JSLogger

Apr 03, 2024 23:25

Supported Clients

SmartClient WebClient NGClient

Property Summary

JSLogBuilder	always	Construct a log event that will always be logged.
JSLogBuilder	debug	Construct a debug log event.
Boolean	debugEnabled	Check if the current logger's logging level enables logging on the debug level.
JSLogBuilder	error	Construct an error log event.
Boolean	errorEnabled	Check if the current logger's logging level enables logging on the error level.
JSLogBuilder	fatal	Construct a fatal log event.
Boolean	fatalEnabled	Check if the current logger's logging level enables logging on the fatal level.
JSLogBuilder	info	Construct an info log event.
Boolean	infoEnabled	Check if the current logger's logging level enables logging on the info level.
String	level	Get the logging level of this logger
JSLogBuilder	trace	Construct a trace log event.
Boolean	traceEnabled	Check if the current logger's logging level enables logging on the trace level.
JSLogBuilder	warn	Construct a warn log event.
Boolean	warnEnabled	Check if the current logger's logging level enables logging on the warn level.

Methods Summary

void setLevel(level) Set the level for this logger.

Property Details

always

```
Construct a log event that will always be logged.
```

Returns

JSLogBuilder a LogBuilder

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var log = application.getLogger();
log.always.log("some message and {} {}", "some", "arguments");
```

debug

```
Construct a debug log event.
```

Returns

JSLogBuilder a LogBuilder

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var log = application.getLogger();
log.debug.log("some message and {} {}", "some", "arguments");
```

debugEnabled

```
Check if the current logger's logging level enables logging on the debug level.
Return true if the logger's level is set to debug or trace.
```

Returns

Boolean true if 'debug' level is enabled for logging

Supported Clients

SmartClient,WebClient,NGClient

Sample

error

Construct an error log event.

Returns

JSLogBuilder a LogBuilder

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var log = application.getLogger();
log.error.log("some message and {} {}", "some", "arguments");
```

errorEnabled

Check if the current logger's logging level enables logging on the error level.
Return true if the logger's level is set to error, warn, info, debug or trace.

Returns

Boolean true if 'error' level is enabled for logging

Supported Clients

SmartClient,WebClient,NGClient

Sample

fatal

Construct a fatal log event.

Returns

JSLogBuilder a LogBuilder

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var log = application.getLogger();
log.fatal.log("some message and {} {}", "some", "arguments");
```

fatalEnabled

Check if the current logger's logging level enables logging on the fatal level.
Return true if the logger's level is set to fatal, error, warn, info, debug or trace.

Returns

Boolean true if 'fatal' level is enabled for logging

Supported Clients

SmartClient,WebClient,NGClient

Sample

info

Construct an info log event.

Returns

JSLogBuilder a LogBuilder

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var log = application.getLogger();
log.info.log("some message and {} {}", "some", "arguments");
```

infoEnabled

Check if the current logger's logging level enables logging on the info level.
Return true if the logger's level is set to info, debug or trace.

Returns

Boolean true if 'info' level is enabled for logging

Supported Clients

SmartClient,WebClient,NGClient

Sample

level

Get the logging level of this logger

Returns

String the logging level of this logger

Supported Clients

SmartClient,WebClient,NGClient

Sample

trace

Construct a trace log event.

Returns

JSLogBuilder a LogBuilder

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var log = application.getLogger();
log.trace.log("some message and {} {}", "some", "arguments");
```

traceEnabled

Check if the current logger's logging level enables logging on the trace level.
Return true if the logger's level is set to trace.

Returns

Boolean true if 'trace' level is enabled for logging

Supported Clients

SmartClient,WebClient,NGClient

Sample

warn

Construct a warn log event.

Returns

JSLogBuilder a LogBuilder

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var log = application.getLogger();
log.warn.log("some message and {} {}", "some", "arguments");
```

warnEnabled

Check if the current logger's logging level enables logging on the warn level.
Return true if the logger's level is set to warn, info, debug or trace.

Returns

Boolean true if 'warn' level is enabled for logging

Supported Clients

SmartClient,WebClient,NGClient

Sample

Methods Details

setLevel(level)

Set the level for this logger.
Be aware that this will override the logging level as configured in log4j.xml,
meaning it affects all JSLogger instances based on that configuration.
This changes the global configuration,
meaning that restarting the client will not reset the logging level to it's default state.
Only restarting the application server will reset the logging level to it's default state.

Parameters

JSLogBuilder level the desired logging level for this logger

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var log = application.getLogger("myLogger");
log.setLevel(log.info);
```