

---

# Py.Saunter Documentation

*Release 0.47*

**Element 34**

**Sep 27, 2017**



---

## Contents

---

<b>1</b>	<b>saunter.testcase.remotecontrol</b>	<b>3</b>
<b>2</b>	<b>Pages</b>	<b>5</b>
<b>3</b>	<b>Elements</b>	<b>7</b>
<b>4</b>	<b>saunter.testcase.webdriver</b>	<b>9</b>
<b>5</b>	<b>Pages</b>	<b>11</b>
<b>6</b>	<b>Elements</b>	<b>13</b>
<b>7</b>	<b>saunter.testcase.base</b>	<b>15</b>
<b>8</b>	<b>saunter.generators</b>	<b>17</b>
8.1	string_data . . . . .	17
<b>9</b>	<b>saunter.providers</b>	<b>19</b>
9.1	csv_provider . . . . .	19
9.2	django_provider . . . . .	19
9.3	sqlite3_provider . . . . .	19
<b>10</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>



This is the class documentation for Py.Saunter. For the larger documentation set in context, see the [Py.Saunter page](#) at Element 34.

Remote Control:



## CHAPTER 1

---

`saunter.testcase.remotecontrol`

---

You likely do not want your scripts to inherit from this class directly. Instead, subclass it in your project and use that instead.





## CHAPTER 2

---

Pages

---



## CHAPTER 3

---

### Elements

---

WebDriver:



## CHAPTER 4

---

`saunter.testcase.webdriver`

---

You likely do not want your scripts to inherit from this class directly. Instead, subclass it in your project and use that instead.



## CHAPTER 5

---

Pages

---





## CHAPTER 6

---

### Elements

---

Common:



## CHAPTER 7

---

`saunter.testcase.base`

---

You likely do not want your scripts to inherit from this class directly. Instead, subclass it in your project and use that instead.



Generators are useful for creating data on the fly.

### string\_data

`saunter.generators.string_data.random_string(random_length=None)`

A generator for creating random string data of letters plus ' ' (whitespace)

**Params** `random_length` how many characters of random string data. if not provided, will be between 1 - 30

**Returns** String



Providers can insert data into scripts from canned resources, or act as conduits into the backend for verification purposes.

### csv\_provider

**class** `saunter.providers.csv_provider.CSVProvider(c)`  
Provides data for either data driven scripting or as oracles from a csv file

**Params** `c` name of csv file located in support/csv directory

**randomRow()**  
Gets a random row from the provider

**Returns** List

### django\_provider

**class** `saunter.providers.django_provider.DjangoProvider`  
Uses Django's Models to access the database

**get\_random\_user()**  
Gets a random user from the provider

**Returns** Dictionary

### sqlite3\_provider

**class** `saunter.providers.sqlite3_provider.DBProvider(db)`  
SQLite3 powered provider

**Params** **db** name of db file located in support/db directory

**get\_random\_user** ( )

Gets a random user from the provider

**Returns** Dictionary



## CHAPTER 10

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### S

`saunter.generators.string_data`, [17](#)  
`saunter.providers.csv_provider`, [19](#)  
`saunter.providers.django_provider`, [19](#)  
`saunter.providers.sqlite3_provider`, [19](#)  
`saunter.testcase.base`, [15](#)



## C

CSVProvider (class in saunter.providers.csv\_provider),  
[19](#)

## D

DBProvider (class in saunter.providers.sqlite3\_provider),  
[19](#)

DjangoProvider (class in  
saunter.providers.django\_provider), [19](#)

## G

get\_random\_user() (saunter.providers.django\_provider.DjangoProvider  
method), [19](#)

get\_random\_user() (saunter.providers.sqlite3\_provider.DBProvider  
method), [20](#)

## R

random\_string() (in module  
saunter.generators.string\_data), [17](#)

randomRow() (saunter.providers.csv\_provider.CSVProvider  
method), [19](#)

## S

saunter.generators.string\_data (module), [17](#)

saunter.providers.csv\_provider (module), [19](#)

saunter.providers.django\_provider (module), [19](#)

saunter.providers.sqlite3\_provider (module), [19](#)

saunter.testcase.base (module), [15](#)