

# Package: washex (via r-universe)

August 20, 2024

**Title** Washington State Legislative Explorer

**Version** 1.2.0

**Description** Gets data from the Washington State Legislature.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

**Depends** R (>= 2.10)

**Imports** dplyr, purrr, rlang, curl, httr, xml2, XML (>= 3.99-0.5)

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** <https://github.com/rwrandles/washex-r>

**BugReports** <https://github.com/rwrandles/washex-r/issues>

**Repository** <https://rwrandles.r-universe.dev>

**RemoteUrl** <https://github.com/rwrandles/washex-r>

**RemoteRef** HEAD

**RemoteSha** ea4c51c096f96838c13380608283036b464dab13

## Contents

getAffectedRCW . . . . .	2
getAmendments . . . . .	3
getBillSponsors . . . . .	4
getCommitteeMembers . . . . .	4
getCommittees . . . . .	5
getCurrentStatus . . . . .	6
getHearings . . . . .	7
getLegislation . . . . .	8

getLegislationByYear . . . . .	9
getLegislationSigned . . . . .	9
getRCWBills . . . . .	10
getRollCalls . . . . .	11
getSponsors . . . . .	12
getStatusChanges . . . . .	13

<b>Index</b>	<b>14</b>
--------------	-----------

---

getAffectedRCW	<i>Link bills to Revised Code of Washington (RCW)</i>
----------------	---

---

### Description

Get a listing of all RCW citations affected by a given bill

### Usage

```
getAffectedRCW(biennium, billId, paired = TRUE, type = c("df", "list", "xml"))
```

### Arguments

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
billId	Character vector containing the bill(s) to be retrieved. Each argument should take the form "XX YYYY", where XX is the prefix (HB, SB, etc.) and YYYY is the bill number.
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

### Value

getAffectedRCW returns an object of type equal to the type argument (defaults to dataframe)

### Note

for more information on RCW codes, see <https://apps.leg.wa.gov/rcw/>

### Examples

```
## usage for a single bill case, XML form
getAffectedRCW("2005-06", "HB 1427", type = "xml")

## generates a dataframe of affected codes from all bills in 2007
## Not run:
bills <- getLegislationByYear("2007")
```

```
codesAffected <- getAffectedRCW("2007-08", bills$BillId)
## End(Not run)
```

---

getAmendments	<i>Get amendments to a bill</i>
---------------	---------------------------------

---

### Description

Get a list of all proposed amendments (accepted and rejected) on the bill, including the URL to the amendment text

### Usage

```
getAmendments(  
  biennium,  
  billNumber,  
  paired = TRUE,  
  type = c("df", "list", "xml")  
)
```

### Arguments

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
billNumber	Character or numeric vector containing the bill number(s) to be retrieved.
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

### Value

getAmendments returns an object of type equal to the type argument (defaults to dataframe)

### Examples

```
## get amendments for a single bill  
getAmendments("2007-08", "1001")  
  
## get amendments for a specific set of bills  
years <- c("2005-06", "2007-08", "2007-08")  
bills <- c(1447, 1219, 1001)  
  
getAmendments(years, bills, paired = TRUE, type = "df")
```

---

getBillSponsors      *Get sponsor information for a bill*

---

### Description

Get sponsor information for a bill

### Usage

```
getBillSponsors(biennium, billId, paired = TRUE, type = c("df", "list", "xml"))
```

### Arguments

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
billId	Character vector containing the bill(s) to be retrieved. Each argument should take the form "XX YYYY", where XX is the prefix (HB, SB, etc.) and YYYY is the bill number.
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

### Value

getBillSponsors returns an object of type equal to the type argument (defaults to dataframe)

### Examples

```
## get the list of all sponsors on a set of bills, filtered for primary sponsorship
spons <- getBillSponsors("2007-08", c("HB 1001", "HB 1002", "HB 1003"))
if(!is.null(spons)) sponsP <- subset(spons, Type == "Primary")
```

---

getCommitteeMembers      *Get committee members*

---

### Description

Get committee members

**Usage**

```
getCommitteeMembers(
  biennium,
  agency = c("House", "Senate"),
  name,
  paired = FALSE,
  type = c("df", "list", "xml")
)
```

**Arguments**

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
agency	One of "House" or "Senate", or a vector with these as its elements.
name	Character vector of committee names. To get the committee names for a particular session, see <a href="#">getCommittees</a> .
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

**Value**

`getCommitteeMembers` returns an object of type equal to the `type` argument (defaults to dataframe)

**Examples**

```
## get all committee members for a select number of committees and years
years <- c("2011-12", "2013-14")
comms <- c("Education", "Judiciary")

getCommitteeMembers(years, agency = "House", comms, paired = TRUE)
```

---

`getCommittees`

*Get legislative committees*

---

**Description**

Get a list of all committees that were active during the biennium, along with their respective committee code

**Usage**

```
getCommittees(biennium, type = c("df", "list", "xml"))
```

**Arguments**

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
type	One of "df", "list", or "xml". Specifies the format for the output.

**Value**

getCommittees returns an object of type equal to the type argument (defaults to dataframe)

**Examples**

```
getCommittees("2007-08")
```

---

```
getCurrentStatus      Get bill status
```

---

**Description**

Get the current status of a given bill

**Usage**

```
getCurrentStatus(  
  biennium,  
  billNumber,  
  paired = TRUE,  
  type = c("df", "list", "xml")  
)
```

**Arguments**

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
billNumber	Character or numeric vector containing the bill number(s) to be retrieved.
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

**Value**

getCurrentStatus returns an object of type equal to the type argument (defaults to dataframe)

**Note**

This function returns the bill's status as of today. If a bill was never passed, it lists the most recent status. To get a bill's complete history, use [getStatusChanges](#)

**Examples**

```

getCurrentStatus("2007-08", "1001")

## get final status for all bills written in 2011
## Not run:
billNums <- getLegislationByYear("2011")
status <- getCurrentStatus("2011-12", billNums$billNumber)
## End(Not run)

```

---

getHearings	<i>Get hearings regarding a bill</i>
-------------	--------------------------------------

---

**Description**

Get a list of dates, locations, and descriptions of all committee hearings on a particular bill

**Usage**

```
getHearings(biennium, billNumber, paired = TRUE, type = c("df", "list", "xml"))
```

**Arguments**

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
billNumber	Character or numeric vector containing the bill number(s) to be retrieved.
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

**Value**

getHearings returns an object of type equal to the type argument (defaults to dataframe)

**Note**

Due to the nature of the resulting XML document, the function trims data from excessively nested lists when type = "df". In order to access the full information, use type = "list" instead.

**Examples**

```

## get hearings for all senate bills in 2011
bills <- getLegislationByYear("2011")
if(!is.null(bills)) billsSenate <- subset(bills, OriginalAgency == "Senate")

## Not run: getHearings(billsSenate$Biennium, billsSenate$BillNumber, paired = TRUE, type = "df")

```

---

getLegislation	<i>Get summary information on a particular bill</i>
----------------	---

---

### Description

Get legislative summary information for a particular bill, including bill ID, introduction date, bill title(s), and description

### Usage

```
getLegislation(  
  biennium,  
  billNumber,  
  paired = TRUE,  
  type = c("df", "list", "xml")  
)
```

### Arguments

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
billNumber	Character or numeric vector containing the bill number(s) to be retrieved.
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

### Value

getLegislation returns an object of type equal to the type argument (defaults to dataframe)

### Examples

```
getLegislation("2007-08", "1001")  
  
## get XML data for the first 100 hundred bills of the 2007-08 session  
## Not run: getLegislation("2007-08", 1001:1100, type = "xml")
```



---

`getLegislationByYear` *Get legislation by year*

---

**Description**

Get a list of all bills introduced during the year

**Usage**

```
getLegislationByYear(year, type = c("df", "list", "xml"))
```

**Arguments**

<code>year</code>	Character or numeric vector representing the year(s) to be searched.
<code>type</code>	One of "df", "list", or "xml". Specifies the format for the output.

**Value**

`getLegislationByYear` returns an object of type equal to the `type` argument (defaults to dataframe)

**Examples**

```
## Not run: getLegislationByYear("2007")
```

---

`getLegislationSigned` *Get all bills signed into law*

---

**Description**

Get a dataframe containing all of the bills that originated in a given chamber and were eventually signed into law

**Usage**

```
getLegislationSigned(  
  biennium,  
  agency = c("House", "Senate"),  
  paired = FALSE,  
  type = c("df", "list", "xml")  
)
```

**Arguments**

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
agency	One of "House" or "Senate", or a vector with these as its elements.
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

**Value**

getLegislationSigned returns an object of type equal to the type argument (defaults to dataframe)

**Examples**

```
## get all bills signed into law from the Senate between 2007-2010
bienniums <- c("2007-08", "2009-10")
getLegislationSigned(bienniums, "Senate")
```

---

getRCWBills

*Search for bills based on Revised Code (RCW) citations*

---

**Description**

Get a list of all bills which reference or amend a particular portion of the Revised Code of Washington (RCW)

**Usage**

```
getRCWBills(biennium, rcwCite, paired = FALSE, type = c("df", "list", "xml"))
```

**Arguments**

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
rcwCite	Character vector for the citation in the RCW to pull legislation from. Optional extensions for title, chapter, and section are allowed. For more information, see <a href="https://apps.leg.wa.gov/rcw/">https://apps.leg.wa.gov/rcw/</a>
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

**Value**

getRCWBills returns an object of type equal to the type argument (defaults to dataframe)

**Examples**

```
getRCWBills("2007-08", "13.40.0357")
```

---

getRollCalls	<i>Get roll call votes</i>
--------------	----------------------------

---

**Description**

Get an XML containing roll call information for all recorded votes on a bill

**Usage**

```
getRollCalls.xml(biennium, billNumber, paired = TRUE)

getRollCalls.summary(
  biennium,
  billNumber,
  paired = TRUE,
  type = c("df", "list")
)

getRollCalls.votes(biennium, billNumber, paired = TRUE, type = c("df", "list"))
```

**Arguments**

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
billNumber	Character or numeric vector containing the bill number(s) to be retrieved.
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

**Value**

getRollCalls.xml returns a list of XML objects for each bill. getRollCalls.summary and getRollCalls.votes return objects of type equal to the type argument (defaults to dataframe)

**Note**

Due to the nested nature of the resulting document, we provide various functions to present simplified views of the data that are compatible with more parsimonious data structures. To see the full, original data, use getRollCalls.xml instead.

**Examples**

```

votes <- getRollCalls.summary("2007-08", "1001") # get roll call votes
if(!is.null(votes)) {
  length(votes) # total number of roll call votes recorded
  votes$CountYeas[3] # number of yea votes on roll call vote #3
}

## example: get member id's for all representatives voting against the bill
## on final passage
votes <- getRollCalls.votes("2007-08", "1001")
if(!is.null(votes)) {
  nay_votesFP <- subset(votes, (Motion == "Final Passage" & Vote == "Nay"))
  print(nay_votesFP$MemberId)
}

```

---

getSponsors

*Get legislators*


---

**Description**

Get a list of all sponsors (all congressmembers) for a given biennium

**Usage**

```
getSponsors(biennium, type = c("df", "list", "xml"))
```

**Arguments**

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
type	One of "df", "list", or "xml". Specifies the format for the output.

**Value**

getSponsors returns an object of type equal to the type argument (defaults to dataframe)

**Examples**

```
getSponsors("2007-08")
```

---

getStatusChanges      *Track historical progress on a bill*

---

**Description**

Get a complete history of all status changes that occurred on a particular bill

**Usage**

```
getStatusChanges(  
  biennium,  
  billNumber,  
  paired = TRUE,  
  type = c("df", "list", "xml")  
)
```

**Arguments**

biennium	Character vector representing the biennium(s) to be searched. Each argument should take the form "XXXX-YY"
billNumber	Character or numeric vector containing the bill number(s) to be retrieved.
paired	If TRUE, will assume that equal length vectors represent paired data. Set to FALSE to generate an NxN grid of input arguments. Applies to equal length vector inputs only.
type	One of "df", "list", or "xml". Specifies the format for the output.

**Value**

getStatusChanges returns an object of type equal to the type argument (defaults to dataframe)

**Examples**

```
getStatusChanges("2007-08", "1001", type = "list")
```

# Index

[getAffectedRCW](#), [2](#)  
[getAmendments](#), [3](#)  
[getBillSponsors](#), [4](#)  
[getCommitteeMembers](#), [4](#)  
[getCommittees](#), [5](#), [5](#)  
[getCurrentStatus](#), [6](#)  
[getHearings](#), [7](#)  
[getLegislation](#), [8](#)  
[getLegislationByYear](#), [9](#)  
[getLegislationSigned](#), [9](#)  
[getRCWBills](#), [10](#)  
[getRollCalls](#), [11](#)  
[getSponsors](#), [12](#)  
[getStatusChanges](#), [6](#), [13](#)