

Package ‘danlex’

September 11, 2026

Title Access Danish Legislation from Retsinformation

Version 0.1.0

Description Provides access to Danish acts, regulations and treaties published by Retsinformation via its European Legislation Identifier (ELI) service. Part of the lexverse family of packages for legal and regulatory data.

License MIT + file LICENSE

Depends R (>= 4.1.0)

URL <https://github.com/KristianVepsalainen/danlex>

BugReports <https://github.com/KristianVepsalainen/danlex/issues>

Encoding UTF-8

Imports htr2, rlang, tibble, xml2

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Language en-US

NeedsCompilation no

Author Kristian Vepsäläinen [aut, cre]

Maintainer Kristian Vepsäläinen <kristian.vepsalainen@proton.me>

Repository CRAN

Date/Publication 2026-09-11 12:00:15 UTC

Contents

dlx_clear_index	2
dlx_get_changes	2
dlx_get_doc	4
dlx_get_docs	6

dlx_get_paragraphs	7
dlx_get_references	9
dlx_get_text	11
dlx_index_status	12
dlx_list_documents	13
Index	15

dlx_clear_index	<i>Discard the session's corpus index</i>
-----------------	---

Description

Frees the memory held by [dlx_list_documents\(\)](#). The next call will retrieve the index again.

Usage

```
dlx_clear_index()
```

Value

Invisibly, TRUE if an index was discarded and FALSE if none was held.

See Also

```
dlx\_list\_documents\(\)
```

Examples

```
dlx_clear_index()
```

dlx_get_changes	<i>Retrieve recent changes to Danish legislation</i>
-----------------	--

Description

Reads the Retsinformation ELI update feed and returns the documents published or amended within its retention window, one row per change.

Usage

```
dlx_get_changes(since = NULL, collection = NULL, reason = NULL)
```

Arguments

since	Optional date (or anything <code>as.Date()</code> accepts). Only changes on or after this date are returned.
collection	Optional character vector of collection codes, e.g. "lta" or <code>c("lta", "ltc")</code> .
reason	Optional character vector of change reasons. Recognised values are "NewDocument", "DocumentContentChanged", "DocumentMetadataChanged", "DocumentMetadataChangedAndDocumentRemovedDocument" and "Unknown".

Value

A `tibble` with one row per change:

`eli` Canonical ELI URI. Pass to `dlx_get_doc()` via `eli`.

`collection, year, number` Derived from the ELI URI.

`title` Document title.

`updated` Date the entry was last published or updated.

`change_date` Date of the change itself, from the source's `changeDate` element.

`reason_for_change` Why the entry appears in the feed.

`published_in` Publication channel.

`n_images` Number of images attached to the document.

Zero rows if no change matches the filters.

Retention window

The feed holds at least 60 days of history and is refreshed daily. It is a static file with no server-side query interface, so `since`, `collection` and `reason` filter the retrieved feed rather than the request. Asking for a `since` date earlier than the feed reaches will therefore return a truncated result, and a warning is issued when that happens.

There is no way to obtain older changes from this API. Use `dlx_list_documents()` and the sitemap's modification dates for anything beyond the window.

Identifying the collection

The feed itself does not record which collection a document belongs to; it reports only `published_in`, the publication channel. `danlex` derives `collection`, `year` and `number` from the ELI identifier, which makes it possible to separate legislation (`lta`, `ltb`, `ltc`, `mt`) from administrative decisions (`retsinfo`) and parliamentary material (`ft`).

The two are related but not equivalent: `published_in` distinguishes Lovtidende A, B and C from Retsinformation's own series, whereas `collection` follows the ELI URI structure.

Source

Retsinformation ELI update feed, <https://www.retsinformation.dk/eli/eli-update-feed.atom>.

See Also

[dlx_get_doc\(\)](#) to retrieve any of the changed documents.

Examples

```
if (FALSE) {
  # Everything the feed holds
  dlx_get_changes()

  # New acts and regulations only, from the past fortnight
  dlx_get_changes(
    since      = Sys.Date() - 14,
    collection = "lta",
    reason     = "NewDocument"
  )

  # Fetch the first changed document in full
  ch <- dlx_get_changes(collection = "lta")
  dlx_get_doc(eli = ch$eli[1])
}
```

dlx_get_doc

Retrieve a Danish legal document

Description

Fetches a single document from Retsinformation and returns its metadata as a one-row tibble. The document may be identified in any one of three ways: by the deterministic triple of collection, year and number; by accession number; or by a full ELI URI.

Usage

```
dlx_get_doc(
  collection = NULL,
  year       = NULL,
  number     = NULL,
  accn       = NULL,
  eli        = NULL
)
```

Arguments

collection	Collection segment of the ELI URI. One of "lta" (Lovtidende A: acts and regulations), "ltb", "ltc" (treaties) or "mt" (Ministerialtidende, discontinued in 2012). The collections "retsinfo", "fob" and "ft" are recognised but not yet supported.
year	Four-digit year.

number	Document number <i>as published</i> . Character or numeric; stored as character, because a small number of documents use non-numeric forms such as "1-1".
accn	Accession number, e.g. "A19990059229".
eli	A full canonical ELI URI.

Value

A one-row [tibble](#) with, among others:

status "ok", "metadata_only" or "not_found".

eli Canonical ELI URI.

collection, year, number Identifier components.

accn Accession number.

title Document title.

document_type Type as recorded, verbatim.

document_type_code Internal code, post-2007 records only.

document_type_std Harmonised type label where known.

legal_status The source's Status field. Named legal_status to avoid collision with status above.

dies_signi, dies_edicti Signing and publication dates.

start_date, end_date Period of validity.

ministry, administrative_authority Issuing body.

has_text Whether full content is available.

n_references Number of references to other instruments.

A document that does not exist yields a one-row tibble with status = "not_found" and all other fields missing, rather than an error.

Identifier forms

The deterministic form is the one users will normally have. A Danish act or regulation is cited as, for example, "bekendtgørelse nr. 763 af 1998", which maps directly onto `dlx_get_doc("lta", 1998, 763)`.

The accession form exists to follow references. `dlx_get_references()` returns accession numbers rather than ELI URIs, so the citation network can be traversed without a lookup table.

The ELI form is for URIs taken from `dlx_list_documents()` or `dlx_get_changes()`.

Text availability

Retsinformation's ELI service returns full document content only for material published from roughly late September 2007 onward. The boundary observed in Lovtidende A is act number 1081 of 2007, announced 25 September 2007. Earlier documents return metadata only, and status is then "metadata_only" rather than "ok".

This is a property of the source, not of danlex. Metadata for earlier documents is complete and reaches back to 1852 for Lovtidende A.

Instrument types

Lovtidende A carries both acts (*love*) and regulations (*bekendtgørelser*) in a single number space, so the instrument type cannot be inferred from the identifier. Always consult `document_type`.

Records also use two different schemas either side of the 2007 boundary: older records give a whole Danish word, newer ones a composite code. Both are returned verbatim in `document_type`, with the internal code split out into `document_type_code` and a harmonised label in `document_type_std`.

Source

Retsinformation, <https://www.retsinformation.dk>. Terms of use: <https://www.retsinformation.dk/api>.

Examples

```
if (FALSE) {
  # A regulation from 1998 – metadata only, as it predates the text boundary
  dlx_get_doc("lta", 1998, 763)

  # A modern act – full content available
  dlx_get_doc("lta", 2025, 1)

  # Following a reference by accession number
  dlx_get_doc(accn = "A19990059229")

  # Straight from an ELI URI
  dlx_get_doc(eli = "https://retsinformation.dk/eli/lta/1998/763")
}
```

dlx_get_docs	<i>Retrieve many documents at once</i>
--------------	--

Description

Fetches metadata for a vector of documents and returns one row per identifier, in the order given. Intended for corpus-scale work: the output of `dlx_list_documents()` or the `ref_accn` column of `dlx_get_references()` can be passed straight in.

Usage

```
dlx_get_docs(eli = NULL, accn = NULL, delay = 0.2, progress = interactive())
```

Arguments

<code>eli</code>	Character vector of canonical ELI URIs.
<code>accn</code>	Character vector of accession numbers. Supply this or <code>eli</code> , not both.
<code>delay</code>	Seconds to pause between requests.
<code>progress</code>	Show progress. Defaults to <code>interactive()</code> .

Value

A [tibble](#) with the columns of [dlx_get_doc\(\)](#), plus `error_message`, which is NA except on failed rows. One row per input identifier, in input order. Duplicated identifiers are fetched once and repeated in the output.

Failures do not stop the run

A batch of any size will eventually meet a network hiccup, and losing an hour of retrieval to one dropped connection would be unacceptable. Failed documents are therefore returned as rows with `status = "error"` and the reason in `error_message`, rather than raised as conditions. Inspect them with `subset(out, status == "error")` and retry those identifiers.

Missing documents behave as in [dlx_get_doc\(\)](#): `status` is `"not_found"`, which is a normal answer rather than a failure.

Rate limiting

The ELI endpoints are not rate limited — the one-call-per-ten-seconds restriction applies to the harvest service at `api.retsinformation.dk`, which `danlex` does not use. `delay` nevertheless defaults to a courtesy pause of 0.2 seconds, since a batch of tens of thousands of requests deserves some restraint. Retrieving 1,000 documents takes roughly five minutes at the default.

See Also

[dlx_get_doc\(\)](#) for a single document, [dlx_list_documents\(\)](#) for identifiers to feed in.

Examples

```
if (FALSE) {
  # Every act and regulation from one year
  idx <- dlx_list_documents(year = 2025, collection = "lta")
  docs <- dlx_get_docs(idx$eli)

  # Anything that failed
  subset(docs, status == "error")

  # Follow a whole set of references
  refs <- dlx_get_references("lta", 1998, 763)
  dlx_get_docs(accn = refs$ref_accn)
}
```

<code>dlx_get_paragraphs</code>	<i>Retrieve a document's provisions</i>
---------------------------------	---

Description

Returns one row per *stykke* (subsection), or one row per section where a section has no subsections. This gives provision-level access to Danish legislation, which is not possible from the plain text alone.

Usage

```
dlx_get_paragraphs(
  collection = NULL,
  year = NULL,
  number = NULL,
  accn = NULL,
  eli = NULL
)
```

Arguments

collection	Collection segment of the ELI URI. One of "lta" (Lovtidende A: acts and regulations), "ltb", "ltc" (treaties) or "mt" (Ministerialtidende, discontinued in 2012). The collections "retsinfo", "fob" and "ft" are recognised but not yet supported.
year	Four-digit year.
number	Document number <i>as published</i> . Character or numeric; stored as character, because a small number of documents use non-numeric forms such as "1-1".
accn	Accession number, e.g. "A19990059229".
eli	A full canonical ELI URI.

Value

A [tibble](#) with one row per provision:

eli ELI URI of the document.

paragraf_index Position of the section in document order.

paragraf_id, paragraf_local_id Source identifiers.

paragraf Section marker, e.g. "§ 1".

stk_index Position of the subsection within its section.

stk Subsection marker, e.g. "Stk. 2.", or NA.

is_amendment Whether this provision is new text destined for another instrument.

text Provision text, including any nested list items.

Zero rows for metadata-only or non-existent documents.

Amending provisions

Danish amending instruments (BEK Æ, LOV Æ, ANG) contain Paragraf elements that are **not their own**: they are new text to be inserted into another instrument, nested under AendringNyTekst. An amending regulation may therefore appear to have sections that in fact belong to the act it amends.

The is_amendment column distinguishes the two. In a consolidated act it is FALSE throughout; in an amending instrument it is typically TRUE throughout. Filter on it before treating the output as the document's own structure.

Known limitation

Only Stk elements that sit inside a Paragraf are returned. A small number of amending instruments place Stk directly under AendringNyTekst with no enclosing section; those are currently omitted.

Source

Retsinformation, <https://www.retsinformation.dk>.

See Also

[dlx_get_text\(\)](#) for the whole document as one string.

Examples

```
if (FALSE) {  
  # A consolidated regulation: all provisions are its own  
  p <- dlx_get_paragraphs("lta", 2025, 50)  
  table(p$sis_amendment)  
  
  # An amending regulation: the sections belong to the amended instrument  
  a <- dlx_get_paragraphs("lta", 2025, 1)  
  table(a$sis_amendment)  
}
```

dlx_get_references	<i>Retrieve the references a document makes to other instruments</i>
--------------------	--

Description

Returns the cross-references recorded in a document's metadata as an edge table, one row per reference. These are typically the enabling provisions (*hjemmel*) a regulation rests on.

Usage

```
dlx_get_references(  
  collection = NULL,  
  year = NULL,  
  number = NULL,  
  accn = NULL,  
  eli = NULL  
)
```

Arguments

collection	Collection segment of the ELI URI. One of "lta" (Lovtidende A: acts and regulations), "ltb", "ltc" (treaties) or "mt" (Ministerialtidende, discontinued in 2012). The collections "retsinfo", "fob" and "ft" are recognised but not yet supported.
year	Four-digit year.
number	Document number <i>as published</i> . Character or numeric; stored as character, because a small number of documents use non-numeric forms such as "1-1".
accn	Accession number, e.g. "A19990059229".
eli	A full canonical ELI URI.

Value

A [tibble](#) with one row per reference:

from_eli ELI URI of the referring document.

from_accn Accession number of the referring document.

ref_index Position of the reference within the document.

ref_accn Accession number of the referenced instrument. Pass this to `dlx_get_doc(accn = )` to follow the edge.

ref_date Date of the referenced version.

ref_title Title of the referenced instrument.

Zero rows if the document has no references. A document that does not exist yields zero rows and a warning.

References are maintained, not historical

This is the single most important thing to understand about these edges. Retsinformation keeps references pointing at the *current* consolidated version of the parent instrument, not at the version in force when the referring document was issued.

A regulation from 1998, for instance, carries references to acts dated 1999 — the links were updated when the parent acts were re-consolidated. Anyone building a citation network and treating the edges as timestamped to the source document's date will get a systematically wrong picture of what the law looked like at any past moment.

Use `ref_date` to see which version an edge currently points at. There is no way to recover the reference as it stood at enactment from this API.

Coverage

References are present on both sides of the 2007 text boundary — roughly 43% of older and 59% of modern documents carry at least one. A document with no enabling provision simply has none, which is not an error.

Source

Retsinformation, <https://www.retsinformation.dk>.

See Also

[dlx_get_doc\(\)](#), whose n_references column reports the row count this function would return.

Examples

```
if (FALSE) {
  # A 1998 regulation resting on two acts
  refs <- dlx_get_references("lta", 1998, 763)

  # Follow the first edge
  dlx_get_doc(accn = refs$ref_accn[1])
}
```

dlx_get_text	<i>Retrieve the full text of a document</i>
--------------	---

Description

Returns the document’s operative text as a single string, rendered from the structured XML.

Usage

```
dlx_get_text(
  collection = NULL,
  year = NULL,
  number = NULL,
  accn = NULL,
  eli = NULL
)
```

Arguments

collection	Collection segment of the ELI URI. One of "lta" (Lovtidende A: acts and regulations), "ltb", "ltc" (treaties) or "mt" (Ministerialtidende, discontinued in 2012). The collections "retsinfo", "fob" and "ft" are recognised but not yet supported.
year	Four-digit year.
number	Document number <i>as published</i> . Character or numeric; stored as character, because a small number of documents use non-numeric forms such as "1-1".
accn	Accession number, e.g. "A19990059229".
eli	A full canonical ELI URI.

Value

A one-row [tibble](#) with eli, status ("ok", "metadata_only" or "not_found"), n_char and text.

Text availability

Full content is available only for material published from late September 2007 onward; see `dlx_get_doc()` for the boundary. Earlier documents return `status = "metadata_only"` and `text = NA`.

How the text is rendered

The source encodes text as Linea (lines) containing Char runs, with Explicatus elements carrying structural markers such as "\$ 1" or "Stk. 2.". A naive `xml_text()` on the whole document runs these together without separation, producing strings like "\$ 1I bekendtgørelse nr. ...".

danlex instead walks Explicatus and Linea in document order, joins them with single spaces, and inserts a blank line before each marker that opens a Paragraf or Stk. The result is readable running text with section breaks preserved. Inline formatting (`formaChar`) is discarded.

For structured access to individual provisions, use `dlx_get_paragraphs()`.

Source

Retsinformation, <https://www.retsinformation.dk>.

See Also

`dlx_get_paragraphs()` for provision-level output.

Examples

```
if (FALSE) {
  txt <- dlx_get_text("lta", 2025, 50)
  cat(substr(txt$text, 1, 500))

  # Pre-2008 documents carry metadata only
  dlx_get_text("lta", 1998, 763)$status
}
```

dlx_index_status

Report on the session's corpus index

Description

Report on the session's corpus index

Usage

```
dlx_index_status()
```

Value

A one-row [tibble](#) with `loaded`, `n_documents`, `n_collections`, `retrieved_at` and `size_mb`. When no index has been retrieved, `loaded` is `FALSE` and the rest are missing.

See Also

[dlx_list_documents\(\)](#)

Examples

```
dlx_index_status()
```

dlx_list_documents	<i>List the documents in the Retsinformation corpus</i>
--------------------	---

Description

Enumerates every document published through the ELI service, one row per document, with the collection, year, number and last modification date.

Usage

```
dlx_list_documents(
  year = NULL,
  collection = NULL,
  refresh = FALSE,
  progress = interactive()
)
```

Arguments

year	Optional year or vector of years.
collection	Optional character vector of collection codes. See dlx_get_doc() for what each covers.
refresh	Re-fetch the index even if this session already holds one.
progress	Show a progress bar during retrieval. Defaults to interactive() .

Value

A [tibble](#) with one row per document:

eli Canonical ELI URI. Pass to [dlx_get_doc\(\)](#) via eli.

collection Collection code.

year Year, or NA for ft, which uses parliamentary session identifiers instead.

number Document number, as character.

lastmod Date the record was last modified.

The first call is slow

The corpus index is published as a sitemap split across 21 pages of up to 10,000 entries each, and the pages are not ordered in any way that permits fetching a single year. The whole index must therefore be retrieved, which takes roughly 40 seconds at a polite request rate and around 50 MB of memory.

The result is held for the remainder of the session, so subsequent calls return immediately. `refresh = TRUE` forces a re-fetch, and `dlx_clear_index()` discards it.

danlex deliberately does not write the index to disk. To keep it between sessions, save it yourself:

```
idx <- dlx_list_documents()
saveRDS(idx, "danlex-index.rds")
```

What the index does and does not contain

The sitemap carries only identifiers and modification dates. Titles, document types and dates of enactment are not present; retrieve those with `dlx_get_doc()` for the documents you actually need.

`lastmod` records when Retsinformation last modified its record, which is unrelated to when the instrument was enacted, amended or repealed. A 1950s act may carry a recent `lastmod` simply because its record was revised.

The sitemap is refreshed monthly, so documents published in the last few weeks may be missing. Use `dlx_get_changes()` for recent material.

See Also

`dlx_index_status()`, `dlx_clear_index()`, `dlx_get_changes()`

Examples

```
if (FALSE) {
  # Whole corpus (slow on first call)
  idx <- dlx_list_documents()

  # Acts and regulations from a single year
  dlx_list_documents(year = 2025, collection = "lta")

  # How many documents per collection
  table(idx$collection)
}
```

Index

`as.Date()`, [3](#)

`dlx_clear_index`, [2](#)

`dlx_clear_index()`, [14](#)

`dlx_get_changes`, [2](#)

`dlx_get_changes()`, [14](#)

`dlx_get_doc`, [4](#)

`dlx_get_doc()`, [3](#), [4](#), [7](#), [11–14](#)

`dlx_get_docs`, [6](#)

`dlx_get_paragraphs`, [7](#)

`dlx_get_paragraphs()`, [12](#)

`dlx_get_references`, [9](#)

`dlx_get_references()`, [6](#)

`dlx_get_text`, [11](#)

`dlx_get_text()`, [9](#)

`dlx_index_status`, [12](#)

`dlx_index_status()`, [14](#)

`dlx_list_documents`, [13](#)

`dlx_list_documents()`, [2](#), [6](#), [7](#), [13](#)

`interactive()`, [6](#), [13](#)

`tibble`, [3](#), [5](#), [7](#), [8](#), [10–13](#)