Nets DanID A/S
Lautrupbjerg 10
DK – 2750 Ballerup

T +45 87 42 45 00
F +45 70 20 66 29
info@danid.dk
www.nets-danid.dk

CVR no. 30808460

NemID JS Sign text validation tool

Table of Contents

History

2014-01-31	Version 1.0	HASOR
2014-05-28	Version 1.1	PHJER

1. The Purpose and Target Group of the Document

This document is part of the Service Provider Package for NemID.

The purpose of this document is to provide an overview of the functionality provided by the sign text validation tool.

The document is aimed at the people at the service provider who are responsible for the implementation of NemID.

2. Introduction

This document provides an overview of the functionality provided by the sign text validation tool, which is part of the release of the NemID JavaScript client.

The aim of the sign text validation tool is to provide service providers with a way to ensure that HTML or PDF documents adhere to the security requirements of the NemID client. The tool uses the same validation engines that are used by the client, and thus makes it possible to validate documents without launching the client.

The tool can be used in two ways: From the command line or as a library.

Please note that this tool does not provide functionality for seeing how the document will look in the client. The "SignText viewer" is available in the developer tools[1] for this purpose.

[1] https://appletk.danid.dk/developers/

3. Command line usage

The validation tool can be used from the command line like in the following example:

```
java -jar signvalidator.jar aftale.pdf
```

The tool will attempt to guess the format of the input-file from its filename. It can also be specified as in the following example :

```
java -jar signvalidator.jar -HTML input-file
```

Invoking the tool without parameters will display a short help text. The tool will output either "Validation successful" or "Validation failed" followed by a list of errors.

4. Library usage

It is possible to integrate the tool in an application or an automated workflow by using its exposed validation API.

Two validation methods are available:

```
dk.danid.javascript.signtextvalidator.HTML.validate(String)

dk.danid.javascript.signtextvalidator.HTML.validate(File)



dk.danid.javascript.signtextvalidator.PDF.validate(byte[])

dk.danid.javascript.signtextvalidator.PDF.validate(File)
```

The methods are overloaded to provide a way to validate either a file or an in-memory document, and will return null when the file is successfully validated. In case of validation failues, the return value is a description of the rule that was infringed upon.