collection of HED schemas.

`HedValidator.issues`

**type:** Array.<Issue>

The running issue list.

`HedValidator.options`

**type:** Object.<string, boolean>

The validation options.

`HedValidator.parsedString`

**type:** ParsedHedString

The parsed HED string to be validated.

`HedValidator._checkForTagAttribute(attribute, fn)`

Validation check based on a tag attribute.

#### Arguments

- **attribute** (*string*) – The name of the attribute.
- **fn** (*function*) – The actual validation code.

`HedValidator.checkForDuplicateTags(tagList)`

Check for duplicate tags at the top level or within a single group.

`HedValidator.checkForMultipleUniqueTags(tagList)`

Check for multiple instances of a unique tag.

`HedValidator.checkForRequiredTags()`

Check that all required tags are present.

`HedValidator.checkIfTagIsValid(tag, previousTag)`

Check if an individual HED tag is in the schema or is an allowed extension.

`HedValidator.checkIfTagRequiresChild(tag)`

Check if a tag is missing a required child.

#### Arguments

- **tag** (*ParsedHedTag*) – The HED tag to be checked.

`HedValidator.checkIfTagUnitClassUnitsAreValid(tag)`

Check that the unit is valid for the tag's unit class.

#### Arguments

- **tag** (*ParsedHedTag*) – A HED tag.

`HedValidator.checkValueTagSyntax(tag)`

Check the syntax of tag values.

#### Arguments

- **tag** (*ParsedHedTag*) – A HED tag.

`HedValidator.pushIssue(internalCode, parameters)`

Generate a new issue object and push it to the end of the issues array.

#### Arguments

- **internalCode** (*string*) – The internal error code.
- **parameters** (*Object.<string, (string|Array.<number>)>*) – The error string parameters.

`HedValidator.validateHedTagGroup(parsedTagGroup)`

Validate a HED tag group.

`HedValidator.validateHedTagGroups()`

Validate the HED tag groups in a parsed HED string.

`HedValidator.validateHedTagLevel(tagList)`

Validate a HED tag level.

`HedValidator.validateHedTagLevels()`

Validate the HED tag levels in a parsed HED string object.

`HedValidator.validateIndividualHedTag(tag, previousTag)`

Validate an individual HED tag.

`HedValidator.validateIndividualHedTags()`

Validate the individual HED tags in a parsed HED string object.

`HedValidator.validateTopLevelTags()`

Validate the top-level HED tags in a parsed HED string.

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`





## INDEX

### B

BidsData() (class), 3  
 BidsData.definitions (BidsData attribute), 3  
 BidsData.hedIssues (BidsData attribute), 3  
 BidsData.parsedStringMapping (BidsData attribute), 4  
 BidsEventFile() (class), 5  
 BidsEventFile.BidsEventFile (BidsEventFile attribute), 5  
 BidsFile() (class), 4  
 BidsFile.file (BidsFile attribute), 4  
 BidsFile.name (BidsFile attribute), 4  
 BidsJsonFile() (class), 4  
 BidsJsonFile.BidsJsonFile (BidsJsonFile attribute), 4  
 BidsJsonFile.jsonData (BidsJsonFile attribute), 4  
 BidsSidecar() (class), 5  
 BidsTabularFile() (class), 5  
 BidsTabularFile.BidsTabularFile (BidsTabularFile attribute), 5  
 BidsTsvFile() (class), 4  
 BidsTsvFile.BidsTsvFile (BidsTsvFile attribute), 4  
 BidsTsvFile.hedColumnHedStrings (BidsTsvFile attribute), 4  
 BidsTsvFile.mergedSidecar (BidsTsvFile attribute), 5  
 BidsTsvFile.parsedTsv (BidsTsvFile attribute), 5  
 BidsTsvFile.potentialSidecars (BidsTsvFile attribute), 5  
 BidsTsvFile.sidecarHedData (BidsTsvFile attribute), 5

### H

Hed3Validator() (class), 7  
 Hed3Validator.checkDefinitionGroupSyntax() (Hed3Validator method), 7  
 Hed3Validator.checkDefinitionStringSyntax() (Hed3Validator method), 7  
 Hed3Validator.checkForInvalidTopLevelTagGroupTags() (Hed3Validator method), 7  
 Hed3Validator.checkForInvalidTopLevelTags() (Hed3Validator method), 7

Hed3Validator.checkForMissingDefinitions() (Hed3Validator method), 7  
 Hed3Validator.checkIfTagIsValid() (Hed3Validator method), 8  
 Hed3Validator.checkIfTagUnitClassUnitsAreValid() (Hed3Validator method), 8  
 Hed3Validator.checkPlaceholderStringSyntax() (Hed3Validator method), 8  
 Hed3Validator.checkPlaceholderTagSyntax() (Hed3Validator method), 8  
 Hed3Validator.checkTemporalSyntax() (Hed3Validator method), 8  
 Hed3Validator.checkValueTagSyntax() (Hed3Validator method), 8  
 Hed3Validator.definitions (Hed3Validator attribute), 7  
 Hed3Validator.Hed3Validator (Hed3Validator attribute), 7  
 Hed3Validator.validateFullParsedHedString() (Hed3Validator method), 8  
 Hed3Validator.validateHedTagGroup() (Hed3Validator method), 8  
 Hed3Validator.validateIndividualHedTag() (Hed3Validator method), 8  
 Hed3Validator.validateTopLevelTagGroups() (Hed3Validator method), 8  
 Hed3Validator.validateTopLevelTags() (Hed3Validator method), 8  
 Hed3Validator.validateUnits() (Hed3Validator method), 8  
 Hed3Validator.validateValue() (Hed3Validator method), 9  
 HedValidator() (class), 9  
 HedValidator.\_checkForTagAttribute() (HedValidator method), 9  
 HedValidator.checkForDuplicateTags() (HedValidator method), 9  
 HedValidator.checkForMultipleUniqueTags() (HedValidator method), 9  
 HedValidator.checkForRequiredTags() (HedValidator method), 10  
 HedValidator.checkIfTagIsValid() (HedValidator

*method*), 10  
HedValidator.checkIfTagRequiresChild() (*HedValidator method*), 10  
HedValidator.checkIfTagUnitClassUnitsAreValid() (*HedValidator method*), 10  
HedValidator.checkValueTagSyntax() (*HedValidator method*), 10  
HedValidator.hedSchemas (*HedValidator attribute*), 9  
HedValidator.HedValidator (*HedValidator attribute*), 9  
HedValidator.issues (*HedValidator attribute*), 9  
HedValidator.options (*HedValidator attribute*), 9  
HedValidator.parsedString (*HedValidator attribute*), 9  
HedValidator.pushIssue() (*HedValidator method*), 10  
HedValidator.validateHedTagGroup() (*HedValidator method*), 10  
HedValidator.validateHedTagGroups() (*HedValidator method*), 10  
HedValidator.validateHedTagLevel() (*HedValidator method*), 10  
HedValidator.validateHedTagLevels() (*HedValidator method*), 10  
HedValidator.validateIndividualHedTag() (*HedValidator method*), 10  
HedValidator.validateIndividualHedTags() (*HedValidator method*), 10  
HedValidator.validateTopLevelTags() (*HedValidator method*), 10

## P

parseTSV() (*built-in function*), 3