

# Package ‘RDSTK’

February 7, 2013

**Type** Package

**Title** An R wrapper for the Data Science Toolkit API

**Version** 1.0

**Depends** plyr, rjson, RCurl

**Date** 2013-02-07

**Author** Ryan Elmore and Andrew Heiss

**Maintainer** Ryan Elmore <rtelmore@gmail.com>

**Description** This package provides an R interface to Pete Warden’s Data Science Toolkit. See [www.datasciencetoolkit.org](http://www.datasciencetoolkit.org) for more information. The source code for this package can be found at [github.com/rtelmore/RDSTK](https://github.com/rtelmore/RDSTK) Happy hacking!

**License** BSD

**LazyLoad** yes

## R topics documented:

RDSTK-package . . . . .	2
coordinates2politics . . . . .	3
html2text . . . . .	4
ip2coordinates . . . . .	5
street2coordinates . . . . .	6
text2people . . . . .	7
text2sentences . . . . .	8
text2times . . . . .	9
<b>Index</b>	<b>11</b>

---

RDSTK-package*RDSTK: A R wrapper for the Data Science Toolkit API*

---

## Description

This package contains several functions that provide direct access to the Data Science Toolkit API. See [www.datasciencetoolkit.org](http://www.datasciencetoolkit.org) for an overview of the API. The package is an attempt to R-ify calls to this API.

By default the packages accesses the API at [www.datasciencetoolkit.org](http://www.datasciencetoolkit.org). Alternatively, because it is possible to clone the DSTK service on a local machine, you can point the package to an alternate API using `options("RDSTK_api_base"="http://localhost:8080")`.

**Important:** Ensure that the alternate API does *not* have a trailing slash.

## Details

Package:	RDSTK
Type:	Package
Version:	1.0
Date:	2013-02-07
License:	BSD
LazyLoad:	yes

## Author(s)

Ryan Elmore and Andrew Heiss  
Maintainer: Ryan Elmore <[rtelmore@gmail.com](mailto:rtelmore@gmail.com)>

## References

<http://www.datasciencetoolkit.org>

## Examples

```
## Not run:
ip2coordinates("134.184.34.17", 48.82.68.161")

# Use local instance of DSTK
options("RDSTK_api_base"="http://localhost:8080")

# Revert to original DSTK API
options("RDSTK_api_base"="http://www.datasciencetoolkit.org")

## End(Not run)
```

---

coordinates2politics    *Coverts latitude and longitude coordinates to politics expressions.*

---

## Description

A function to return the countries, states, provinces, cities, constituencies and neighborhoods that the latitude and longitude point lies within (from DSTK website).

## Usage

```
coordinates2politics(latitude, longitude, session=getCurlHandle())
```

## Arguments

latitude	The latitude (numeric) of the point you wish to reference.
longitude	The longitude (numeric) of the point you wish to reference.
session	This is the CURLHandle object giving the structure for the options and that will process the command. For curlMultiPerform, this is an object of class code MultiCURLHandle-class.

## Value

Returns a JSON string.

## Author(s)

Ryan Elmore

## References

<http://www.datasciencetoolkit.org/developerdocs#coordinates2politics>

## See Also

[getURL](#), [getCurlHandle](#)

## Examples

```
## Not run:  
coordinates2politics(37.769456, -122.429128)  
  
## End(Not run)
```

---

`html2text`*Identifies the text of an html string*

---

## Description

This function is used for processing an html string in order to find the main text of this string. The output is a list that contains the extracted text.

## Usage

```
html2text(html, session=getCurlHandle())
```

## Arguments

<code>html</code>	A string containing valid html code.
<code>session</code>	This is the <code>CURLHandle</code> object giving the structure for the options and that will process the command. For <code>curlMultiPerform</code> , this is an object of class code <code>MultiCURLHandle</code> -class.

## Value

A list with the main text in the html.

## Author(s)

Ryan Elmore

## References

<http://www.datasciencetoolkit.org/developerdocs#html2text>

## See Also

[curlPerform](#), [getCurlHandle](#), [dynCurlReader](#)

## Examples

```
## Not run:
html <- '<html><head><title>MyTitle</title></head><body><script type="text/javascript">something();</script>'
html2text(html)

## End(Not run)
```

---

ip2coordinates	<i>Finds geographic information related to an IP address.</i>
----------------	---

---

## Description

This function returns geographic information related to one or possibly more IP addresses.

## Usage

```
ip2coordinates(ip, session=getCurlHandle())
```

## Arguments

ip	A string containing a single IP address or multiple, comma-separated IPs.
session	This is the CURLHandle object giving the structure for the options and that will process the command. For curlMultiPerform, this is an object of class code MultiCURLHandle-class.

## Value

A data.frame containing

ip.address	IP address of the request
ip.address	Longitude of the IP address' location
country_name	Country of origin
postal_code	Post code
region	State in the US; not sure elsewhere
locality	City in the US; not sure elsewhere
country_code	Two letter country abbreviation
dma_code	Hell if I know
latitude	Latitude of the IP address' location
country_code3	If two digits aren't enough!
area_code	Area code in the US; not sure elsewhere

## Author(s)

Ryan Elmore

## References

<http://www.datasciencetoolkit.org/developerdocs#ip2coordinates>

## See Also

[getURL](#), [getCurlHandle](#)

**Examples**

```
## Not run:
ip2coordinates("134.184.34.17, 48.82.68.161")

## End(Not run)
```

---

street2coordinates	<i>Converts a street address into useful geographic information.</i>
--------------------	--

---

**Description**

This function returns a host of geographic information related to a given street address.

**Usage**

```
street2coordinates(address, session=getCurlHandle())
```

**Arguments**

address	A text string giving a street address.
session	This is the CURLHandle object giving the structure for the options and that will process the command. For curlMultiPerform, this is an object of class code MultiCURLHandle-class.

**Value**

A data frame containing:

full.address	The complete address that was analyzed.
country_name	The country of the address.
longitude	The longitude associate with the address.
fips_county	The fips county of the address. WTF?
region	The region of the address (state in US).
locality	The locality (city in US) of the address.
confidence	The degree of confidence associated with retrieving the address' information. Presumable near one is good.
street_address	Exactly as it sounds.
country_code	Country code of the address.
street number	The street number of the address.
country_code3	For those times when 2 just ain't enough!
country_code	Country code of the address.
latitude	The latitude of the address.
street_name	Why are you still reading this? It's a street name!

**Author(s)**

Ryan Elmore

## References

<http://www.datasciencetoolkit.org/developerdocs#street2coordinates>

## See Also

[getURL](#), [getCurlHandle](#)

## Examples

```
## Not run:
street2coordinates("2543 Graystone Place, Simi Valley, CA 93065")

## End(Not run)
```

---

text2people

*Finds some good info related to people*

---

## Description

This function will return information such as first and last name, title, etc. for a given person or persons.

## Usage

```
text2people(text, session=getCurlHandle())
```

## Arguments

text	A text string containing a person's name or a comma-separated list of names.
session	This is the CURLHandle object giving the structure for the options and that will process the command. For curlMultiPerform, this is an object of class code MultiCURLHandle-class.

## Value

A data.frame containing

gender	Gender of the person.
first_name	The person's first name
title	A title associated with this person.
surnames	The person's last name
start_index	The beginning of the matched string in the original string.
end_index	The end of the matched string in the original string.
matched_string	The matched string used to look up this information.

## Author(s)

Ryan Elmore

## References

<http://www.datasciencetoolkit.org/developerdocs#text2people>

## See Also

[curlPerform](#), [getCurlHandle](#), [dynCurlReader](#)

## Examples

```
## Not run:
text2people("Tim O'Reilly, Archbishop Huxley")

## End(Not run)
```

---

text2sentences	<i>Identifies sentences in a text string.</i>
----------------	---

---

## Description

This function returns the legitimate sentences (if they exist) from a text string.

## Usage

```
text2sentences(text, session=getCurlHandle())
```

## Arguments

text	A string (hopefully) containing sentences.
session	This is the CURLHandle object giving the structure for the options and that will process the command. For curlMultiPerform, this is an object of class code MultiCURLHandle-class.

## Value

A list containing	
sentences	A string identifying the sentences in the text.

## Author(s)

Ryan Elmore

## References

<http://www.datasciencetoolkit.org/developerdocs#text2sentences>

## See Also

[curlPerform](#), [getCurlHandle](#), [dynCurlReader](#)



## Examples

```
## Not run:
sentences <- "But this does, it contains enough words. So does this one, it appears correct. This is long a
text2sentences(sentences)

## End(Not run)
```

---

text2times	<i>Parses a text string for time information.</i>
------------	---

---

## Description

This function take a text string and returns any time-specific information that it finds.

## Usage

```
text2times(text, session=getCurlHandle())
```

## Arguments

text	A text string containing possible time information.
session	This is the CURLHandle object giving the structure for the options and that will process the command. For curlMultiPerform, this is an object of class code MultiCURLHandle-class.

## Value

A data.frame containing

duration	Length of time in seconds of the recognized event.
start_index	The beginning of the matched string in the original string.
is_relative	Logical value for matched string.
end_index	The end of the matched string in the original string.
time_seconds	The unix timestamp of the event (time since epoch).
matched_string	The string that was used in the processing of the request.
time_string	The time string of the recognized time event.

## Author(s)

Ryan Elmore

## References

text2times

## See Also

[curlPerform](#), [getCurlHandle](#), [dynCurlReader](#)

**Examples**

```
## Not run:  
text <- "02/01/2010, Meeting this Wednesday"  
text2times(text)  
  
## End(Not run)
```

# Index

`coordinates2politics`, [3](#)  
`curlPerform`, [4](#), [8](#), [9](#)

`dynCurlReader`, [4](#), [8](#), [9](#)

`getCurlHandle`, [3–5](#), [7–9](#)  
`getURL`, [3](#), [5](#), [7](#)

`html2text`, [4](#)

`ip2coordinates`, [5](#)

`RDSTK (RDSTK-package)`, [2](#)  
`RDSTK-package`, [2](#)

`street2coordinates`, [6](#)

`text2people`, [7](#)  
`text2sentences`, [8](#)  
`text2times`, [9](#)