

Class Contactbook

Defined in: [Contactbook.js](#).

Class Summary

[Contactbook\(\)](#)

A contactbook defines import and storage logic for a specific format and source.

Field Summary

[data](#)

Contains the contactbook data.

[dataSourceType](#)

Defines the type of the contactbook.

[implementInterface](#)

Contactbook Interface implementation declaration ('Model/Interfaces/AddressbookInterface').

[type](#)

Contactbook path definition.

Method Summary

[count\(\)](#)

Gets the number of contactbook entries.

[getEntries\(\)](#)

Gets the contactbook entries.

[load\(source, successCallback\)](#)

Loads the contactbook data initial or on every application start.

Class Detail

[Contactbook\(\)](#)

A contactbook defines import and storage logic for a specific format and source. The contactbook ist stored in the localStorage. After restarting the application, the contactbooks will be reinstantiated and injected with the data from the storage.

Field Detail

[data](#)

Contains the contactbook data.

[dataSourceType](#)

Defines the type of the contactbook. This information is used to restore the contactbook correctly from storage. The type has to correspond with the configuration in 'appConfiguration.contactbookImport.file'.

```
Addressbook.dataSourceTypes.file
```

[implementInterface](#)

Contactbook Interface implementation declaration ('Model/Interfaces/AddressbookInterface'). The contactbook

management checks for this declaration and will throw an Error, if the declaration is missing or functions are missing.

type

Contactbook path definition. Require.js uses this information to reinstantiate the contactbook after restarting the application. The path should be relatively to the 'Src' directory and the class is declared without '.js'.

```
'Model/Domain/Addressbook/AddressbookRemoteJson'
```

Method Detail

{int} **count()**

Gets the number of contactbook entries.

Returns:

{int}

{array} **getEntries()**

Gets the contactbook entries.

Returns:

{array} A list of contactbook entries. Should return a reference to a member and not a copy cause the UI will track changes on this list.

load(source, successCallback)

Loads the contactbook data initial or on every application start. Every contactbook type uses an own signature of load!

Parameters:

{string|array|url} **source**

- A string containing the content of the file the user uploaded (contactbook type 'file') or a list of strings containing the content of the files the user uploaded (contactbook type 'directory') or the url (string) of the remote contactbook data (contactbook type 'online').

{function} **successCallback** *Optional*

- This function has to be called after successful loading of the online source (contactbook type 'online' only).