

Brook framework

Luciano Souza, Silvio Clecio

September 12, 2013

Contents

1	Unit BrookAction	5
1.1	Description	5
1.2	Uses	5
1.3	Overview	6
1.4	Classes, Interfaces, Objects and Records	6
1.5	Types	16
2	Unit BrookActionHelper	17
2.1	Description	17
2.2	Uses	17
2.3	Overview	17
2.4	Classes, Interfaces, Objects and Records	18
3	Unit BrookApplication	21
3.1	Description	21
3.2	Uses	21
3.3	Overview	21
3.4	Classes, Interfaces, Objects and Records	22
3.5	Functions and Procedures	22
4	Unit BrookClasses	24
4.1	Description	24
4.2	Uses	24
4.3	Overview	24
4.4	Classes, Interfaces, Objects and Records	24
4.5	Types	25
5	Unit BrookConfigurator	26
5.1	Description	26
5.2	Uses	26
5.3	Overview	26
5.4	Classes, Interfaces, Objects and Records	27
5.5	Types	29

6	Unit BrookConsts	30
6.1	Description	30
6.2	Constants	30
6.3	Variables	50
7	Unit BrookDataBase	51
7.1	Description	51
7.2	Uses	51
7.3	Overview	51
7.4	Classes, Interfaces, Objects and Records	52
7.5	Types	57
8	Unit BrookDBAction	58
8.1	Description	58
8.2	Uses	58
8.3	Overview	59
8.4	Classes, Interfaces, Objects and Records	59
8.5	Types	61
9	Unit BrookDBConsts	62
9.1	Description	62
9.2	Uses	62
9.3	Constants	62
9.4	Variables	64
10	Unit BrookDBUtils	65
10.1	Description	65
10.2	Uses	65
10.3	Overview	65
10.4	Functions and Procedures	66
10.5	Types	66
11	Unit BrookException	68
11.1	Description	68
11.2	Uses	68
11.3	Overview	68
11.4	Classes, Interfaces, Objects and Records	68
11.5	Types	70
12	Unit BrookHTTPClient	71
12.1	Description	71
12.2	Uses	71
12.3	Overview	71
12.4	Classes, Interfaces, Objects and Records	72
12.5	Types	79

13 Unit BrookHTTPConsts	80
13.1 Description	80
13.2 Uses	80
13.3 Constants	80
13.4 Variables	97
14 Unit BrookHTTPUtils	98
14.1 Description	98
14.2 Uses	98
14.3 Overview	99
14.4 Functions and Procedures	99
14.5 Types	102
15 Unit BrookJSONHelper	103
15.1 Description	103
15.2 Uses	103
15.3 Overview	103
15.4 Classes, Interfaces, Objects and Records	104
16 Unit BrookMessages	106
16.1 Description	106
16.2 Uses	106
16.3 Overview	106
16.4 Classes, Interfaces, Objects and Records	107
16.5 Functions and Procedures	109
16.6 Types	110
16.7 Variables	110
17 Unit BrookQuery	113
17.1 Description	113
17.2 Uses	113
17.3 Overview	114
17.4 Classes, Interfaces, Objects and Records	114
17.5 Types	122
18 Unit BrookRequestHelper	123
18.1 Description	123
18.2 Uses	123
18.3 Overview	123
18.4 Classes, Interfaces, Objects and Records	124
19 Unit BrookResponseHelper	127
19.1 Description	127
19.2 Uses	127
19.3 Overview	127
19.4 Classes, Interfaces, Objects and Records	128

20 Unit BrookRESTActions	130
20.1 Description	130
20.2 Uses	130
20.3 Overview	130
20.4 Classes, Interfaces, Objects and Records	131
21 Unit BrookRouter	137
21.1 Description	137
21.2 Uses	137
21.3 Overview	138
21.4 Classes, Interfaces, Objects and Records	138
21.5 Types	143
22 Unit BrookSession	145
22.1 Description	145
22.2 Uses	145
22.3 Overview	146
22.4 Classes, Interfaces, Objects and Records	146
22.5 Types	148
23 Unit BrookTable	149
23.1 Description	149
23.2 Uses	149
23.3 Overview	150
23.4 Classes, Interfaces, Objects and Records	150
23.5 Types	158
24 Unit BrookUtils	159
24.1 Description	159
24.2 Uses	159
24.3 Overview	159
24.4 Classes, Interfaces, Objects and Records	160
24.5 Functions and Procedures	161
24.6 Types	163
24.7 Variables	164

Chapter 1

Unit BrookAction

1.1 Description

Brook Action unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

1.2 Uses

- `BrookClasses(4)`
- `BrookException(11)`
- `BrookMessages(16)`
- `BrookUtils(24)`
- `BrookHTTPUtils(14)`
- `BrookConsts(6)`
- `BrookHTTPConsts(13)`
- `HTTPDefs`
- `FPJSON`
- `Classes`
- `SysUtils`
- `Variants`

1.3 Overview

EBrookAction Class Handles exceptions for `TBrookAction`(1.4).

TBrookAction Class Provides features to handle with HTTP requests and responses.

1.4 Classes, Interfaces, Objects and Records

EBrookAction Class

Hierarchy

`EBrookAction` > `EBrook`(11.4) > `Exception`

Description

Handles exceptions for `TBrookAction`(1.4).

TBrookAction Class

Hierarchy

`TBrookAction` > `TBrookObject`(4.4) > `TObject`

Description

Provides features to handle with HTTP requests and responses.

Properties

Files public property `Files: TUploadedFiles` read `GetFiles`;
The list of files coming from a request called by the POST method.

Fields public property `Fields: TJSONObject` read `FFields`;
The list of variables coming from a request called by the POST method.

Params public property `Params: TJSONObject` read `FParams`;
The list of variables coming from a request called by the GET method.

Values public property `Values: TJSONObject` read `FValues`;
The list of variables received from parametrized URLs.

Method public property `Method: string` read `GetMethod`;
Returns the HTTP request method.

Methods

Create

Declaration `public constructor Create; virtual;`

Description Creates an instance of a `TBrookAction(1.4)` class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of `TBrookAction(1.4)` class.

FillFields

Declaration `public procedure FillFields(ARequest: TRequest); virtual;`

Description Fills the `Fields(1.4)` with data coming from a request called by means of POST method.

FillParams

Declaration `public procedure FillParams(ARequest: TRequest); virtual;`

Description Fills the `Params(1.4)` with data coming from a request called by means of GET method.

FillValues

Declaration `public procedure FillValues(ANames, AValues: TBrookArrayOfString); virtual;`

Description Fills the `Values(1.4)` with variables passed through the URL.

Register

Declaration `public class procedure Register(const APattern: string; const ADefault: Boolean = False); overload;`

Description Registers an action.

Parameters **APattern** Is an expression defining which URLs is used to call an action. It is possible to use variables inside URLs:

:name – Represents a variable that spans single URL component between slashes.

Examples:

`TMyAction.Register('/foo/:myvar');`

Value of a variable "myvar" can be read from the property `Values(1.4)`, e.g.:

`Write(Values['myvar'].AsString);`

Any number of variables can be combined:

`TMyAction.Register('/foo/:cat/:id');`

***name** – Represents a variable that spans one or more levels between slashes in the current

Examples:

```
TMyAction.Register('/home/*path');
```

Any of the following URLs will match:

`http://localhost/cgi-bin/cgi1/home/file`

`http://localhost/cgi-bin/cgi1/home/dir/file`

`http://localhost/cgi-bin/cgi1/home/dir/subdir/file` etc.

Variable Values['path'] will receive 'file', 'dir/file' or 'dir/subdir/file' correspondingly.

You can also add static text after variable part:

```
TMyAction.Register('/home/*path/download');
```

`http://localhost/cgi-bin/cgi1/home/dir/file/download` – This will match,

`http://localhost/cgi-bin/cgi1/home/dir/file/info` – but not this, because ending is different.

Multi-level variable can be combined with any number of single-level variables in any order:

```
TMyAction.Register('/home/user/:uid/file/*fpath/version/:vid/info');
```

NOTE: Only one multi-level variable can be specified per URL.

url/ – Adds a slash to the end of the URL if does not exist. Example:

```
TMyAction.Register('/foo/');
```

An action can be accessed as `http://localhost/cgi-bin/cgi1/foo` or `http://localhost/cgi-bin/cgi1/foo/`. When called as `http://localhost/cgi-bin/cgi1/foo`, it will be automatically redirected to `http://localhost/cgi-bin/cgi1/foo/`. If the pathinfo is different from `/foo` a 404 page is returned;

NOTE: Two actions can't be registered with the same pattern except when they are called by means of different HTTP methods.

ADefault A action registered as *Default* will be called automatically if the URL does not match with *Pattern* of any registered actions. It is not allowed to register more than one action as default. A typical example of use is:

```
TMyAction.Register('*', True);
```

Register

Declaration `public class procedure Register(const APattern: string; const AMethod: TBrookRequestMethod; const ADefault: Boolean = False); overload;`

Description Registers an action specifying the HTTP request method.

Parameters **AMethod** Informs the HTTP request method being valid the following options: `rmAll`, `rmGet`, `rmHead`, `rmOptions`, `rmPost`, `rmPut` or `rmDelete`. The only way to register two actions with the same pattern is differentiating the value of this parameter. If at least one action has this parameter changed, the route mapping is enabled in `Mapped(24.4)`. A typical example of use is:

```

procedure TMyAction1.Get;
begin
    Write('GET');
end;

procedure TMyAction1.Put;
begin
    Write('PUT');
end;

procedure TMyAction2.Post;
begin
    Write('POST');
end;

initialization
    TMyAction1.Register('/foo1', rmGet);
    TMyAction1.Register('/foo1', rmPut);
    TMyAction2.Register('/foo1', rmPost)

```

GetPath

Declaration public class function GetPath: string;

Description Returns the path of action. Exemple:
/cgi-bin/cgi1/myaction.

DoRequest

Declaration public procedure DoRequest(ARequest: TRequest; AResponse: TResponse);
overload; virtual;

Description Calls the method Request(1.4).

DoRequest

Declaration public procedure DoRequest(ARequest: TRequest; AResponse: TResponse; var
AHandled: Boolean); overload; virtual;

Description Calls the method Request(1.4).

UrlFor

Declaration public function UrlFor(AActionClass: TBrookActionClass): string; overload;

Description Creates an URL for action.

UrlFor

Declaration `public function UrlFor(AActionClass: TBrookActionClass; const AParams: array of string): string; overload;`

Description Creates an URL for an action informing an array of parameters. Example:

```
procedure TMyAction.Get;
begin
    // When calling with http://localhost/cgi-bin/cgi1/foo/myvalue
    // the output will be /cgi-bin/cgi1/foo/myvalue
    Write(UrlFor(TMyAction, ['myvalue']));
end;

initialization
    TMyAction.Register('/foo/:myvar')
```

UrlFor

Declaration `public function UrlFor(AClassName: string; const AParams: array of string): string; overload;`

Description Creates an URL for an action passing an array of parameters however informing the class name as string.

UrlFor

Declaration `public function UrlFor(AClassName: string): string; overload;`

Description Creates an URL for an action informing the class name as string.

UrlFor

Declaration `public function UrlFor(AActionClass: TBrookActionClass; const AParams: TJSONData): string; overload;`

Description Creates an URL for an action passing a JSON data.

UrlFor

Declaration `public function UrlFor(AClassName: string; const AParams: TJSONData): string; overload;`

Description Creates an URL for an action passing a JSON data however informing the class name as string.

Request

Declaration `public procedure Request(ARequest: TRequest; AResponse: TResponse); virtual;`

Description Is triggered by a request of any HTTP method.

Get

Declaration `public procedure Get; virtual;`

Description Is triggered by a GET HTTP request method.

Post

Declaration `public procedure Post; virtual;`

Description Is triggered by a POST HTTP request method.

Put

Declaration `public procedure Put; virtual;`

Description Is triggered by a PUT HTTP request method.

Delete

Declaration `public procedure Delete; virtual;`

Description Is triggered by a DELETE HTTP request method.

Head

Declaration `public procedure Head; virtual;`

Description Is triggered by a HEAD HTTP request method.

Options

Declaration `public procedure Options; virtual;`

Description Is triggered by a OPTIONS HTTP request method.

Redirect

Declaration `public procedure Redirect(const AUrl: string); overload;`

Description Redirects to an URL.

Redirect

Declaration `public procedure Redirect(const AUrl: string; const AStatusCode: Word); overload;`

Description Redirects to an URL informing the (302, 307) status code.

Redirect

Declaration `public procedure Redirect(const AUrl: string; const AUseRootUrl: Boolean); overload;`

Description Redirects to an URL informing the root URL.

Redirect

Declaration `public procedure Redirect(const AUrl: string; const AUseRootUrl: Boolean; const AStatusCode: Word); overload;`

Description Redirects to an URL informing the (302, 307) status code and the ScriptName.

Error

Declaration `public procedure Error(const AMsg: string); overload;`

Description Raises a message for action exceptions.

Error

Declaration `public procedure Error(const AMsg: string; const AArgs: array of const); overload;`

Description Raises a formatted message for action exceptions.

Render

Declaration `public procedure Render(const AFileName: TFileName); overload;`

Description Writes the content of a file.

Render

Declaration `public procedure Render(const AFileName: TFileName; const AArgs: array of const); overload;`

Description Writes the content of a file passing parameters to the output.

Clear

Declaration `public procedure Clear;`

Description Clears all written content with `Write()`, `WriteLn()`, `Render()` etc.

Write

Declaration `public procedure Write(const AString: string); overload;`

Description Writes a string.

Write

Declaration `public procedure Write(const ABoolean: Boolean); overload;`

Description Writes a boolean.

Write

Declaration `public procedure Write(const ABoolean: Boolean; const ATrueStr, AFalseStr: string); overload;`

Description Writes a boolean formatting output.

Write

Declaration `public procedure Write(const AInteger: Integer); overload;`

Description Writes an integer.

Write

Declaration `public procedure Write(const AFloat: Double); overload;`

Description Writes a float.

Write

Declaration `public procedure Write(const AFloat: Double; const AFormatSettings: TFormatSettings); overload;`

Description Writes a float formatting output.

Write

Declaration `public procedure Write(AStream: TStream); overload;`

Description Writes a content of stream.

Write

Declaration `public procedure Write(const AFmt: string; const AArgs: array of const); overload;`

Description Writes a formatted string.

Write

Declaration `public procedure Write(const AFmt: string; const AArgs: array of const; const AFormatSettings: TFormatSettings); overload;`

Description Writes a formatted string.

Write

Declaration `public procedure Write(AJSON: TJSONObject); overload;`

Description Writes the content of a TJSONObject.

Write

Declaration `public procedure Write(AJSON: TJSONArray); overload;`

Description Writes the content of a TJSONArray.

Write

Declaration `public procedure Write(S: TStrings); overload;`

Description Writes the content of a TStrings.

Write

Declaration `public procedure Write(const AArgs: array of const); overload;`

Description Writes value of any type. Exemple:
`Write([1, 3.14, False, 'ABC']).`

Write

Declaration `public procedure Write(const AValue: Variant); overload;`

Description Writes a variant.

WriteLn

Declaration `public procedure WriteLn(const AString: string = ES); overload;`

Description Writes a string adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(const ABoolean: Boolean); overload;`

Description Writes a boolean adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(const ABoolean: Boolean; const ATrueStr, AFalseStr: string); overload;`

Description Writes a boolean formatting output and adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(const AInteger: Integer); overload;`

Description Writes an integer adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(const AFloat: Double); overload;`

Description Writes a float adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(const AFloat: Double; const AFormatSettings: TFormatSettings); overload;`

Description Writes a float formatting output and adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(const AFmt: string; const AArgs: array of const); overload;`

Description Writes a formatted string adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(const AFmt: string; const AArgs: array of const; const AFormatSettings: TFormatSettings); overload;`

Description Writes a formatted string adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(const AArgs: array of const); overload;`

Description Writes value of any type adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(const AValue: Variant); overload;`

Description Writes a variant adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(AJSON: TJSONObject); overload;`

Description Writes the content of a `TJSONObject` adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(AJSON: TJSONArray); overload;`

Description Writes the content of a `TJSONArray` adding the BR HTML tag to the end.

WriteLn

Declaration `public procedure WriteLn(S: TStrings); overload;`

Description Writes the content of a `TStrings` adding, for each item, the BR HTML tag to the end.

1.5 Types

TBrookActionClass

Declaration `TBrookActionClass = class of TBrookAction;`

Description Is a metaclass for `TBrookAction(1.4)` class.

Chapter 2

Unit BrookActionHelper

2.1 Description

Brook Action Helper unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

2.2 Uses

- `BrookAction(1)`
- `BrookConsts(6)`
- `SysUtils`
- `HTTPDefs`
- `FPJSON`

2.3 Overview

`TBrookActionHelper` Class Adds features to `TBrookAction` class.

2.4 Classes, Interfaces, Objects and Records

TBrookActionHelper Class helper for TBrookAction

Hierarchy

TBrookActionHelper > TObject

Description

Adds features to TBrookAction class.

Methods

Alert

Declaration `public procedure Alert(const S: string);`

Description Shows a dialog with a information message.

Confirm

Declaration `public function Confirm(const S: string): Boolean;`

Description Shows a dialog with a confirmation message.

Prompt

Declaration `public function Prompt(const S, ADefaultValue: string): string;`

Description Shows a dialog with a question message.

SetCookie

Declaration `public procedure SetCookie(const AName, AValue: string; const AExpires: TDateTime = NullDate; const APath: string = ES; const ADomain: string = ES; const ASecure: Boolean = False; const AHttpOnly: Boolean = False);`

Description Creates a cookie.

GetCookie

Declaration `public function GetCookie(const AName: string): string;`

Description Get a cookie value.

DeleteCookie

Declaration `public procedure DeleteCookie(const AName: string; const APath: string = ES; const ADomain: string = ES);`

Description Deletes a cookie.

Link

Declaration `public function Link(ACaption: string; const AUrl: string = ES; const AClass: string = ES): string;`

Description Creates a link HTML tag.

LinkTo

Declaration `public function LinkTo(const ACaption: string; AActionClass: TBrookActionClass; const AParams: array of string; const AClass: string = ES): string;`

Description Creates a link HTML tag for an action passing an array of parameters however informing the class name as string

LinkTo

Declaration `public function LinkTo(const ACaption: string; AActionClass: TBrookActionClass; const AParams: TJSONData; const AClass: string = ES): string;`

Description Creates an link HTML tag for an action passing a JSON data.

LinkTo

Declaration `public function LinkTo(const ACaption: string; AActionClass: TBrookActionClass; const AClass: string = ES): string;`

Description Creates an link HTML tag for an action.

LinkTo

Declaration `public function LinkTo(const ACaption, AClassName: string; const AParams: array of string; const AClass: string = ES): string;`

Description Creates an link HTML tag for an action passing an array of parameters however informing the class name as string

LinkTo

Declaration `public function LinkTo(const ACaption, AClassName: string; const AParams: TJSONData; const AClass: string = ES): string;`

Description Creates an link HTML tag for an action passing a JSON data however informing the class name as string.

LinkTo

Declaration `public function LinkTo(const ACaption, AClassName, AClass: string): string;`

Description Creates an link HTML tag for an action informing the class name as string.

ButtonTo

Declaration `public function ButtonTo(const ACaption: string; AActionClass: TBrookActionClass; const AParams: array of string; const AValue: ShortString = 'delete'): string;`

Description Creates a button HTML tag for an action passing an array of parameters.

ButtonTo

Declaration `public function ButtonTo(const ACaption: string; AActionClass: TBrookActionClass; const AParams: TJSONData; const AValue: ShortString = 'delete'): string;`

Description Creates an button HTML tag for an action passing a JSON data.

ButtonTo

Declaration `public function ButtonTo(const ACaption: string; AActionClass: TBrookActionClass; const AValue: ShortString = 'delete'): string;`

Description Creates an button HTML tag for an action.

ButtonTo

Declaration `public function ButtonTo(const ACaption, AClassName: string; const AParams: array of string; const AValue: ShortString = 'delete'): string;`

Description Creates a button HTML tag for an action passing an array of parameters however informing the class name as string

ButtonTo

Declaration `public function ButtonTo(const ACaption, AClassName: string; const AParams: TJSONData; const AValue: ShortString = 'delete'): string;`

Description Creates an button HTML tag for an action passing a JSON data however informing the class name as string.

ButtonTo

Declaration `public function ButtonTo(const ACaption, AClassName: string; const AValue: ShortString = 'delete'): string;`

Description Creates an button HTML tag for an action informing the class name as string.

Chapter 3

Unit BrookApplication

3.1 Description

Brook Application unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

3.2 Uses

- `BrookConsts(6)`
- `BrookClasses(4)`
- `BrookException(11)`
- `BrookMessages(16)`

3.3 Overview

`EBrookApplication Class` Handles exceptions for application.

`IBrookApplication Interface` Interface of the application.

`BrookApp` Returns the application instance.

`BrookRegisterApp` Register the application.

`BrookUnregisterApp` Unregister the application.

3.4 Classes, Interfaces, Objects and Records

EBrookApplication Class

Hierarchy

EBrookApplication > EBrook(11.4) > Exception

Description

Handles exceptions for application.

IBrookApplication Interface

Hierarchy

IBrookApplication > IBrookInterface(4.4) > IInterface

Description

Interface of the application.

Methods

Instance

Declaration `public function Instance: TObject;`

Description Returns the instance of broker application.

Run

Declaration `public procedure Run;`

Description Initializes and runs the application.

Terminate

Declaration `public procedure Terminate;`

Description Terminates the application.

3.5 Functions and Procedures

BrookApp

Declaration `function BrookApp: IBrookApplication;`

Description Returns the application instance.

BrookRegisterApp

Declaration `procedure BrookRegisterApp(AApp: IBrookApplication);`

Description Register the application.

BrookUnregisterApp

Declaration `procedure BrookUnregisterApp;`

Description Unregister the application.

Chapter 4

Unit BrookClasses

4.1 Description

Brook Classes unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

4.2 Uses

- `BrookConsts(6)`

4.3 Overview

`IBrookInterface` Interface Is the main interface for Brook.

`TBrookObject` Class Is the main object for Brook.

`TBrookInterfacedObject` Class Is the main interfaced object for Brook.

4.4 Classes, Interfaces, Objects and Records

`IBrookInterface` Interface

Hierarchy

`IBrookInterface` > `IInterface`

Description

Is the main interface for Brook.

TBrookObject Class

Hierarchy

TBrookObject > TObject

Description

Is the main object for Brook.

TBrookInterfacedObject Class

Hierarchy

TBrookInterfacedObject > TInterfacedObject

Description

Is the main interfaced object for Brook.

4.5 Types

TBrookClass

Declaration `TBrookClass = class of TBrookObject;`

Description Is the main class for Brook.

Chapter 5

Unit BrookConfigurator

5.1 Description

Brook Configurator unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

5.2 Uses

- BrookClasses(4)
- BrookUtils(24)
- BrookException(11)
- BrookMessages(16)
- BrookConsts(6)
- Classes
- SysUtils
- TypInfo

5.3 Overview

EBrookConfigurator Class Handles exceptions for TBrookConfigurator(5.4).

TBrookConfigurator Class Configures objects by means of string or file.

5.4 Classes, Interfaces, Objects and Records

EBrookConfigurator Class

Hierarchy

EBrookConfigurator > EBrook(11.4) > Exception

Description

Handles exceptions for TBrookConfigurator(5.4).

TBrookConfigurator Class

Hierarchy

TBrookConfigurator > TBrookObject(4.4) > TObject

Description

Configures objects by means of string or file.

Properties

AutoLoaded `public property AutoLoaded: Boolean read FAutoLoaded write FAutoLoaded;`
Defines if the configuration is read automatically.

Configuration `public property Configuration: string read FConfiguration write FConfiguration;`

Defines the configuration using one of the following formats. Exemple:

`cfg.Configuration := 'param1=value1;param2=value2;param3=value2';`

or:

`cfg.Configuration := 'cgi1.cfg';`

This option requires the following file format:

```
param1=value1
param2=value2
param3=value2
```

NOTE: If a configuration is passed to `BrookSettings.Configuration`, the data informed in this property will be disregarded.

Target `public property Target: TObject read FTarget write FTarget;`
Defines the object to be configured.

Values `public property Values[AName: string]: string read GetValues;`
Devolve o valor de um parmetro.

ClassChecking public property ClassChecking: Boolean read FClassChecking write FClassChecking;
 Enables the class name checking, i.e., if the class of the configuring object is a 'TObject1', the following string will be informed as configuration:
 'tobject1.param1=value1;tobject1.param2=value2;tobject1.param3=value3'
 This feature is useful when more then one object will be configured in one or more configurator objects.

Methods

Create

Declaration public constructor Create;

Description Creates an instance of a TBrookConfigurator(5.4) class.

Create

Declaration public constructor Create(const AConfiguration: string); reintroduce;

Description Creates an instance of a TBrookConfigurator(5.4) class passing the configuration in its parameter.

Destroy

Declaration public destructor Destroy; override;

Description Frees an instance of TBrookConfigurator(5.4) class.

Load

Declaration public function Load: TBrookConfigurator; virtual;

Description Loads the configuration file to the memory.

Clear

Declaration public procedure Clear;

Description Clears the internal configuration.

Configure

Declaration public procedure Configure;

Description Configures the object informed in the Target property.

5.5 Types

TBrookConfiguratorClass

Declaration `TBrookConfiguratorClass = class of TBrookConfigurator;`

Description Is a metaclass for TBrookConfigurator(5.4) class.

Chapter 6

Unit BrookConsts

6.1 Description

Brook Consts unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

6.2 Constants

ES _____

Declaration ES = '';

Description Empty string.

NS _____

Declaration NS = 'null';

Description Null string.

BR _____

Declaration BR = '
';

Description BR HTML tag.

HS

Declaration HS = ' : ';

Description Param separator.

LE

Declaration LE = LineEnding;

Description Line ending.

NU

Declaration NU = #0;

Description NULL

HT

Declaration HT = #9;

Description TAB

LF

Declaration LF = #10;

Description LF

CR

Declaration CR = #13;

Description CR

CRLF

Declaration CRLF = CR + LF;

Description CRLF

NullDate

Declaration NullDate = 0;

Description Null date.

SP _____

Declaration SP = #32;

Description Space.

DQ _____

Declaration DQ = #34;

Description ”

PO _____

Declaration PO = #35;

Description #

PT _____

Declaration PT = #37;

Description %

AM _____

Declaration AM = #38;

Description &

AP _____

Declaration AP = #39;

Description ’

PS _____

Declaration PS = #40;

Description (

PE _____

Declaration PE = #41;

Description)

AK _____

Declaration AK = #42;

Description *

CS _____

Declaration CS = #44;

Description ,

DT _____

Declaration DT = #46;

Description .

US _____

Declaration US = #47;

Description /

CO _____

Declaration CO = #58;

Description :

SC _____

Declaration SC = #59;

Description ;

LT _____

Declaration LT = #60;

Description <

EQ _____

Declaration EQ = #61;

Description =

GT

Declaration GT = #62;

Description >

QU

Declaration QU = #63;

Description ?

AT

Declaration AT = #64;

Description @

DS

Declaration DS = #92;

Description \

LS

Declaration LS = #91;

Description [

RS

Declaration RS = #93;

Description]

BROOK_GUID

Declaration BROOK_GUID = '{D2BDD8EF-78C0-47CD-95C3-664CDFFDAA9E}';

BROOK_APP_GUID

Declaration BROOK_APP_GUID = '{669B03B7-AA2D-4B64-AAFC-4FBD4A41267E}';

BROOK_ERROR_MASK

Declaration BROOK_ERROR_MASK = '%s: %s';

BROOK_GMT_FRMT

Declaration BROOK_GMT_FRMT = '%s, %.2d %s %d %.2d:%.2d:%.2d GMT';

BROOK_UUID_MASK

Declaration BROOK_UUID_MASK = '%.8x%.4x%.4x%.2x%.2x%.2x%.2x%.2x%.2x%.2x';

BROOK_UUID_SEP_MASK

Declaration BROOK_UUID_SEP_MASK = '%.8x-%.4x-%.4x%.2x%.2x%.2x%.2x%.2x%.2x%.2x';

BROOK_OVERRIDDEN_METHOD

Declaration BROOK_OVERRIDDEN_METHOD = '_METHOD';

BROOK_DEFAULT_LANGUAGE

Declaration BROOK_DEFAULT_LANGUAGE = 'en-US';

BROOK_SESS_ID

Declaration BROOK_SESS_ID = 'BRKSESSID';

BROOK_SESS_PREFIX

Declaration BROOK_SESS_PREFIX = 'brook_sess_';

BROOK_SESS_DEFAULT_TIMEOUT

Declaration BROOK_SESS_DEFAULT_TIMEOUT = 3 * 60;

Description 3 minutes.

BROOK_MAX_MIME_TYPE

Declaration BROOK_MAX_MIME_TYPE = 682;

BROOK_MIME_TYPE

Declaration BROOK_MIME_TYPE: array[0..BROOK_MAX_MIME_TYPE, 0..2] of string = (('3D Crossword Plugin', 'application/vnd.hzn-3d-crossword', '.x3d'), ('3GP', 'video/3gpp', '.3gp'), ('3GP2', 'video/3gpp2', '.3g2'), ('3GPP MSEQ File', 'application/vnd.mseq', '.mseq'), ('3M Post It Notes', 'application/vnd.3m.post-it-notes', '.pwn'), ('3rd Generation Partnership Project - Pic Large', 'application/vnd.3gpp.pic-bw-large', '.plb'), ('3rd Generation Partnership Project - Pic Small', 'application/vnd.3gpp.pic-bw-small', '.psb'), ('3rd Generation Partnership Project - Pic Var', 'application/vnd.3gpp.pic-bw-var', '.pvb'), ('3rd Generation Partnership Project - Transaction Capabilities Application Part', 'application/vnd.3gpp2.tcap', '.tcap'), ('7-Zip', 'application/x-7z-compressed', '.7z'), ('AbiWord', 'application/x-abiword', '.abw'), ('Ace Archive', 'application/x-ace-compressed', '.ace'), ('Active

Content Compression', 'application/vnd.americandynamics.acc', '.acc'), ('ACU
 Cobol', 'application/vnd.acucobol', '.acu'), ('ACU Cobol',
 'application/vnd.acucorp', '.atc'), ('Adaptive differential pulse-code
 modulation', 'audio/adpcm', '.adp'), ('Adobe (Macropedia) Authorware - Binary
 File', 'application/x-authorware-bin', '.aab'), ('Adobe (Macropedia)
 Authorware - Map', 'application/x-authorware-map', '.aam'), ('Adobe
 (Macropedia) Authorware - Segment File', 'application/x-authorware-seg',
 '.aas'), ('Adobe AIR Application',
 'application/vnd.adobe.air-application-installer-package+zip', '.air'),
 ('Adobe Flash', 'application/x-shockwave-flash', '.swf'), ('Adobe Flex
 Project', 'application/vnd.adobe.fxp', '.fxp'), ('Adobe Portable Document
 Format', 'application/pdf', '.pdf'), ('Adobe PostScript Printer Description
 File Format', 'application/vnd.cups-ppd', '.ppd'), ('Adobe Shockwave Player',
 'application/x-director', '.dir'), ('Adobe XML Data Package',
 'application/vnd.adobe.xdp+xml', '.xdp'), ('Adobe XML Forms Data Format',
 'application/vnd.adobe.xfdf', '.xfdf'), ('Advanced Audio Coding (AAC)',
 'audio/x-aac', '.aac'), ('Ahead AIR Application',
 'application/vnd.ahead.space', '.ahead'), ('AirZip FileSECURE',
 'application/vnd.airzip.filesecure.azf', '.azf'), ('AirZip FileSECURE',
 'application/vnd.airzip.filesecure.azs', '.azs'), ('Amazon Kindle eBook
 format', 'application/vnd.amazon.ebook', '.azw'), ('AmigaDE',
 'application/vnd.amiga.ami', '.ami'), ('Andrew Toolkit',
 'application/andrew-inset', 'N/A'), ('Android Package Archive',
 'application/vnd.android.package-archive', '.apk'), ('ANSER-WEB Terminal
 Client - Certificate Issue',
 'application/vnd.anser-web-certificate-issue-initiation', '.cii'),
 ('ANSER-WEB Terminal Client - Web Funds Transfer',
 'application/vnd.anser-web-funds-transfer-initiation', '.fti'), ('Antix Game
 Player', 'application/vnd.antix.game-component', '.atx'), ('Apple Installer
 Package', 'application/vnd.apple.installer+xml', '.mpkg'), ('Applixware',
 'application/applixware', '.aw'), ('Archipelago Lesson Player',
 'application/vnd.hhe.lesson-player', '.les'), ('Arista Networks Software
 Image', 'application/vnd.aristanetworks.swi', '.swi'), ('Assembler Source
 File', 'text/x-asm', '.s'), ('Atom Publishing Protocol',
 'application/atomcat+xml', '.atomcat'), ('Atom Publishing Protocol Service
 Document', 'application/atomsvc+xml', '.atomsvc'), ('Atom Syndication
 Format', 'application/atom+xml', '.atom', '.xml'), ('Attribute Certificate',
 'application/pkix-attr-cert', '.ac'), ('Audio Interchange File Format',
 'audio/x-aiff', '.aif'), ('Audio Video Interleave (AVI)', 'video/x-msvideo',
 '.avi'), ('Audiograph', 'application/vnd.audiograph', '.aep'), ('AutoCAD
 DXF', 'image/vnd.dxf', '.dxf'), ('Autodesk Design Web Format (DWF)',
 'model/vnd.dwf', '.dwf'), ('Binary CPIO Archive', 'application/x-bcpio',
 '.bcpio'), ('Binary Data', 'application/octet-stream', '.bin'), ('Bitmap
 Image File', 'image/bmp', '.bmp'), ('BitTorrent', 'application/x-bittorrent',
 '.torrent'), ('Blackberry COD File', 'application/vnd.rim.cod', '.cod'),
 ('Blueice Research Multipass', 'application/vnd.blueice.multipass', '.mpm'),

('BMI Drawing Data Interchange', 'application/vnd.bmi', '.bmi'), ('Bourne Shell Script', 'application/x-sh', '.sh'), ('BTIF', 'image/prs.btif', '.btif'), ('BusinessObjects', 'application/vnd.businessobjects', '.rep'), ('Bzip Archive', 'application/x-bzip', '.bz'), ('Bzip2 Archive', 'application/x-bzip2', '.bz2'), ('C Shell Script', 'application/x-csh', '.csh'), ('C Source File', 'text/x-c', '.c'), ('CambridgeSoft Chem Draw', 'application/vnd.chemdraw+xml', '.cdxml'), ('Cascading Style Sheets (CSS)', 'text/css', '.css'), ('ChemDraw eXchange file', 'chemical/x-cdx', '.cdx'), ('Chemical Markup Language', 'chemical/x-cml', '.cml'), ('Chemical Style Markup Language', 'chemical/x-csml', '.csml'), ('CIM Database', 'application/vnd.contact.cmsg', '.cdbcmg'), ('Claymore Data Files', 'application/vnd.claymore', '.cla'), ('Clonk Game', 'application/vnd.clonk.c4group', '.c4g'), ('Close Captioning - Subtitle', 'image/vnd.dvb.subtitle', '.sub'), ('Cloud Data Management Interface (CDMI) - Capability', 'application/cdmi-capability', '.cdmia'), ('Cloud Data Management Interface (CDMI) - Container', 'application/cdmi-container', '.cdmic'), ('Cloud Data Management Interface (CDMI) - Domain', 'application/cdmi-domain', '.cdmid'), ('Cloud Data Management Interface (CDMI) - Object', 'application/cdmi-object', '.cdmio'), ('Cloud Data Management Interface (CDMI) - Queue', 'application/cdmi-queue', '.cdmiq'), ('ClueTrust CartoMobile - Config', 'application/vnd.cluetrust.cartomobile-config', '.c11amc'), ('ClueTrust CartoMobile - Config Package', 'application/vnd.cluetrust.cartomobile-config-pkg', '.c11amz'), ('CMU Image', 'image/x-cmu-raster', '.ras'), ('COLLADA', 'model/vnd.collada+xml', '.dae'), ('Comma-Separated Values', 'text/csv', '.csv'), ('Compact Pro', 'application/mac-compactpro', '.cpt'), ('Compiled Wireless Markup Language (WMLC)', 'application/vnd.wap.wmlc', '.wmlc'), ('Computer Graphics Metafile', 'image/cgm', '.cgm'), ('CoolTalk', 'x-conference/x-cooltalk', '.ice'), ('Corel Metafile Exchange (CMX)', 'image/x-cmx', '.cmx'), ('CorelXARA', 'application/vnd.xara', '.xar'), ('CosmoCaller', 'application/vnd.cosmocaller', '.cmc'), ('CPIO Archive', 'application/x-cpio', '.cpio'), ('CrickSoftware - Clicker', 'application/vnd.crick.clicker', '.clkx'), ('CrickSoftware - Clicker - Keyboard', 'application/vnd.crick.clicker.keyboard', '.clkk'), ('CrickSoftware - Clicker - Palette', 'application/vnd.crick.clicker.palette', '.clkp'), ('CrickSoftware - Clicker - Template', 'application/vnd.crick.clicker.template', '.clkt'), ('CrickSoftware - Clicker - Wordbank', 'application/vnd.crick.clicker.wordbank', '.clkw'), ('Critical Tools - PERT Chart EXPERT', 'application/vnd.criticaltools.wbs+xml', '.wbs'), ('CryptoNote', 'application/vnd.rig.cryptonote', '.cryptonote'), ('Crystallographic Interchange Format', 'chemical/x-cif', '.cif'), ('CrystalMaker Data Format', 'chemical/x-cmdf', '.cmdf'), ('CU-SeeMe', 'application/cu-seeme', '.cu'), ('CU-Writer', 'application/prs.cww', '.cww'), ('Curl - Applet', 'text/vnd.curl', '.curl'), ('Curl - Detached Applet',

'text/vnd.curl.dcurl', '.dcurl'), ('Curl - Manifest File',
 'text/vnd.curl.mcurl', '.mcurl'), ('Curl - Source Code',
 'text/vnd.curl.scurl', '.scurl'), ('CURL Applet', 'application/vnd.curl.car',
 '.car'), ('CURL Applet', 'application/vnd.curl.pcurl', '.pcurl'),
 ('CustomMenu', 'application/vnd.yellowriver-custom-menu', '.cmp'), ('Data
 Structure for the Security Suitability of Cryptographic Algorithms',
 'application/dssc+der', '.dssc'), ('Data Structure for the Security
 Suitability of Cryptographic Algorithms', 'application/dssc+xml', '.xdssc'),
 ('Debian Package', 'application/x-debian-package', '.deb'), ('DECE Audio',
 'audio/vnd.dece.audio', '.uva'), ('DECE Graphic', 'image/vnd.dece.graphic',
 '.uvi'), ('DECE High Definition Video', 'video/vnd.dece.hd', '.uvh'), ('DECE
 Mobile Video', 'video/vnd.dece.mobile', '.uvm'), ('DECE MP4',
 'video/vnd.uvu.mp4', '.uvu'), ('DECE PD Video', 'video/vnd.dece.pd',
 '.uvp'), ('DECE SD Video', 'video/vnd.dece.sd', '.uvs'), ('DECE Video',
 'video/vnd.dece.video', '.uvv'), ('Device Independent File Format (DVI)',
 'application/x-dvi', '.dvi'), ('Digital Siesmograph Networks - SEED
 Datafiles', 'application/vnd.fdsn.seed', '.seed'), ('Digital Talking Book',
 'application/x-dtbook+xml', '.dtb'), ('Digital Talking Book - Resource File',
 'application/x-dtbresource+xml', '.res'), ('Digital Video Broadcasting',
 'application/vnd.dvb.ait', '.ait'), ('Digital Video Broadcasting',
 'application/vnd.dvb.service', '.svc'), ('Digital Winds Music',
 'audio/vnd.digital-winds', '.eol'), ('DjVu', 'image/vnd.djvu', '.djvu'),
 ('Document Type Definition', 'application/xml-dtd', '.dtd'), ('Dolby Meridian
 Lossless Packing', 'application/vnd.dolby.mlp', '.mlp'), ('Doom Video Game',
 'application/x-doom', '.wad'), ('DPGraph', 'application/vnd.dpgraph',
 '.dpg'), ('DRA Audio', 'audio/vnd.dra', '.dra'), ('DreamFactory',
 'application/vnd.dreamfactory', '.dfac'), ('DTS Audio', 'audio/vnd.dts',
 '.dts'), ('DTS High Definition Audio', 'audio/vnd.dts.hd', '.dtshd'), ('DWG
 Drawing', 'image/vnd.dwg', '.dwg'), ('DynaGeo', 'application/vnd.dynageo',
 '.geo'), ('ECMAScript', 'application/ecmascript', '.es'), ('EcoWin Chart',
 'application/vnd.ecowin.chart', '.mag'), ('EDMICS 2000',
 'image/vnd.fujixerox.edmics-mmr', '.mmr'), ('EDMICS 2000',
 'image/vnd.fujixerox.edmics-rlc', '.rlc'), ('Efficient XML Interchange',
 'application/exi', '.exi'), ('EFI Proteus',
 'application/vnd.proteus.magazine', '.mgz'), ('Electronic Publication',
 'application/epub+zip', '.epub'), ('Email Message', 'message/rfc822',
 '.eml'), ('Enliven Viewer', 'application/vnd.enliven', '.nml'), ('Express by
 Infoseek', 'application/vnd.is-xpr', '.xpr'), ('eXtended Image File Format
 (XIFF)', 'image/vnd.xiff', '.xif'), ('Extensible Forms Description Language',
 'application/vnd.xfdl', '.xfdl'), ('Extensible MultiModal Annotation',
 'application/emma+xml', '.emma'), ('EZPix Secure Photo Album',
 'application/vnd.ezpix-album', '.ez2'), ('EZPix Secure Photo Album',
 'application/vnd.ezpix-package', '.ez3'), ('FAST Search & Transfer ASA',
 'image/vnd.fst', '.fst'), ('FAST Search & Transfer ASA', 'video/vnd.fvt',
 '.fvt'), ('FastBid Sheet', 'image/vnd.fastbidsheet', '.fbs'), ('FCS Express
 Layout Link', 'application/vnd.denovo.fcselayout-link', '.fe_launch'),

('Flash Video', 'video/x-f4v', '.f4v'), ('Flash Video', 'video/x-flv',
 '.flv'), ('FlashPix', 'image/vnd.fpx', '.fpx'), ('FlashPix',
 'image/vnd.net-fpx', '.npx'), ('FLEXSTOR', 'text/vnd.fmi.flexstor', '.flx'),
 ('FLI/FLC Animation Format', 'video/x-fli', '.fli'), ('FluxTime Clip',
 'application/vnd.fluxtime.clip', '.ftc'), ('Forms Data Format',
 'application/vnd.fdf', '.fdf'), ('Fortran Source File', 'text/x-fortran',
 '.f'), ('FrameMaker Interchange Format', 'application/vnd.mif', '.mif'),
 ('FrameMaker Normal Format', 'application/vnd.frameMaker', '.fm'), ('FreeHand
 MX', 'image/x-freehand', '.fh'), ('Friendly Software Corporation',
 'application/vnd.fsc.weblaunch', '.fsc'), ('Frogans Player',
 'application/vnd.frogans.fnc', '.fnc'), ('Frogans Player',
 'application/vnd.frogans.ltf', '.ltf'), ('Fujitsu - Xerox 2D CAD Data',
 'application/vnd.fujixerox.ddd', '.ddd'), ('Fujitsu - Xerox DocuWorks',
 'application/vnd.fujixerox.docuworks', '.xdw'), ('Fujitsu - Xerox DocuWorks
 Binder', 'application/vnd.fujixerox.docuworks.binder', '.xbd'), ('Fujitsu
 Oasys', 'application/vnd.fujitsu.oasys', '.oas'), ('Fujitsu Oasys',
 'application/vnd.fujitsu.oasys2', '.oa2'), ('Fujitsu Oasys',
 'application/vnd.fujitsu.oasys3', '.oa3'), ('Fujitsu Oasys',
 'application/vnd.fujitsu.oasysgp', '.fg5'), ('Fujitsu Oasys',
 'application/vnd.fujitsu.oasysprs', '.bh2'), ('FutureSplash Animator',
 'application/x-futuresplash', '.spl'), ('FuzzySheet',
 'application/vnd.fuzzysheet', '.fzs'), ('G3 Fax Image', 'image/g3fax',
 '.g3'), ('GameMaker ActiveX', 'application/vnd.gmx', '.gmx'), ('Gen-Trix
 Studio', 'model/vnd.gtw', '.gtw'), ('Genomatix Tuxedo Framework',
 'application/vnd.genomatix.tuxedo', '.txd'), ('GeoGebra',
 'application/vnd.geogebra.file', '.ggb'), ('GeoGebra',
 'application/vnd.geogebra.tool', '.ggt'), ('Geometric Description Language
 (GDL)', 'model/vnd.gdl', '.gdl'), ('GeoMetry Explorer',
 'application/vnd.geometry-explorer', '.gex'), ('GEONExT and JSXGraph',
 'application/vnd.geonext', '.gxt'), ('GeoplanW', 'application/vnd.geoplan',
 '.g2w'), ('GeospaceW', 'application/vnd.geospace', '.g3w'), ('Ghostscript
 Font', 'application/x-font-ghostscript', '.gsf'), ('Glyph Bitmap Distribution
 Format', 'application/x-font-bdf', '.bdf'), ('GNU Tar Files',
 'application/x-gtar', '.gtar'), ('GNU Texinfo Document',
 'application/x-texinfo', '.texinfo'), ('Gnumeric', 'application/x-gnumeric',
 '.gnumeric'), ('Google Earth - KML', 'application/vnd.google-earth.kml+xml',
 '.kml'), ('Google Earth - Zipped KML', 'application/vnd.google-earth.kmz',
 '.kmz'), ('GrafEq', 'application/vnd.grafeq', '.gqf'), ('Graphics Interchange
 Format', 'image/gif', '.gif'), ('Graphviz', 'text/vnd.graphviz', '.gv'),
 ('Groove - Account', 'application/vnd.groove-account', '.gac'), ('Groove -
 Help', 'application/vnd.groove-help', '.ghf'), ('Groove - Identity Message',
 'application/vnd.groove-identity-message', '.gim'), ('Groove - Injector',
 'application/vnd.groove-injector', '.grv'), ('Groove - Tool Message',
 'application/vnd.groove-tool-message', '.gtm'), ('Groove - Tool Template',
 'application/vnd.groove-tool-template', '.tpl'), ('Groove - Vcard',
 'application/vnd.groove-vcard', '.vcg'), ('H.261', 'video/h261', '.h261'),

('H.263', 'video/h263', '.h263'), ('H.264', 'video/h264', '.h264'), ('Hewlett
 Packard Instant Delivery', 'application/vnd.hp-hpid', '.hpid'),
 ('Hewlett-Packard's WebPrintSmart', 'application/vnd.hp-hps', '.hps'),
 ('Hierarchical Data Format', 'application/x-hdf', '.hdf'), ('Hit'n'Mix',
 'audio/vnd.rip', '.rip'), ('Homebanking Computer Interface (HBCI)',
 'application/vnd.hbci', '.hbci'), ('HP Indigo Digital Press - Job Layout
 Language', 'application/vnd.hp-jlyt', '.jlt'), ('HP Printer Command
 Language', 'application/vnd.hp-pcl', '.pcl'), ('HP-GL/2 and HP RTL',
 'application/vnd.hp-hpgl', '.hpgl'), ('HV Script',
 'application/vnd.yamaha.hv-script', '.hvs'), ('HV Voice Dictionary',
 'application/vnd.yamaha.hv-dic', '.hvd'), ('HV Voice Parameter',
 'application/vnd.yamaha.hv-voice', '.hvp'), ('Hydrostatix Master Suite',
 'application/vnd.hydrostatix.sof-data', '.sfd-hdstx'), ('Hyperstudio',
 'application/hyperstudio', '.stk'), ('Hypertext Application Language',
 'application/vnd.hal+xml', '.hal'), ('HyperText Markup Language (HTML)',
 'text/html', '.html'), ('IBM DB2 Rights Manager',
 'application/vnd.ibm.rights-management', '.irm'), ('IBM Electronic Media
 Management System - Secure Container',
 'application/vnd.ibm.secure-container', '.sc'), ('iCalendar',
 'text/calendar', '.ics'), ('ICC profile', 'application/vnd.iccprofile',
 '.icc'), ('Icon Image', 'image/x-icon', '.ico'), ('igLoader',
 'application/vnd.igloader', '.igl'), ('Image Exchange Format', 'image/ief',
 '.ief'), ('ImmerVision PURE Players', 'application/vnd.immervision-ivp',
 '.ivp'), ('ImmerVision PURE Players', 'application/vnd.immervision-ivu',
 '.ivu'), ('IMS Networks', 'application/reginfo+xml', '.rif'), ('In3D - 3DML',
 'text/vnd.in3d.3dml', '.3dml'), ('In3D - 3DML', 'text/vnd.in3d.spot',
 '.spot'), ('Initial Graphics Exchange Specification (IGES)', 'model/iges',
 '.igs'), ('Interactive Geometry Software', 'application/vnd.intergeo',
 '.i2g'), ('Interactive Geometry Software Cinderella',
 'application/vnd.cinderella', '.cdy'), ('Intercon FormNet',
 'application/vnd.intercon.formnet', '.xpw'), ('International Society for
 Advancement of Cytometry', 'application/vnd.isac.fcs', '.fcs'), ('Internet
 Protocol Flow Information Export', 'application/ipfix', '.ipfix'), ('Internet
 Public Key Infrastructure - Certificate', 'application/pkix-cert', '.cer'),
 ('Internet Public Key Infrastructure - Certificate Management Protocols',
 'application/pkixcmp', '.pki'), ('Internet Public Key Infrastructure -
 Certificate Revocation Lists', 'application/pkix-crl', '.crl'), ('Internet
 Public Key Infrastructure - Certification Path', 'application/pkix-pkipath',
 '.pkipath'), ('IOCOM Visimeet', 'application/vnd.insors.igm', '.igm'), ('IP
 Unplugged Roaming Client', 'application/vnd.ipunplugged.rcprofile',
 '.rcprofile'), ('iRepository / Lucidoc Editor',
 'application/vnd.irepository.package+xml', '.irp'), ('J2ME App Descriptor',
 'text/vnd.sun.j2me.app-descriptor', '.jad'), ('Java Archive',
 'application/java-archive', '.jar'), ('Java Bytecode File',
 'application/java-vm', '.class'), ('Java Network Launching Protocol',
 'application/x-java-jnlp-file', '.jnlp'), ('Java Serialized Object',

'application/java-serialized-object', '.ser'), ('Java Source File',
 'text/x-java-source,java', '.java'), ('JavaScript', 'application/javascript',
 '.js'), ('JavaScript Object Notation (JSON)', 'application/json', '.json'),
 ('Joda Archive', 'application/vnd.joost.joda-archive', '.joda'), ('JPEG 2000
 Compound Image File Format', 'video/jpm', '.jpm'), ('JPEG Image',
 'image/jpeg', '.jpeg, .jpg'), ('JPGVideo', 'video/jpeg', '.jpgv'),
 ('Kahootz', 'application/vnd.kahootz', '.ktz'), ('Karaoke on Chipnuts
 Chipsets', 'application/vnd.chipnuts.karaoke-mmd', '.mmd'), ('KDE KOffice
 Office Suite - Karbon', 'application/vnd.kde.karbon', '.karbon'), ('KDE
 KOffice Office Suite - KChart', 'application/vnd.kde.kchart', '.chrt'), ('KDE
 KOffice Office Suite - Kformula', 'application/vnd.kde.kformula', '.kfo'),
 ('KDE KOffice Office Suite - Kivio', 'application/vnd.kde.kivio', '.flw'),
 ('KDE KOffice Office Suite - Kontour', 'application/vnd.kde.kontour',
 '.kon'), ('KDE KOffice Office Suite - Kpresenter',
 'application/vnd.kde.kpresenter', '.kpr'), ('KDE KOffice Office Suite -
 Kspread', 'application/vnd.kde.kspread', '.ksp'), ('KDE KOffice Office Suite
 - Kword', 'application/vnd.kde.kword', '.kwd'), ('Kenamea App',
 'application/vnd.kenameaapp', '.htke'), ('Kidspiration',
 'application/vnd.kidspiration', '.kia'), ('Kinar Applications',
 'application/vnd.kinar', '.kne'), ('Kodak Storyshare',
 'application/vnd.kodak-descriptor', '.sse'), ('Laser App Enterprise',
 'application/vnd.las.las+xml', '.lasxml'), ('LaTeX', 'application/x-latex',
 '.latex'), ('Life Balance - Desktop Edition',
 'application/vnd.llamagraphics.life-balance.desktop', '.lbd'), ('Life Balance
 - Exchange Format',
 'application/vnd.llamagraphics.life-balance.exchange+xml', '.lbe'),
 ('Lightspeed Audio Lab', 'application/vnd.jam', '.jam'), ('Lotus 1-2-3',
 'application/vnd.lotus-1-2-3', '.123'), ('Lotus Approach',
 'application/vnd.lotus-approach', '.apr'), ('Lotus Freelance',
 'application/vnd.lotus-freelance', '.pre'), ('Lotus Notes',
 'application/vnd.lotus-notes', '.nsf'), ('Lotus Organizer',
 'application/vnd.lotus-organizer', '.org'), ('Lotus Screencam',
 'application/vnd.lotus-screencam', '.scm'), ('Lotus Wordpro',
 'application/vnd.lotus-wordpro', '.lwp'), ('Lucent Voice',
 'audio/vnd.lucent.voice', '.lvp'), ('M3U (Multimedia Playlist)',
 'audio/x-mpegurl', '.m3u'), ('M4v', 'video/x-m4v', '.m4v'), ('Macintosh
 BinHex 4.0', 'application/mac-binhex40', '.hqx'), ('MacPorts Port System',
 'application/vnd.macports.portpkg', '.portpkg'), ('MapGuide DBXML',
 'application/vnd.osgeo.mapguide.package', '.mvp'), ('MARC Formats',
 'application/marc', '.mrc'), ('MARC21 XML Schema', 'application/marcxml+xml',
 '.mrcx'), ('Material Exchange Format', 'application/mxf', '.mxf'),
 ('Mathematica Notebook Player', 'application/vnd.wolfram.player', '.nbp'),
 ('Mathematica Notebooks', 'application/mathematica', '.ma'), ('Mathematical
 Markup Language', 'application/mathml+xml', '.mathml'), ('Mbox database
 files', 'application/mbox', '.mbox'), ('MedCalc',
 'application/vnd.medcalldata', '.mc1'), ('Media Server Control Markup

Language', 'application/mediaservercontrol+xml', '.mscml'), ('MediaRemote',
 'application/vnd.mediastation.cdkey', '.cdkey'), ('Medical Waveform Encoding
 Format', 'application/vnd.mfer', '.mwf'), ('Melody Format for Mobile
 Platform', 'application/vnd.mfmp', '.mfmp'), ('Mesh Data Type', 'model/mesh',
 '.msh'), ('Metadata Authority Description Schema', 'application/mads+xml',
 '.mads'), ('Metadata Encoding and Transmission Standard',
 'application/mets+xml', '.mets'), ('Metadata Object Description Schema',
 'application/mods+xml', '.mods'), ('Metalink', 'application/metalink4+xml',
 '.meta4'), ('Microsoft PowerPoint - Macro-Enabled Template File',
 'application/vnd.ms-powerpoint.template.macroenabled.12', '.potm'),
 ('Microsoft Word - Macro-Enabled Document',
 'application/vnd.ms-word.document.macroenabled.12', '.docm'), ('Microsoft Word
 - Macro-Enabled Template',
 'application/vnd.ms-word.template.macroenabled.12', '.dotm'), ('Micro CADAM
 Helix D&D', 'application/vnd.mcd', '.mcd'), ('Micrografx',
 'application/vnd.micrografx.flo', '.flo'), ('Micrografx iGrafX Professional',
 'application/vnd.micrografx.igx', '.igx'), ('MICROSEC e-Sign',
 'application/vnd.eszigno3+xml', '.es3'), ('Microsoft Access',
 'application/x-msaccess', '.mdb'), ('Microsoft Advanced Systems Format
 (ASF)', 'video/x-ms-asf', '.asf'), ('Microsoft Application',
 'application/x-msdownload', '.exe'), ('Microsoft Artgalry',
 'application/vnd.ms-artgalry', '.cil'), ('Microsoft Cabinet File',
 'application/vnd.ms-cab-compressed', '.cab'), ('Microsoft Class Server',
 'application/vnd.ms-ims', '.ims'), ('Microsoft ClickOnce',
 'application/x-ms-application', '.application'), ('Microsoft Clipboard Clip',
 'application/x-msclip', '.clp'), ('Microsoft Document Imaging Format',
 'image/vnd.ms-modi', '.mdi'), ('Microsoft Embedded OpenType',
 'application/vnd.ms-fontobject', '.eot'), ('Microsoft Excel',
 'application/vnd.ms-excel', '.xls'), ('Microsoft Excel - Add-In File',
 'application/vnd.ms-excel.addin.macroenabled.12', '.xlam'), ('Microsoft Excel
 - Binary Workbook', 'application/vnd.ms-excel.sheet.binary.macroenabled.12',
 '.xlsb'), ('Microsoft Excel - Macro-Enabled Template File',
 'application/vnd.ms-excel.template.macroenabled.12', '.xltm'), ('Microsoft
 Excel - Macro-Enabled Workbook',
 'application/vnd.ms-excel.sheet.macroenabled.12', '.xlsm'), ('Microsoft Html
 Help File', 'application/vnd.ms-htmlhelp', '.chm'), ('Microsoft Information
 Card', 'application/x-mscardfile', '.crd'), ('Microsoft Learning Resource
 Module', 'application/vnd.ms-lrm', '.lrm'), ('Microsoft MediaView',
 'application/x-msmediaview', '.mvb'), ('Microsoft Money',
 'application/x-msmoney', '.mny'), ('Microsoft Office - OOXML - Presentation',
 'application/vnd.openxmlformats-officedocument.presentationml.presentation',
 '.pptx'), ('Microsoft Office - OOXML - Presentation (Slide)',
 'application/vnd.openxmlformats-officedocument.presentationml.slide',
 '.sldx'), ('Microsoft Office - OOXML - Presentation (Slideshow)',
 'application/vnd.openxmlformats-officedocument.presentationml.slideshow',
 '.ppsx'), ('Microsoft Office - OOXML - Presentation Template',

'application/vnd.openxmlformats-officedocument.presentationml.template',
 '.potx'), ('Microsoft Office - OOXML - Spreadsheet',
 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',
 '.xlsx'), ('Microsoft Office - OOXML - Spreadsheet Teplate',
 'application/vnd.openxmlformats-officedocument.spreadsheetml.template',
 '.xltx'), ('Microsoft Office - OOXML - Word Document',
 'application/vnd.openxmlformats-officedocument.wordprocessingml.document',
 '.docx'), ('Microsoft Office - OOXML - Word Document Template',
 'application/vnd.openxmlformats-officedocument.wordprocessingml.template',
 '.dotx'), ('Microsoft Office Binder', 'application/x-msbinder', '.obd'),
 ('Microsoft Office System Release Theme', 'application/vnd.ms-officetheme',
 '.thmx'), ('Microsoft OneNote', 'application/onenote', '.onetoc'),
 ('Microsoft PlayReady Ecosystem', 'audio/vnd.ms-playready.media.pya',
 '.pya'), ('Microsoft PlayReady Ecosystem Video',
 'video/vnd.ms-playready.media.pyv', '.pyv'), ('Microsoft PowerPoint',
 'application/vnd.ms-powerpoint', '.ppt'), ('Microsoft PowerPoint - Add-in
 file', 'application/vnd.ms-powerpoint.addin.macroenabled.12', '.ppam'),
 ('Microsoft PowerPoint - Macro-Enabled Open XML Slide',
 'application/vnd.ms-powerpoint.slide.macroenabled.12', '.sldm'), ('Microsoft
 PowerPoint - Macro-Enabled Presentation File',
 'application/vnd.ms-powerpoint.presentation.macroenabled.12', '.pptm'),
 ('Microsoft PowerPoint - Macro-Enabled Slide Show File',
 'application/vnd.ms-powerpoint.slideshow.macroenabled.12', '.ppsm'),
 ('Microsoft Project', 'application/vnd.ms-project', '.mpp'), ('Microsoft
 Publisher', 'application/x-mspublisher', '.pub'), ('Microsoft Schedule+',
 'application/x-msschedule', '.scd'), ('Microsoft Silverlight',
 'application/x-silverlight-app', '.xap'), ('Microsoft Trust UI Provider -
 Certificate Trust Link', 'application/vnd.ms-pki.stl', '.stl'), ('Microsoft
 Trust UI Provider - Security Catalog', 'application/vnd.ms-pki.seccat',
 '.cat'), ('Microsoft Visio', 'application/vnd.visio', '.vsd'), ('Microsoft
 Windows Media', 'video/x-ms-wm', '.wm'), ('Microsoft Windows Media Audio',
 'audio/x-ms-wma', '.wma'), ('Microsoft Windows Media Audio Redirector',
 'audio/x-ms-wax', '.wax'), ('Microsoft Windows Media Audio/Video Playlist',
 'video/x-ms-wmx', '.wmx'), ('Microsoft Windows Media Player Download
 Package', 'application/x-ms-wmd', '.wmd'), ('Microsoft Windows Media Player
 Playlist', 'application/vnd.ms-wpl', '.wpl'), ('Microsoft Windows Media
 Player Skin Package', 'application/x-ms-wmz', '.wmz'), ('Microsoft Windows
 Media Video', 'video/x-ms-wmv', '.wmv'), ('Microsoft Windows Media Video
 Playlist', 'video/x-ms-wvx', '.wvx'), ('Microsoft Windows Metafile',
 'application/x-msmetafile', '.wmf'), ('Microsoft Windows Terminal Services',
 'application/x-msterminal', '.trm'), ('Microsoft Word', 'application/msword',
 '.doc'), ('Microsoft Wordpad', 'application/x-mswrite', '.wri'), ('Microsoft
 Works', 'application/vnd.ms-works', '.wps'), ('Microsoft XAML Browser
 Application', 'application/x-ms-xbap', '.xbap'), ('Microsoft XML Paper
 Specification', 'application/vnd.ms-xpsdocument', '.xps'), ('MIDI - Musical
 Instrument Digital Interface', 'audio/midi', '.mid'), ('MiniPay',

'application/vnd.ibm.minipay', '.mpy'), ('MO:DCA-P',
 'application/vnd.ibm.modcap', '.afp'), ('Mobile Information Device Profile',
 'application/vnd.jcp.javame.midlet-rms', '.rms'), ('MobileTV',
 'application/vnd.tmobile-livetv', '.tmo'), ('Mobipocket',
 'application/x-mobipocket-ebook', '.prc'), ('Mobius Management Systems -
 Basket file', 'application/vnd.mobius.mbk', '.mbk'), ('Mobius Management
 Systems - Distribution Database', 'application/vnd.mobius.dis', '.dis'),
 ('Mobius Management Systems - Policy Definition Language File',
 'application/vnd.mobius.plc', '.plc'), ('Mobius Management Systems - Query
 File', 'application/vnd.mobius.mqy', '.mqy'), ('Mobius Management Systems -
 Script Language', 'application/vnd.mobius.msl', '.msl'), ('Mobius Management
 Systems - Topic Index File', 'application/vnd.mobius.txf', '.txf'), ('Mobius
 Management Systems - UniversalArchive', 'application/vnd.mobius.daf',
 '.daf'), ('mod_fly / fly.cgi', 'text/vnd.fly', '.fly'), ('Mophun
 Certificate', 'application/vnd.mophun.certificate', '.mpc'), ('Mophun VM',
 'application/vnd.mophun.application', '.mpn'), ('Motion JPEG 2000',
 'video/mj2', '.mj2'), ('MPEG Audio', 'audio/mpeg', '.mpga'), ('MPEG Url',
 'video/vnd.mpegurl', '.mxu'), ('MPEG Video', 'video/mpeg', '.mpeg'),
 ('MPEG-21', 'application/mp21', '.m21'), ('MPEG-4 Audio', 'audio/mp4',
 '.mp4a'), ('MPEG-4 Video', 'video/mp4', '.mp4'), ('MPEG4', 'application/mp4',
 '.mp4'), ('Multimedia Playlist Unicode', 'application/vnd.apple.mpegurl',
 '.m3u8'), ('MUsical Score Interpreted Code Invented for the ASCII designation
 of Notation', 'application/vnd.musician', '.mus'), ('Muvee Automatic Video
 Editing', 'application/vnd.muvee.style', '.msty'), ('MXML',
 'application/xv+xml', '.mxml'), ('N-Gage Game Data',
 'application/vnd.nokia.n-gage.data', '.ngdat'), ('N-Gage Game Installer',
 'application/vnd.nokia.n-gage.symbian.install', '.n-gage'), ('Navigation
 Control file for XML (for ePub)', 'application/x-dtbnex+xml', '.ncx'),
 ('Network Common Data Form (NetCDF)', 'application/x-netcdf', '.nc'),
 ('neuroLanguage', 'application/vnd.neurolanguage.nlu', '.nlu'), ('New Moon
 Liftoff/DNA', 'application/vnd.dna', '.dna'), ('NobleNet Directory',
 'application/vnd.noblenet-directory', '.nnd'), ('NobleNet Sealer',
 'application/vnd.noblenet-sealer', '.nns'), ('NobleNet Web',
 'application/vnd.noblenet-web', '.nnw'), ('Nokia Radio Application - Preset',
 'application/vnd.nokia.radio-preset', '.rpst'), ('Nokia Radio Application -
 Preset', 'application/vnd.nokia.radio-presets', '.rpss'), ('Notation3',
 'text/n3', '.n3'), ('Novadigm's RADIA and EDM products',
 'application/vnd.novadigm.edm', '.edm'), ('Novadigm's RADIA and EDM
 products', 'application/vnd.novadigm.edx', '.edx'), ('Novadigm's RADIA and
 EDM products', 'application/vnd.novadigm.ext', '.ext'), ('NpGraphIt',
 'application/vnd.flographit', '.gph'), ('Nuera ECELP 4800',
 'audio/vnd.nuera.ecelp4800', '.ecelp4800'), ('Nuera ECELP 7470',
 'audio/vnd.nuera.ecelp7470', '.ecelp7470'), ('Nuera ECELP 9600',
 'audio/vnd.nuera.ecelp9600', '.ecelp9600'), ('Office Document Architecture',
 'application/oda', '.oda'), ('Ogg', 'application/ogg', '.ogx'), ('Ogg Audio',
 'audio/ogg', '.oga'), ('Ogg Video', 'video/ogg', '.ogv'), ('OMA Download

Agents', 'application/vnd.oma.dd2+xml', '.dd2'), ('Open Document Text Web',
 'application/vnd.oasis.opendocument.text-web', '.oth'), ('Open eBook
 Publication Structure', 'application/oebps-package+xml', '.opf'), ('Open
 Financial Exchange', 'application/vnd.intu.qbo', '.qbo'), ('Open Office
 Extension', 'application/vnd.openofficeorg.extension', '.oxt'), ('Open Score
 Format', 'application/vnd.yamaha.openscoreformat', '.osf'), ('Open Web Media
 Project - Audio', 'audio/webm', '.weba'), ('Open Web Media Project - Video',
 'video/webm', '.webm'), ('OpenDocument Chart',
 'application/vnd.oasis.opendocument.chart', '.odc'), ('OpenDocument Chart
 Template', 'application/vnd.oasis.opendocument.chart-template', '.otc'),
 ('OpenDocument Database', 'application/vnd.oasis.opendocument.database',
 '.odb'), ('OpenDocument Formula',
 'application/vnd.oasis.opendocument.formula', '.odf'), ('OpenDocument Formula
 Template', 'application/vnd.oasis.opendocument.formula-template', '.odft'),
 ('OpenDocument Graphics', 'application/vnd.oasis.opendocument.graphics',
 '.odg'), ('OpenDocument Graphics Template',
 'application/vnd.oasis.opendocument.graphics-template', '.otg'),
 ('OpenDocument Image', 'application/vnd.oasis.opendocument.image', '.odi'),
 ('OpenDocument Image Template',
 'application/vnd.oasis.opendocument.image-template', '.oti'), ('OpenDocument
 Presentation', 'application/vnd.oasis.opendocument.presentation', '.odp'),
 ('OpenDocument Presentation Template',
 'application/vnd.oasis.opendocument.presentation-template', '.otp'),
 ('OpenDocument Spreadsheet',
 'application/vnd.oasis.opendocument.spreadsheet', '.ods'), ('OpenDocument
 Spreadsheet Template',
 'application/vnd.oasis.opendocument.spreadsheet-template', '.ots'),
 ('OpenDocument Text', 'application/vnd.oasis.opendocument.text', '.odt'),
 ('OpenDocument Text Master',
 'application/vnd.oasis.opendocument.text-master', '.odm'), ('OpenDocument
 Text Template', 'application/vnd.oasis.opendocument.text-template', '.ott'),
 ('OpenGL Textures (KTX)', 'image/ktx', '.ktx'), ('OpenOffice - Calc
 (Spreadsheet)', 'application/vnd.sun.xml.calc', '.sxc'), ('OpenOffice - Calc
 Template (Spreadsheet)', 'application/vnd.sun.xml.calc.template', '.stc'),
 ('OpenOffice - Draw (Graphics)', 'application/vnd.sun.xml.draw', '.sxd'),
 ('OpenOffice - Draw Template (Graphics)',
 'application/vnd.sun.xml.draw.template', '.std'), ('OpenOffice - Impress
 (Presentation)', 'application/vnd.sun.xml.impress', '.sxi'), ('OpenOffice -
 Impress Template (Presentation)', 'application/vnd.sun.xml.impress.template',
 '.sti'), ('OpenOffice - Math (Formula)', 'application/vnd.sun.xml.math',
 '.sxm'), ('OpenOffice - Writer (Text - HTML)',
 'application/vnd.sun.xml.writer', '.sxw'), ('OpenOffice - Writer (Text -
 HTML)', 'application/vnd.sun.xml.writer.global', '.sxc'), ('OpenOffice -
 Writer Template (Text - HTML)', 'application/vnd.sun.xml.writer.template',
 '.stw'), ('OpenType Font File', 'application/x-font-otf', '.otf'), ('OSFPVG',
 'application/vnd.yamaha.openscoreformat.osfpvg+xml', '.osfpvg'), ('OSGi

Deployment Package', 'application/vnd.osgi.dp', '.dp'), ('PalmOS Data',
 'application/vnd.palm', '.pdb'), ('Pascal Source File', 'text/x-pascal',
 '.p'), ('PawaaFILE', 'application/vnd.pawaafile', '.paw'), ('PCL 6 Enhanced
 (Formerly PCL XL)', 'application/vnd.hp-pclxl', '.pclxl'), ('Pcset eFIF File',
 'application/vnd.picset', '.efif'), ('PCX Image', 'image/x-pcx', '.pcx'),
 ('Photoshop Document', 'image/vnd.adobe.photoshop', '.psd'), ('PICS Rules',
 'application/pics-rules', '.prf'), ('PICT Image', 'image/x-pict', '.pic'),
 ('pIRch', 'application/x-chat', '.chat'), ('PKCS #10 - Certification Request
 Standard', 'application/pkcs10', '.p10'), ('PKCS #12 - Personal Information
 Exchange Syntax Standard', 'application/x-pkcs12', '.p12'), ('PKCS #7 -
 Cryptographic Message Syntax Standard', 'application/pkcs7-mime', '.p7m'),
 ('PKCS #7 - Cryptographic Message Syntax Standard',
 'application/pkcs7-signature', '.p7s'), ('PKCS #7 - Cryptographic Message
 Syntax Standard (Certificate Request Response)',
 'application/x-pkcs7-certreqresp', '.p7r'), ('PKCS #7 - Cryptographic Message
 Syntax Standard (Certificates)', 'application/x-pkcs7-certificates', '.p7b'),
 ('PKCS #8 - Private-Key Information Syntax Standard', 'application/pkcs8',
 '.p8'), ('PocketLearn Viewers', 'application/vnd.pocketlearn', '.plf'),
 ('Portable Anymap Image', 'image/x-portable-anymap', '.pnm'), ('Portable
 Bitmap Format', 'image/x-portable-bitmap', '.pbm'), ('Portable Compiled
 Format', 'application/x-font-pcf', '.pcf'), ('Portable Font Resource',
 'application/font-tdpfr', '.pfr'), ('Portable Game Notation (Chess Games)',
 'application/x-chess-pgn', '.pgn'), ('Portable Graymap Format',
 'image/x-portable-graymap', '.pgm'), ('Portable Network Graphics (PNG)',
 'image/png', '.png'), ('Portable Pixmap Format', 'image/x-portable-pixmap',
 '.ppm'), ('Portable Symmetric Key Container', 'application/pskc+xml',
 '.pskcxml'), ('PosML', 'application/vnd.ctc-posml', '.pml'), ('PostScript',
 'application/postscript', '.ai'), ('PostScript Fonts',
 'application/x-font-type1', '.pfa'), ('PowerBuilder',
 'application/vnd.powerbuilder6', '.pbd'), ('Pretty Good Privacy',
 'application/pgp-encrypted', ''), ('Pretty Good Privacy - Signature',
 'application/pgp-signature', '.pgp'), ('Preview Systems ZipLock/VBox',
 'application/vnd.previewsystems.box', '.box'), ('Princeton Video Image',
 'application/vnd.pvi.ptid1', '.ptid'), ('Pronunciation Lexicon
 Specification', 'application/pls+xml', '.pls'), ('Proprietary P&G Standard
 Reporting System', 'application/vnd.pg.format', '.str'), ('Proprietary P&G
 Standard Reporting System', 'application/vnd.pg.osasli', '.ei6'), ('PRS Lines
 Tag', 'text/prs.lines.tag', '.dsc'), ('PSF Fonts',
 'application/x-font-linux-psf', '.psf'), ('PubliShare Objects',
 'application/vnd.publishare-delta-tree', '.qps'), ('Qualcomm's Plaza Mobile
 Internet', 'application/vnd.pmi.widget', '.wg'), ('QuarkXpress',
 'application/vnd.quark.quarkxpress', '.qxd'), ('QUASS Stream Player',
 'application/vnd.epson.esf', '.esf'), ('QUASS Stream Player',
 'application/vnd.epson.msf', '.msf'), ('QUASS Stream Player',
 'application/vnd.epson.ssf', '.ssf'), ('QuickAnime Player',
 'application/vnd.epson.quickanime', '.qam'), ('Quicken',

'application/vnd.intu.qfx', '.qfx'), ('Quicktime Video', 'video/quicktime',
 '.qt'), ('RAR Archive', 'application/x-rar-compressed', '.rar'), ('Real Audio
 Sound', 'audio/x-pn-realaudio', '.ram'), ('Real Audio Sound',
 'audio/x-pn-realaudio-plugin', '.rmp'), ('Really Simple Discovery',
 'application/rsd+xml', '.rsd'), ('RealMedia', 'application/vnd.rn-realmedia',
 '.rm'), ('RealVNC', 'application/vnd.realvnc.bed', '.bed'), ('Recordare
 Applications', 'application/vnd.recordare.musicxml', '.mxl'), ('Recordare
 Applications', 'application/vnd.recordare.musicxml+xml', '.musicxml'),
 ('Relax NG Compact Syntax', 'application/relax-ng-compact-syntax', '.rnc'),
 ('RemoteDocs R-Viewer', 'application/vnd.data-vision.rdz', '.rdz'),
 ('Resource Description Framework', 'application/rdf+xml', '.rdf'),
 ('RetroPlatform Player', 'application/vnd.cloanto.rp9', '.rp9'), ('RhymBox',
 'application/vnd.jisp', '.jisp'), ('Rich Text Format', 'application/rtf',
 '.rtf'), ('Rich Text Format (RTF)', 'text/richtext', '.rtx'), ('ROUTE 66
 Location Based Services', 'application/vnd.route66.link66+xml', '.link66'),
 ('RSS - Really Simple Syndication', 'application/rss+xml', '.rss, .xml'), ('S
 Hexdump Format', 'application/shf+xml', '.shf'), ('SailingTracker',
 'application/vnd.sailingtracker.track', '.st'), ('Scalable Vector Graphics
 (SVG)', 'image/svg+xml', '.svg'), ('ScheduleUs',
 'application/vnd.sus-calendar', '.sus'), ('Search/Retrieve via URL Response
 Format', 'application/sru+xml', '.sru'), ('Secure Electronic Transaction -
 Payment', 'application/set-payment-initiation', '.setpay'), ('Secure
 Electronic Transaction - Registration',
 'application/set-registration-initiation', '.setreg'), ('Secured eMail',
 'application/vnd.sema', '.sema'), ('Secured eMail', 'application/vnd.semd',
 '.semd'), ('Secured eMail', 'application/vnd.semf', '.semf'), ('SeeMail',
 'application/vnd.seemail', '.see'), ('Server Normal Format',
 'application/x-font-snf', '.snf'), ('Server-Based Certificate Validation
 Protocol - Validation Policies - Request', 'application/scvp-vp-request',
 '.spq'), ('Server-Based Certificate Validation Protocol - Validation Policies
 - Response', 'application/scvp-vp-response', '.spp'), ('Server-Based
 Certificate Validation Protocol - Validation Request',
 'application/scvp-cv-request', '.scq'), ('Server-Based Certificate Validation
 Protocol - Validation Response', 'application/scvp-cv-response', '.scs'),
 ('Session Description Protocol', 'application/sdp', '.sdp'), ('Setext',
 'text/x-setext', '.etx'), ('SGI Movie', 'video/x-sgi-movie', '.movie'),
 ('Shana Informed Filler', 'application/vnd.shana.informed.formdata', '.ifm'),
 ('Shana Informed Filler', 'application/vnd.shana.informed.formtemplate',
 '.itp'), ('Shana Informed Filler',
 'application/vnd.shana.informed.interchange', '.iif'), ('Shana Informed
 Filler', 'application/vnd.shana.informed.package', '.ipk'), ('Sharing
 Transaction Fraud Data', 'application/thraud+xml', '.tfi'), ('Shell Archive',
 'application/x-shar', '.shar'), ('Silicon Graphics RGB Bitmap',
 'image/x-rgb', '.rgb'), ('SimpleAnimeLite Player',
 'application/vnd.epson.salt', '.slt'), ('Simply Accounting',
 'application/vnd.accpac.simply.aso', '.aso'), ('Simply Accounting - Data

Import', 'application/vnd.accpac.simply.imp', '.imp'), ('SimTech MindMapper',
 'application/vnd.simtech-mindmapper', '.twd'), ('Sixth Floor Media -
 CommonSpace', 'application/vnd.commonspace', '.csp'), ('SMAF Audio',
 'application/vnd.yamaha.smaf-audio', '.saf'), ('SMAF File',
 'application/vnd.smaf', '.mmf'), ('SMAF Phrase',
 'application/vnd.yamaha.smaf-phrase', '.spf'), ('SMART Technologies Apps',
 'application/vnd.smart.teacher', '.teacher'), ('SourceView Document',
 'application/vnd.svd', '.svd'), ('SPARQL - Query',
 'application/sparql-query', '.rq'), ('SPARQL - Results',
 'application/sparql-results+xml', '.srx'), ('Speech Recognition Grammar
 Specification', 'application/srgs', '.gram'), ('Speech Recognition Grammar
 Specification - XML', 'application/srgs+xml', '.grxml'), ('Speech Synthesis
 Markup Language', 'application/ssml+xml', '.ssml'), ('SSEYO Koan Play File',
 'application/vnd.koan', '.skp'), ('Standard Generalized Markup Language
 (SGML)', 'text/sgml', '.sgml'), ('StarOffice - Calc',
 'application/vnd.stardivision.calc', '.sdc'), ('StarOffice - Draw',
 'application/vnd.stardivision.draw', '.sda'), ('StarOffice - Impress',
 'application/vnd.stardivision.impress', '.sdd'), ('StarOffice - Math',
 'application/vnd.stardivision.math', '.smf'), ('StarOffice - Writer',
 'application/vnd.stardivision.writer', '.sdw'), ('StarOffice - Writer
 (Global)', 'application/vnd.stardivision.writer-global', '.sgl'),
 ('StepMania', 'application/vnd.stepmania.stepchart', '.sm'), ('Stuffit
 Archive', 'application/x-stuffit', '.sit'), ('Stuffit Archive',
 'application/x-stuffitx', '.sitx'), ('SudokuMagic',
 'application/vnd.solent.sdkm+xml', '.sdkm'), ('Sugar Linux Application
 Bundle', 'application/vnd.olpc-sugar', '.xo'), ('Sun Audio - Au file format',
 'audio/basic', '.au'), ('SundaHus WQ', 'application/vnd.wqd', '.wqd'),
 ('Symbian Install Package', 'application/vnd.symbian.install', '.sis'),
 ('Synchronized Multimedia Integration Language', 'application/smil+xml',
 '.smi'), ('SyncML', 'application/vnd.syncml+xml', '.xsm'), ('SyncML - Device
 Management', 'application/vnd.syncml.dm+wbxml', '.bdm'), ('SyncML - Device
 Management', 'application/vnd.syncml.dm+xml', '.xdm'), ('System V Release 4
 CPIO Archive', 'application/x-sv4cpio', '.sv4cpio'), ('System V Release 4
 CPIO Checksum Data', 'application/x-sv4crc', '.sv4crc'), ('Systems Biology
 Markup Language', 'application/sbml+xml', '.sbml'), ('Tab Separated Values',
 'text/tab-separated-values', '.tsv'), ('Tagged Image File Format',
 'image/tiff', '.tiff'), ('Tao Intent',
 'application/vnd.tao.intent-module-archive', '.tao'), ('Tar File (Tape
 Archive)', 'application/x-tar', '.tar'), ('Tcl Script', 'application/x-tcl',
 '.tcl'), ('TeX', 'application/x-tex', '.tex'), ('TeX Font Metric',
 'application/x-tex-tfm', '.tfm'), ('Text Encoding and Interchange',
 'application/tei+xml', '.tei'), ('Text File', 'text/plain', '.txt'), ('TIBCO
 Spotfire', 'application/vnd.spotfire.dxp', '.dxp'), ('TIBCO Spotfire',
 'application/vnd.spotfire.sfs', '.sfs'), ('Time Stamped Data Envelope',
 'application/timestamped-data', '.tsd'), ('TRI Systems Config',
 'application/vnd.trid.tpt', '.tpt'), ('Triscape Map Explorer',

'application/vnd.triscape.mxs', '.mxs'), ('troff', 'text/troff', '.t'),
 ('True BASIC', 'application/vnd.trueapp', '.tra'), ('TrueType Font',
 'application/x-font-ttf', '.ttf'), ('Turtle (Terse RDF Triple Language)',
 'text/turtle', '.ttl'), ('UMAJIN', 'application/vnd.umajin', '.umj'),
 ('Unique Object Markup Language', 'application/vnd.uoml+xml', '.uoml'),
 ('Unity 3d', 'application/vnd.unity', '.unityweb'), ('Universal Forms
 Description Language', 'application/vnd.ufdl', '.ufd'), ('URI Resolution
 Services', 'text/uri-list', '.uri'), ('User Interface Quartz - Theme
 (Symbian)', 'application/vnd.uiq.theme', '.utz'), ('Ustar (Uniform Standard
 Tape Archive)', 'application/x-ustar', '.ustar'), ('UUEncode',
 'text/x-uencode', '.uu'), ('vCalendar', 'text/x-vcalendar', '.vcs'),
 ('vCard', 'text/x-vcard', '.vcf'), ('Video CD', 'application/x-cdlink',
 '.vcd'), ('Viewport+', 'application/vnd.vsf', '.vsf'), ('Virtual Reality
 Modeling Language', 'model/vrml', '.wrl'), ('VirtualCatalog',
 'application/vnd.vcx', '.vcx'), ('Virtue MTS', 'model/vnd.mts', '.mts'),
 ('Virtue VTU', 'model/vnd.vtu', '.vtu'), ('Visionary',
 'application/vnd.visionary', '.vis'), ('Vivo', 'video/vnd.vivo', '.viv'),
 ('Voice Browser Call Control', 'application/ccxml+xml', '.ccxml'),
 ('VoiceXML', 'application/voicexml+xml', '.vxml'), ('WAIS Source',
 'application/x-wais-source', '.src'), ('WAP Binary XML (WBXML)',
 'application/vnd.wap.wbxml', '.wbxml'), ('WAP Bitamp (WBMP)',
 'image/vnd.wap.wbmp', '.wbmp'), ('Waveform Audio File Format (WAV)',
 'audio/x-wav', '.wav'), ('Web Distributed Authoring and Versioning',
 'application/davmount+xml', '.davmount'), ('Web Open Font Format',
 'application/x-font-woff', '.woff'), ('Web Services Policy',
 'application/wspolicy+xml', '.wspolicy'), ('WebP Image', 'image/webp',
 '.webp'), ('WebTurbo', 'application/vnd.webturbo', '.wtb'), ('Widget
 Packaging and XML Configuration', 'application/widget', '.wgt'), ('WinHelp',
 'application/winhelp', '.hlp'), ('Wireless Markup Language (WML)',
 'text/vnd.wap.wml', '.wml'), ('Wireless Markup Language Script (WMLScript)',
 'text/vnd.wap.wmlscript', '.wmls'), ('WMLScript',
 'application/vnd.wap.wmlscriptc', '.wmlsc'), ('Wordperfect',
 'application/vnd.wordperfect', '.wpd'), ('Worldtalk',
 'application/vnd.wt.stf', '.stf'), ('WSDL - Web Services Description
 Language', 'application/wsdl+xml', '.wsdl'), ('X BitMap', 'image/x-xbitmap',
 '.xbm'), ('X PixMap', 'image/x-pixmap', '.xpm'), ('X Window Dump',
 'image/x-xwindowdump', '.xwd'), ('X.509 Certificate',
 'application/x-x509-ca-cert', '.der'), ('Xfig', 'application/x-xfig',
 '.fig'), ('XHTML - The Extensible HyperText Markup Language',
 'application/xhtml+xml', '.xhtml'), ('XML - Extensible Markup Language',
 'application/xml', '.xml'), ('XML Configuration Access Protocol - XCAP Diff',
 'application/xcap-diff+xml', '.xdf'), ('XML Encryption Syntax and
 Processing', 'application/xenc+xml', '.xenc'), ('XML Patch Framework',
 'application/patch-ops-error+xml', '.xer'), ('XML Resource Lists',
 'application/resource-lists+xml', '.rl'), ('XML Resource Lists',
 'application/rls-services+xml', '.rs'), ('XML Resource Lists Diff',

```
'application/resource-lists-diff+xml', '.rld'), ('XML Transformations',
'application/xslt+xml', '.xslt'), ('XML-Binary Optimized Packaging',
'application/xop+xml', '.xop'), ('XPInstall - Mozilla',
'application/x-xpinstall', '.xpi'), ('XSPF - XML Shareable Playlist Format',
'application/xspf+xml', '.xspf'), ('XUL - XML User Interface Language',
'application/vnd.mozilla.xul+xml', '.xul'), ('XYZ File Format',
'chemical/x-xyz', '.xyz'), ('YANG Data Modeling Language',
'application/yang', '.yang'), ('YIN (YANG - XML)', 'application/yin+xml',
'.yin'), ('Z.U.L. Geometry', 'application/vnd.zul', '.zir'), ('Zip Archive',
'application/zip', '.zip'), ('ZVUE Media Manager',
'application/vnd.handheld-entertainment+xml', '.zmm'), ('Zzazz Deck',
'application/vnd.zzazz.deck+xml', '.zaz') );
```

6.3 Variables

BROOK_HTTP_DAEMON_DESCRIPTION _____

Declaration BROOK_HTTP_DAEMON_DESCRIPTION: string = 'Brook HTTP daemon';

BROOK_HTTP_DAEMON_DISPLAYNAME _____

Declaration BROOK_HTTP_DAEMON_DISPLAYNAME: string = 'Brook daemon';

BROOK_HTTP_DAEMON_NAME _____

Declaration BROOK_HTTP_DAEMON_NAME: string = 'BrookHTTPTaemon';

BROOK_HTTP_DAEMON_CLASSNAME _____

Declaration BROOK_HTTP_DAEMON_CLASSNAME: string = 'TBrookHTTPTaemon';

Chapter 7

Unit BrookDataBase

7.1 Description

Brook DataBase unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

7.2 Uses

- BrookClasses(4)
- BrookException(11)
- BrookConsts(6)
- BrookDBConsts(9)
- BrookMessages(16)
- BrookConfigurator(5)
- Classes
- SysUtils

7.3 Overview

EBrookDataBase Class Handles exceptions for TBrookDataBase(7.4).

EBrookDataBases Class Handles exceptions for **TBrookDataBases**(7.4).

TBrookDataBase Class Offers general abstract features for databases handling.

TBrookDataBases Class Registers and configure database classes.

7.4 Classes, Interfaces, Objects and Records

EBrookDataBase Class

Hierarchy

EBrookDataBase > EBrook(11.4) > Exception

Description

Handles exceptions for **TBrookDataBase**(7.4).

EBrookDataBases Class

Hierarchy

EBrookDataBases > EBrook(11.4) > Exception

Description

Handles exceptions for **TBrookDataBases**(7.4).

TBrookDataBase Class

Hierarchy

TBrookDataBase > TBrookObject(4.4) > TObject

Description

Offers general abstract features for databases handling.

Properties

Connected	<code>public property Connected: Boolean read GetConnected;</code> Checks if the database is connected.
Driver	<code>public property Driver: string read GetDriver write SetDriver;</code> Defines the driver name. In Zeos driver is know as "Protocol"; in Unidac, "Provider", etc.
Database	<code>public property Database: string read GetDatabase write SetDatabase;</code> Defines the database name or the database path.

User `public property User: string read GetUser write SetUser;`
 Defines the database user.

Password `public property Password: string read GetPassword write SetPassword;`
 Defines the database password.

Host `public property Host: string read GetHost write SetHost;`
 Defines the database host.

Port `public property Port: Integer read GetPort write SetPort;`
 Defines the database port.

Connection `public property Connection: TObject read GetConnection;`
 Is the instance of the database broker.

Methods

Init

Declaration `public constructor Init; virtual;`

Description Creates an instance of a TBrookDataBase(7.4) class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of TBrookDataBase(7.4) class.

Create

Declaration `public class function Create: TBrookDataBase;`

Description Creates an instance of a TBrookDataBase(7.4) class.

Register

Declaration `public class procedure Register;`

Description Register the broker class.

Unregister

Declaration `public class procedure Unregister;`

Description Unregister the broker class.

AddObject

Declaration public procedure AddObject(AObject: TObject);

Description Adds objects that will be automatically freed when the database object is destroyed.

RemoveObject

Declaration public procedure RemoveObject(AObject: TObject);

Description Removes objects that would be automatically freed when the database object is destroyed.

GetLibrary

Declaration public class function GetLibrary: string; virtual; abstract;

Description Get the broker library name, exemple: SQLdb, Zeos, UniDAC etc.

Connect

Declaration public procedure Connect; virtual; abstract;

Description Connects to the data base.

Disconnect

Declaration public procedure Disconnect; virtual; abstract;

Description Disconnects from the data base.

InTransaction

Declaration public function InTransaction: Boolean; virtual; abstract;

Description Checks if a transaction is active.

StartTransaction

Declaration public procedure StartTransaction; virtual; abstract;

Description Starts a transaction.

Commit

Declaration public procedure Commit; virtual; abstract;

Description Commits a transaction.

Rollback

Declaration `public procedure Rollback; virtual; abstract;`

Description Rollbacks a transaction.

TBrookDataBases Class

Hierarchy

TBrookDataBases > TBrookObject(4.4) > TObject

Description

Registers and configure database classes.

Properties

Current `public property Current: TBrookDataBase read FCurrent write FCurrent;`
The current database item.

Items `public property Items[const AIndex: Integer]: TBrookDataBaseClass read GetItems write SetItems;`
The list of database items.

Configurator `public property Configurator: TBrookConfigurator read FConfigurator;`
A configurator for database items.

Methods

Create

Declaration `public constructor Create;`

Description Creates an instance of a TBrookDataBases(7.4) class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of TBrookDataBases(7.4) class.

RegisterService

Declaration `public class procedure RegisterService;`

Description Registers the service provided by this class.

UnregisterService

Declaration `public class procedure UnregisterService;`

Description Unregisters the service provided by this class.

Service

Declaration `public class function Service: TBrookDataBases;`

Description Return a instance of this class.

CreateDataBase

Declaration `public function CreateDataBase: TBrookDataBase;`

Description Creates an database item.

FreeCurrent

Declaration `public procedure FreeCurrent;`

Description Frees the current database item.

Count

Declaration `public function Count: Integer;`

Description Returns the number of registered databases.

Find

Declaration `public function Find(const ALibrary: string): TBrookDataBaseClass;`

Description Finds a database item by its library name.

ItemByLibrary

Declaration `public function ItemByLibrary(const ALibrary: string):
TBrookDataBaseClass;`

Description Returns a database item by its library name.

Add

Declaration `public procedure Add(AClass: TBrookDataBaseClass);`

Description Adds a database item.

Remove

Declaration `public procedure Remove(AClass: TBrookDataBaseClass);`

Description Removes a database item.

7.5 Types

TBrookDataBaseClass _____

Declaration `TBrookDataBaseClass = class of TBrookDataBase;`

Description Is a metaclass for TBrookDataBase(7.4) class.

TBrookDataBasesClass _____

Declaration `TBrookDataBasesClass = class of TBrookDataBases;`

Description Is a metaclass for TBrookDataBases(7.4) class.

Chapter 8

Unit BrookDBAction

8.1 Description

Brook DB Action unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

8.2 Uses

- `BrookAction(1)`
- `BrookException(11)`
- `BrookDataBase(7)`
- `BrookTable(23)`
- `BrookUtils(24)`
- `BrookMessages(16)`
- `BrookConsts(6)`
- `DB`
- `Classes`
- `SysUtils`

8.3 Overview

EBrookDBAction Class Handles exceptions for TBrookDBAction(8.4).

TBrookCustomDBAction Class Custom class for manages HTTP requests and responses when data persistence is required.

TBrookDBAction Class Manages HTTP requests and responses when data persistence is required.

8.4 Classes, Interfaces, Objects and Records

EBrookDBAction Class

Hierarchy

EBrookDBAction > EBrook(11.4) > Exception

Description

Handles exceptions for TBrookDBAction(8.4).

TBrookCustomDBAction Class

Hierarchy

TBrookCustomDBAction > TBrookAction(1.4) > TBrookObject(4.4) > TObject

Description

Custom class for manages HTTP requests and responses when data persistence is required.

Methods

Register

Declaration `public class procedure Register(const ATableName, APattern: string; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table.

Register

Declaration `public class procedure Register(const ATableName, APattern: string; const AMethod: TBrookRequestMethod; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table.

Register

Declaration `public class procedure Register(const ATableName, APattern, AIgnoredFields: string; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table and defining the fields that will be ignored for persistence purposes.

Register

Declaration `public class procedure Register(const ATableName, APattern, AIgnoredFields: string; const AMethod: TBrookRequestMethod; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table and defining the fields that will be ignored for persistence purposes.

SetTableName

Declaration `public class procedure SetTableName(const ATableName: string);`

Description Defines the table name.

GetTableName

Declaration `public class function GetTableName: string;`

Description Returns the table name.

SetIgnoredFields

Declaration `public class procedure SetIgnoredFields(const AFields: string);`

Description Define the fields that will be ignored for persistence purposes.

GetIgnoredFields

Declaration `public class function GetIgnoredFields: string;`

Description Returns the fields were ignored for persistence purposes.

TBrookDBAction Class

Hierarchy

TBrookDBAction > TBrookCustomDBAction(8.4) > TBrookAction(1.4) > TBrookObject(4.4) > TObject

Description

Manages HTTP requests and responses when data persistence is required.

Properties

DataBase `public property DataBase: TBrookDataBase read GetDataBase write SetDataBase;`
Defines the current database or return it.

Table `public property Table: TBrookTable read GetTable write SetTable;`
Get de table linked to the action.

Methods

Create

Declaration `public constructor Create; override;`

Description Creates an instance of TBrookDBAction.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of TBrookDBAction(8.4) class.

8.5 Types

TBrookDBActionClass

Declaration `TBrookDBActionClass = class of TBrookDBAction;`

Description Is a metaclass for TBrookDBAction(8.4) class.

Chapter 9

Unit BrookDBConsts

9.1 Description

Brook DB Consts unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

9.2 Uses

- `BrookConsts(6)`

9.3 Constants

BROOK_FT_NULL _____

Declaration `BROOK_FT_NULL = 'null';`

BROOK_FT_STRING _____

Declaration `BROOK_FT_STRING = 'string';`

BROOK_FT_BOOLEAN _____

Declaration `BROOK_FT_BOOLEAN = 'boolean';`

BROOK_FT_DATE _____

Declaration `BROOK_FT_DATE = 'date';`

BROOK_FT_FLOAT _____

Declaration BROOK_FT_FLOAT = 'float';

BROOK_FT_INT _____

Declaration BROOK_FT_INT = 'int';

BROOK_SQL_SELECT_TOKEN _____

Declaration BROOK_SQL_SELECT_TOKEN = 'select';

BROOK_SQL_FROM_TOKEN _____

Declaration BROOK_SQL_FROM_TOKEN = 'from';

BROOK_SQL_WHERE_TOKEN _____

Declaration BROOK_SQL_WHERE_TOKEN = 'where';

BROOK_SQL_NOTHING_WHERE_TOKEN _____

Declaration BROOK_SQL_NOTHING_WHERE_TOKEN = BROOK_SQL_WHERE_TOKEN + SP + '1=2';

BROOK_SQL_ORDER_BY_TOKEN _____

Declaration BROOK_SQL_ORDER_BY_TOKEN = 'order by';

BROOK_SQL_INSERT_TOKEN _____

Declaration BROOK_SQL_INSERT_TOKEN = 'insert into';

BROOK_SQL_VALUES_TOKEN _____

Declaration BROOK_SQL_VALUES_TOKEN = 'values';

BROOK_SQL_UPDATE_TOKEN _____

Declaration BROOK_SQL_UPDATE_TOKEN = 'update';

BROOK_SQL_SET_TOKEN _____

Declaration BROOK_SQL_SET_TOKEN = 'set';

BROOK_SQL_DELETE_TOKEN _____

Declaration BROOK_SQL_DELETE_TOKEN = 'delete';

BROOK_SQL_EQ_PARAM_TOKEN _____

Declaration BROOK_SQL_EQ_PARAM_TOKEN = ' = ':';

BROOK_SQL_LOWER_TOKEN _____

Declaration BROOK_SQL_LOWER_TOKEN = 'lower';

BROOK_SQL_LIKE_TOKEN _____

Declaration BROOK_SQL_LIKE_TOKEN = 'like';

BROOK_SQL_AND_TOKEN _____

Declaration BROOK_SQL_AND_TOKEN = 'and';

9.4 Variables

BROOK_DEFAULT_LIBRARY_PARAM _____

Declaration BROOK_DEFAULT_LIBRARY_PARAM: string = 'library';

BROOK_DEFAULT_KEY_NAME _____

Declaration BROOK_DEFAULT_KEY_NAME: string = 'id';

Chapter 10

Unit BrookDBUtils

10.1 Description

Brook DB Utils unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

10.2 Uses

- `BrookConsts(6)`
- `DB`
- `SysUtils`
- `StrUtils`

10.3 Overview

`BrookDataSetToHTMLTable` Converts dataset into HTML table.

`BrookDataSetToHTML5Table` Converts dataset into HTML-5 table.

`BrookFieldDefsToHTMLForm` Converts FieldDefs into HTML form.

10.4 Functions and Procedures

BrookDataSetToHTMLTable

Declaration `function BrookDataSetToHTMLTable(ADataSet: TDataSet; const ATitles: array of string; const AClass: string = ES; const ABorderWidth: Integer = 1; const AUnavailableFields: TBrookUnavailableFields = []; ACallback: TBrookDataSetToHTML5TableCallback = nil): string;`

Description Converts dataset into HTML table.

BrookDataSetToHTML5Table

Declaration `function BrookDataSetToHTML5Table(ADataSet: TDataSet; const ATitles: array of string; const ACaption: string = ES; const AFoot: string = ES; const AClass: string = ES; const ABorderWidth: Integer = 1; const AUnavailableFields: TBrookUnavailableFields = []; ACallback: TBrookDataSetToHTML5TableCallback = nil): string;`

Description Converts dataset into HTML-5 table.

BrookFieldDefsToHTMLForm

Declaration `function BrookFieldDefsToHTMLForm(AFieldDefs: TFieldDefs; const AAction: string = '#'; const AMethod: string = 'post'; const ALineBreak: Boolean = False; const AClass: string = ES; const ALabelAbove: Boolean = False): string;`

Description Converts FieldDefs into HTML form.

10.5 Types

TBrookUnavailableFields

Declaration `TBrookUnavailableFields = set of TFieldType;`

Description Set of data types for BrookDataSetToHTMLTable and BrookDataSetToHTML5Table.

TBrookHTMLTableWritingState

Declaration `TBrookHTMLTableWritingState = (...);`

Description Informs the writing stage of the table.

Values wtHeadTR
wtHeadTD
wtBodyTR
wtBodyTD

TBrookDataSetToHTMLTableCallback ---

Declaration `TBrookDataSetToHTMLTableCallback = function(ADataset: TDataSet; const
AWritingType: TBrookHTMLTableWritingState; const APosition, AMax: Integer;
var AData: string): string of object;`

Description Callback for BrookDataSetToHTMLTable.

TBrookDataSetToHTML5TableCallback ---

Declaration `TBrookDataSetToHTML5TableCallback = TBrookDataSetToHTMLTableCallback;`

Description Callback for BrookDataSetToHTML5Table.

Chapter 11

Unit BrookException

11.1 Description

Brook Exception unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

11.2 Uses

- BrookConsts(6)
- SysUtils

11.3 Overview

EBrook Class Handles general exception in Brook.

EBrookHTTP404 Class Handles the exception for 404 error.

EBrookHTTP500 Class Handles the exception for 500 error.

11.4 Classes, Interfaces, Objects and Records

EBrook Class

Hierarchy

EBrook > Exception

Description

Handles general exception in Brook.

Methods

Create

Declaration `public constructor Create(AInstance: TObject; const AMsg: string); overload;`

Description Creates an instance of EBrook.

Create

Declaration `public constructor Create(AClass: TClass; const AMsg: string); overload;`

Description Creates an instance of EBrook.

Create

Declaration `public constructor Create(AName: string; const AMsg: string); overload;`

Description Creates an instance of EBrook.

CreateFmt

Declaration `public constructor CreateFmt(AInstance: TObject; const AMsg: string; const AArgs: array of const); overload;`

Description Creates an instance of EBrook with a formatted message.

CreateFmt

Declaration `public constructor CreateFmt(AClass: TClass; const AMsg: string; const AArgs: array of const); overload;`

Description Creates an instance of EBrook with a formatted message.

CreateFmt

Declaration `public constructor CreateFmt(AName: string; const AMsg: string; const AArgs: array of const); overload;`

Description Creates an instance of EBrook with a formatted message.

EBrookHTTP404 Class

Hierarchy

EBrookHTTP404 > EBrook(11.4) > Exception

Description

Handles the exception for 404 error.

EBrookHTTP500 Class

Hierarchy

EBrookHTTP500 > EBrook(11.4) > Exception

Description

Handles the exception for 500 error.

11.5 Types

EBrookClass

Declaration `EBrookClass = class of EBrook;`

Description Is a metaclass for EBrook(11.4) class.

Chapter 12

Unit BrookHTTPClient

12.1 Description

Brook HTTP Client unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

12.2 Uses

- `BrookClasses(4)`
- `BrookException(11)`
- `BrookMessages(16)`
- `BrookConsts(6)`
- `BrookHTTPConsts(13)`
- `Classes`
- `SysUtils`
- `FGL`

12.3 Overview

`EBrookHTTPDef` Class Handles exceptions for `TBrookHTTPDef(12.4)`.

EBrookHTTPDefs Class Handles exceptions for **TBrookHTTPDefs**(12.4).

EBrookHTTPClient Class Handles exceptions for **EBrookHTTPClient**(12.4).

TBrookHTTPResult record Information returned after a request.

TBrookHTTPDef Class Offers general abstract features for HTTP handling.

TBrookHTTPDefs Class Registers HTTP definitions.

TBrookHTTPClient Class Client to perform HTTP requests.

12.4 Classes, Interfaces, Objects and Records

EBrookHTTPDef Class

Hierarchy

EBrookHTTPDef > EBrook(11.4) > Exception

Description

Handles exceptions for **TBrookHTTPDef**(12.4).

EBrookHTTPDefs Class

Hierarchy

EBrookHTTPDefs > EBrook(11.4) > Exception

Description

Handles exceptions for **TBrookHTTPDefs**(12.4).

EBrookHTTPClient Class

Hierarchy

EBrookHTTPClient > EBrook(11.4) > Exception

Description

Handles exceptions for **EBrookHTTPClient**(12.4).

TBrookHTTPResult record

Description

Information returned after a request.

Fields

StatusCode `StatusCode: Integer;`
ReasonPhrase `ReasonPhrase: string;`
Header `Header: string;`
Content `Content: string;`

TBrookHTTPDef Class

Hierarchy

TBrookHTTPDef > TBrookObject(4.4) > TObject

Description

Offers general abstract features for HTTP handling.

Properties

ContentType `public property ContentType: string read GetContentType write SetContentType;`
Content type of sending data.

Contents `public property Contents: TStrings read GetContents;`
Strings received from the request.

Cookies `public property Cookies: TStrings read GetCookies;`
Set cookies to be sent and/or received from the server.

Document `public property Document: TStream read GetDocument;`
Document received from the request.

Headers `public property Headers: TStrings read GetHeaders;`
Request headers.

StatusCode `public property StatusCode: Integer read GetStatusCode;`
Result code after successful request.

ReasonPhrase `public property ReasonPhrase: string read GetReasonPhrase;`
Result text after successful request.

Method `public property Method: string read GetMethod write SetMethod;`
Method for requests.

Url `public property Url: string read GetUrl write SetUrl;`
URL that request is driven to.

Client `public property Client: TObject read GetClient;`
Is the instance of the HTTP client broker.

Methods

Create

Declaration `public constructor Create; virtual; abstract;`

Description Creates an instance of a `TBrookHTTPDef(12.4)` class.

Register

Declaration `public class procedure Register;`

Description Register the broker class.

Unregister

Declaration `public class procedure Unregister;`

Description Unregister the broker class.

GetLibrary

Declaration `public class function GetLibrary: string; virtual; abstract;`

Description Get the broker library name, for example: FCLWeb, Synapse, LNet, Indy etc.

AddHeader

Declaration `public procedure AddHeader(const AName, AValue: string); virtual; abstract;`

Description Adds header, replacing an existing one if it exists.

Request

Declaration `public function Request: Boolean; virtual; abstract;`

Description Sends request to server.

Get

Declaration `public class function Get(const AUrl: string; AResponse: TStream): Boolean; virtual; abstract;`

Description Sends request by a GET HTTP request method.

Post

Declaration `public class function Post(const AUrl: string; AResponse: TStream): Boolean; virtual; abstract;`

Description Sends request by a POST HTTP request method.

Put

Declaration `public class function Put(const AUrl: string; AResponse: TStream): Boolean; virtual; abstract;`

Description Sends request by a PUT HTTP request method.

Delete

Declaration `public class function Delete(const AUrl: string; AResponse: TStream): Boolean; virtual; abstract;`

Description Sends request by a DELETE HTTP request method.

Options

Declaration `public class function Options(const AUrl: string; AResponse: TStream): Boolean; virtual; abstract;`

Description Sends request by a OPTIONS HTTP request method.

Head

Declaration `public class function Head(const AUrl: string; AHeaders: TStrings): Boolean; virtual; abstract;`

Description Sends request by a HEAD HTTP request method.

PostForm

Declaration `public class function PostForm(const AUrl: string; AFormData, AResponse: TStream): Boolean; virtual; abstract;`

Description Sends request by a POST HTTP request method, passing a form-data as parameter.

PostForm

Declaration `public class function PostForm(const AUrl, AFormData: string; AResponse: TStream): Boolean; virtual; abstract;`

Description Sends request by a POST HTTP request method, passing a form-data as parameter.

PostFile

Declaration `public class function PostFile(const AUrl, AFieldName, AFileName: string; AFile, AResponse: TStream): Boolean; virtual; abstract;`

Description Sends request by a POST HTTP request method, passing a file as parameter.

PostFile

Declaration `public class function PostFile(const AUrl, AFieldName, AFileName: string; AResponse: TStream): Boolean; virtual; abstract;`

Description Sends request by a POST HTTP request method, passing a file as parameter.

TBrookHTTPDefs Class

Hierarchy

TBrookHTTPDefs > TFPGList

Description

Registers HTTP definitions.

Methods

Service

Declaration `public class function Service: TBrookHTTPDefs;`

Description Registers the service provided by this class.

Find

Declaration `public function Find(const ALibrary: string): TBrookHTTPDefClass;`

Description Finds a database item by its library name.

ItemByLibrary

Declaration `public function ItemByLibrary(const ALibrary: string): TBrookHTTPDefClass;`

Description Returns a database item by its library name.

TBrookHTTPClient Class

Hierarchy

TBrookHTTPClient > TBrookObject(4.4) > TObject

Description

Client to perform HTTP requests.

Methods

Create

Declaration `public constructor Create(const ALibrary: string);`

Description Creates an instance of a `TBrookHTTPClient(12.4)` class.

SetLibrary

Declaration `public class procedure SetLibrary(const ALibrary: string);`

Description Specifies the library to be used by this class.

Prepare

Declaration `public class procedure Prepare(out AHttp: TBrookHTTPDef);`

Description Prepares an instance of `TBrookHTTPDef`.

Get

Declaration `public class function Get(const AUrl: string; AResponse: TStream): Boolean;`

Description Sends request by a GET HTTP request method.

Post

Declaration `public class function Post(const AUrl: string; AResponse: TStream): Boolean;`

Description Sends request by a POST HTTP request method.

Put

Declaration `public class function Put(const AUrl: string; AResponse: TStream): Boolean;`

Description Sends request by a PUT HTTP request method.

Delete

Declaration `public class function Delete(const AUrl: string; AResponse: TStream): Boolean;`

Description Sends request by a DELETE HTTP request method.

Options

Declaration `public class function Options(const AUrl: string; AResponse: TStream): Boolean;`

Description Sends request by a OPTIONS HTTP request method.

Head

Declaration `public class function Head(const AUrl: string; AHeaders: TStrings): Boolean;`

Description Sends request by a HEAD HTTP request method.

PostForm

Declaration `public class function PostForm(const AUrl: string; AFormData, AResponse: TStream): Boolean;`

Description Sends request by a POST HTTP request method, passing a form-data as parameter.

PostForm

Declaration `public class function PostForm(const AUrl, AFormData: string; AResponse: TStream): Boolean;`

Description Sends request by a POST HTTP request method, passing a form-data as parameter.

PostFile

Declaration `public class function PostFile(const AUrl, AFieldName, AFileName: string; AFile, AResponse: TStream): Boolean;`

Description Sends request by a POST HTTP request method, passing a file as parameter.

PostFile

Declaration `public class function PostFile(const AUrl, AFieldName, AFileName: string; AResponse: TStream): Boolean;`

Description Sends request by a POST HTTP request method, passing a file as parameter.

Request

Declaration `public function Request(AHttp: TBrookHTTPDef): TBrookHTTPResult;`

Description Performs the request, passing a TBrookHTTPDef as parameter.

Request

Declaration `public function Request(const AMethod, AUrl: string): TBrookHTTPResult;`

Description Performs the request, passing the method and URL as parameter.

Request

Declaration `public function Request(const AUrl: string): TBrookHTTPResult;`

Description Performs the request, passing URL as parameter.

12.5 Types

TBrookHTTPDefClass

Declaration `TBrookHTTPDefClass = class of TBrookHTTPDef;`

Description Is a metaclass for TBrookHTTPDef(12.4) class.

TBrookHTTPDefsClass

Declaration `TBrookHTTPDefsClass = class of TBrookHTTPDefs;`

Description Is a metaclass for TBrookHTTPDefs(12.4) class.

TBrookHTTPClientClass

Declaration `TBrookHTTPClientClass = class of TBrookHTTPClient;`

Description Is a metaclass for TBrookHTTPClient(12.4) class.

Chapter 13

Unit BrookHTTPConsts

13.1 Description

Brook HTTP consts.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

13.2 Uses

- BrookConsts(6)

13.3 Constants

BROOK_HTTP_STATUS_CODE_CONTINUE _____

Declaration BROOK_HTTP_STATUS_CODE_CONTINUE = 100;

BROOK_HTTP_STATUS_CODE_SWITCHING_PROTOCOLS _____

Declaration BROOK_HTTP_STATUS_CODE_SWITCHING_PROTOCOLS = 101;

BROOK_HTTP_STATUS_CODE_OK _____

Declaration BROOK_HTTP_STATUS_CODE_OK = 200;

BROOK_HTTP_STATUS_CODE_CREATED _____

Declaration BROOK_HTTP_STATUS_CODE_CREATED = 201;

BROOK_HTTP_STATUS_CODE_ACCEPTED _____

Declaration BROOK_HTTP_STATUS_CODE_ACCEPTED = 202;

BROOK_HTTP_STATUS_CODE_NON_AUTHORITATIVE_INFORMATION _____

Declaration BROOK_HTTP_STATUS_CODE_NON_AUTHORITATIVE_INFORMATION = 203;

BROOK_HTTP_STATUS_CODE_NO_CONTENT _____

Declaration BROOK_HTTP_STATUS_CODE_NO_CONTENT = 204;

BROOK_HTTP_STATUS_CODE_RESET_CONTENT _____

Declaration BROOK_HTTP_STATUS_CODE_RESET_CONTENT = 205;

BROOK_HTTP_STATUS_CODE_PARTIAL_CONTENT _____

Declaration BROOK_HTTP_STATUS_CODE_PARTIAL_CONTENT = 206;

BROOK_HTTP_STATUS_CODE_MULTIPLE_CHOICES _____

Declaration BROOK_HTTP_STATUS_CODE_MULTIPLE_CHOICES = 300;

BROOK_HTTP_STATUS_CODE_MOVED_PERMANENTLY _____

Declaration BROOK_HTTP_STATUS_CODE_MOVED_PERMANENTLY = 301;

BROOK_HTTP_STATUS_CODE_FOUND _____

Declaration BROOK_HTTP_STATUS_CODE_FOUND = 302;

BROOK_HTTP_STATUS_CODE_SEE_OTHER _____

Declaration BROOK_HTTP_STATUS_CODE_SEE_OTHER = 303;

BROOK_HTTP_STATUS_CODE_NOT_MODIFIED _____

Declaration BROOK_HTTP_STATUS_CODE_NOT_MODIFIED = 304;

BROOK_HTTP_STATUS_CODE_USE_PROXY _____

Declaration BROOK_HTTP_STATUS_CODE_USE_PROXY = 305;

BROOK_HTTP_STATUS_CODE_TEMPORARY_REDIRECT _____

Declaration BROOK_HTTP_STATUS_CODE_TEMPORARY_REDIRECT = 307;

BROOK_HTTP_STATUS_CODE_BAD_REQUEST _____

Declaration BROOK_HTTP_STATUS_CODE_BAD_REQUEST = 400;

BROOK_HTTP_STATUS_CODE_UNAUTHORIZED _____

Declaration BROOK_HTTP_STATUS_CODE_UNAUTHORIZED = 401;

BROOK_HTTP_STATUS_CODE_PAYMENT_REQUIRED _____

Declaration BROOK_HTTP_STATUS_CODE_PAYMENT_REQUIRED = 402;

BROOK_HTTP_STATUS_CODE_FORBIDDEN _____

Declaration BROOK_HTTP_STATUS_CODE_FORBIDDEN = 403;

BROOK_HTTP_STATUS_CODE_NOT_FOUND _____

Declaration BROOK_HTTP_STATUS_CODE_NOT_FOUND = 404;

BROOK_HTTP_STATUS_CODE_METHOD_NOT_ALLOWED _____

Declaration BROOK_HTTP_STATUS_CODE_METHOD_NOT_ALLOWED = 405;

BROOK_HTTP_STATUS_CODE_NOT_ACCEPTABLE _____

Declaration BROOK_HTTP_STATUS_CODE_NOT_ACCEPTABLE = 406;

BROOK_HTTP_STATUS_CODE_PROXY_AUTHENTICATION_REQUIRED _____

Declaration BROOK_HTTP_STATUS_CODE_PROXY_AUTHENTICATION_REQUIRED = 407;

BROOK_HTTP_STATUS_CODE_REQUEST_TIMEOUT _____

Declaration BROOK_HTTP_STATUS_CODE_REQUEST_TIMEOUT = 408;

BROOK_HTTP_STATUS_CODE_CONFLICT _____

Declaration BROOK_HTTP_STATUS_CODE_CONFLICT = 409;

BROOK_HTTP_STATUS_CODE_GONE _____

Declaration BROOK_HTTP_STATUS_CODE_GONE = 410;

BROOK_HTTP_STATUS_CODE_LENGTH_REQUIRED _____

Declaration BROOK_HTTP_STATUS_CODE_LENGTH_REQUIRED = 411;

BROOK_HTTP_STATUS_CODE_PRECONDITION_FAILED _____

Declaration BROOK_HTTP_STATUS_CODE_PRECONDITION_FAILED = 412;

BROOK_HTTP_STATUS_CODE_REQUEST_ENTITY_TOO_LARGE _____

Declaration BROOK_HTTP_STATUS_CODE_REQUEST_ENTITY_TOO_LARGE = 413;

BROOK_HTTP_STATUS_CODE_REQUEST_URI_TOO_LONG _____

Declaration BROOK_HTTP_STATUS_CODE_REQUEST_URI_TOO_LONG = 414;

BROOK_HTTP_STATUS_CODE_UNSUPPORTED_MEDIA_TYPE _____

Declaration BROOK_HTTP_STATUS_CODE_UNSUPPORTED_MEDIA_TYPE = 415;

BROOK_HTTP_STATUS_CODE_REQUESTED_RANGE_NOT_SATISFIABLE

Declaration BROOK_HTTP_STATUS_CODE_REQUESTED_RANGE_NOT_SATISFIABLE = 416;

BROOK_HTTP_STATUS_CODE_EXPECTATION_FAILED _____

Declaration BROOK_HTTP_STATUS_CODE_EXPECTATION_FAILED = 417;

BROOK_HTTP_STATUS_CODE_INTERNAL_SERVER_ERROR _____

Declaration BROOK_HTTP_STATUS_CODE_INTERNAL_SERVER_ERROR = 500;

BROOK_HTTP_STATUS_CODE_NOT_IMPLEMENTED _____

Declaration BROOK_HTTP_STATUS_CODE_NOT_IMPLEMENTED = 501;

BROOK_HTTP_STATUS_CODE_BAD_GATEWAY _____

Declaration BROOK_HTTP_STATUS_CODE_BAD_GATEWAY = 502;

BROOK_HTTP_STATUS_CODE_SERVICE_UNAVAILABLE _____

Declaration BROOK_HTTP_STATUS_CODE_SERVICE_UNAVAILABLE = 503;

BROOK_HTTP_STATUS_CODE_GATEWAY_TIMEOUT _____

Declaration BROOK_HTTP_STATUS_CODE_GATEWAY_TIMEOUT = 504;

BROOK_HTTP_STATUS_CODE_HTTP_VERSION_NOT_SUPPORTED _____

Declaration BROOK_HTTP_STATUS_CODE_HTTP_VERSION_NOT_SUPPORTED = 505;

BROOK_HTTP_REASON_PHRASE_CONTINUE _____

Declaration BROOK_HTTP_REASON_PHRASE_CONTINUE = 'Continue';

BROOK_HTTP_REASON_PHRASE_SWITCHING_PROTOCOLS _____

Declaration BROOK_HTTP_REASON_PHRASE_SWITCHING_PROTOCOLS = 'Switching Protocols';

BROOK_HTTP_REASON_PHRASE_OK _____

Declaration BROOK_HTTP_REASON_PHRASE_OK = 'OK';

BROOK_HTTP_REASON_PHRASE_CREATED _____

Declaration BROOK_HTTP_REASON_PHRASE_CREATED = 'Created';

BROOK_HTTP_REASON_PHRASE_ACCEPTED _____

Declaration BROOK_HTTP_REASON_PHRASE_ACCEPTED = 'Accepted';

BROOK_HTTP_REASON_PHRASE_NON_AUTHORITATIVE_INFORMATION

Declaration BROOK_HTTP_REASON_PHRASE_NON_AUTHORITATIVE_INFORMATION = 'Non-Authoritative Information';

BROOK_HTTP_REASON_PHRASE_NO_CONTENT _____

Declaration BROOK_HTTP_REASON_PHRASE_NO_CONTENT = 'No Content';

BROOK_HTTP_REASON_PHRASE_RESET_CONTENT _____

Declaration BROOK_HTTP_REASON_PHRASE_RESET_CONTENT = 'Reset Content';

BROOK_HTTP_REASON_PHRASE_PARTIAL_CONTENT _____

Declaration BROOK_HTTP_REASON_PHRASE_PARTIAL_CONTENT = 'Partial Content';

BROOK_HTTP_REASON_PHRASE_MULTIPLE_CHOICES _____

Declaration BROOK_HTTP_REASON_PHRASE_MULTIPLE_CHOICES = 'Multiple Choices';

BROOK_HTTP_REASON_PHRASE_MOVED_PERMANENTLY _____

Declaration BROOK_HTTP_REASON_PHRASE_MOVED_PERMANENTLY = 'Moved Permanently';

BROOK_HTTP_REASON_PHRASE_FOUND _____

Declaration BROOK_HTTP_REASON_PHRASE_FOUND = 'Found';

BROOK_HTTP_REASON_PHRASE_SEE_OTHER _____

Declaration BROOK_HTTP_REASON_PHRASE_SEE_OTHER = 'See Other';

BROOK_HTTP_REASON_PHRASE_NOT_MODIFIED _____

Declaration BROOK_HTTP_REASON_PHRASE_NOT_MODIFIED = 'Not Modified';

BROOK_HTTP_REASON_PHRASE_USE_PROXY _____

Declaration BROOK_HTTP_REASON_PHRASE_USE_PROXY = 'Use Proxy';

BROOK_HTTP_REASON_PHRASE_TEMPORARY_REDIRECT _____

Declaration BROOK_HTTP_REASON_PHRASE_TEMPORARY_REDIRECT = 'Temporary Redirect';

BROOK_HTTP_REASON_PHRASE_BAD_REQUEST _____

Declaration BROOK_HTTP_REASON_PHRASE_BAD_REQUEST = 'Bad Request';

BROOK_HTTP_REASON_PHRASE_UNAUTHORIZED _____

Declaration BROOK_HTTP_REASON_PHRASE_UNAUTHORIZED = 'Unauthorized';

BROOK_HTTP_REASON_PHRASE_PAYMENT_REQUIRED _____

Declaration BROOK_HTTP_REASON_PHRASE_PAYMENT_REQUIRED = 'Payment Required';

BROOK_HTTP_REASON_PHRASE_FORBIDDEN _____

Declaration BROOK_HTTP_REASON_PHRASE_FORBIDDEN = 'Forbidden';

BROOK_HTTP_REASON_PHRASE_NOT_FOUND _____

Declaration BROOK_HTTP_REASON_PHRASE_NOT_FOUND = 'Not Found';

BROOK_HTTP_REASON_PHRASE_METHOD_NOT_ALLOWED _____

Declaration BROOK_HTTP_REASON_PHRASE_METHOD_NOT_ALLOWED = 'Method Not Allowed';

BROOK_HTTP_REASON_PHRASE_NOT_ACCEPTABLE _____

Declaration BROOK_HTTP_REASON_PHRASE_NOT_ACCEPTABLE = 'Not Acceptable';

BROOK_HTTP_REASON_PHRASE_PROXY_AUTHENTICATION_REQUIRED

Declaration BROOK_HTTP_REASON_PHRASE_PROXY_AUTHENTICATION_REQUIRED = 'Proxy Authentication Required';

BROOK_HTTP_REASON_PHRASE_REQUEST_TIMEOUT _____

Declaration BROOK_HTTP_REASON_PHRASE_REQUEST_TIMEOUT = 'Request Timeout';

BROOK_HTTP_REASON_PHRASE_CONFLICT _____

Declaration BROOK_HTTP_REASON_PHRASE_CONFLICT = 'Conflict';

BROOK_HTTP_REASON_PHRASE_GONE _____

Declaration BROOK_HTTP_REASON_PHRASE_GONE = 'Gone';

BROOK_HTTP_REASON_PHRASE_LENGTH_REQUIRED _____

Declaration BROOK_HTTP_REASON_PHRASE_LENGTH_REQUIRED = 'Length Required';

BROOK_HTTP_REASON_PHRASE_PRECONDITION_FAILED _____

Declaration BROOK_HTTP_REASON_PHRASE_PRECONDITION_FAILED = 'Precondition Failed';

BROOK_HTTP_REASON_PHRASE_REQUEST_ENTITY_TOO_LARGE _____

Declaration BROOK_HTTP_REASON_PHRASE_REQUEST_ENTITY_TOO_LARGE = 'Request Entity Too
Large';

BROOK_HTTP_REASON_PHRASE_REQUEST_URI_TOO_LONG _____

Declaration BROOK_HTTP_REASON_PHRASE_REQUEST_URI_TOO_LONG = 'Request-URI Too Long';

BROOK_HTTP_REASON_PHRASE_UNSUPPORTED_MEDIA_TYPE _____

Declaration BROOK_HTTP_REASON_PHRASE_UNSUPPORTED_MEDIA_TYPE = 'Unsupported Media Type';

BROOK_HTTP_REASON_PHRASE_REQUESTED_RANGE_NOT_SATISFIABLE

Declaration BROOK_HTTP_REASON_PHRASE_REQUESTED_RANGE_NOT_SATISFIABLE = 'Requested Range
Not Satisfiable';

BROOK_HTTP_REASON_PHRASE_EXPECTATION_FAILED _____

Declaration BROOK_HTTP_REASON_PHRASE_EXPECTATION_FAILED = 'Expectation Failed';

BROOK_HTTP_REASON_PHRASE_INTERNAL_SERVER_ERROR _____

Declaration BROOK_HTTP_REASON_PHRASE_INTERNAL_SERVER_ERROR = 'Internal Server Error';

BROOK_HTTP_REASON_PHRASE_NOT_IMPLEMENTED _____

Declaration BROOK_HTTP_REASON_PHRASE_NOT_IMPLEMENTED = 'Not Implemented';

BROOK_HTTP_REASON_PHRASE_BAD_GATEWAY _____

Declaration BROOK_HTTP_REASON_PHRASE_BAD_GATEWAY = 'Bad Gateway';

BROOK_HTTP_REASON_PHRASE_SERVICE_UNAVAILABLE _____

Declaration BROOK_HTTP_REASON_PHRASE_SERVICE_UNAVAILABLE = 'Service Unavailable';

BROOK_HTTP_REASON_PHRASE_GATEWAY_TIMEOUT _____

Declaration BROOK_HTTP_REASON_PHRASE_GATEWAY_TIMEOUT = 'Gateway Timeout';

BROOK_HTTP_REASON_PHRASE_HTTP_VERSION_NOT_SUPPORTED _____

Declaration BROOK_HTTP_REASON_PHRASE_HTTP_VERSION_NOT_SUPPORTED = 'HTTP Version Not Supported';

BROOK_HTTP_REQUEST_METHOD_GET _____

Declaration BROOK_HTTP_REQUEST_METHOD_GET = 'GET';

BROOK_HTTP_REQUEST_METHOD_POST _____

Declaration BROOK_HTTP_REQUEST_METHOD_POST = 'POST';

BROOK_HTTP_REQUEST_METHOD_PUT _____

Declaration BROOK_HTTP_REQUEST_METHOD_PUT = 'PUT';

BROOK_HTTP_REQUEST_METHOD_DELETE _____

Declaration BROOK_HTTP_REQUEST_METHOD_DELETE = 'DELETE';

BROOK_HTTP_REQUEST_METHOD_HEAD _____

Declaration BROOK_HTTP_REQUEST_METHOD_HEAD = 'HEAD';

BROOK_HTTP_REQUEST_METHOD_OPTIONS _____

Declaration BROOK_HTTP_REQUEST_METHOD_OPTIONS = 'OPTIONS';

BROOK_HTTP_REQUEST_METHOD_PATCH _____

Declaration BROOK_HTTP_REQUEST_METHOD_PATCH = 'PATCH';

BROOK_HTTP_REQUEST_METHOD_PROPFIND _____

Declaration BROOK_HTTP_REQUEST_METHOD_PROPFIND = 'PROPFIND';

BROOK_HTTP_REQUEST_METHOD_PROPPATCH _____

Declaration BROOK_HTTP_REQUEST_METHOD_PROPPATCH = 'PROPPATCH';

BROOK_HTTP_REQUEST_METHOD_COPY _____

Declaration BROOK_HTTP_REQUEST_METHOD_COPY = 'COPY';

BROOK_HTTP_REQUEST_METHOD_MOVE _____

Declaration BROOK_HTTP_REQUEST_METHOD_MOVE = 'MOVE';

BROOK_HTTP_REQUEST_METHOD_MKCOL _____

Declaration BROOK_HTTP_REQUEST_METHOD_MKCOL = 'MKCOL';

BROOK_HTTP_REQUEST_METHOD_TRACE _____

Declaration BROOK_HTTP_REQUEST_METHOD_TRACE = 'TRACE';

BROOK_HTTP_REQUEST_METHOD_LOCK _____

Declaration BROOK_HTTP_REQUEST_METHOD_LOCK = 'LOCK';

BROOK_HTTP_REQUEST_METHOD_UNLOCK _____

Declaration BROOK_HTTP_REQUEST_METHOD_UNLOCK = 'UNLOCK';

BROOK_HTTP_HEADER_CACHE_CONTROL _____

Declaration BROOK_HTTP_HEADER_CACHE_CONTROL = 'Cache-Control';

BROOK_HTTP_HEADER_CONTENT_DESCRIPTION _____

Declaration BROOK_HTTP_HEADER_CONTENT_DESCRIPTION = 'Content-Description';

BROOK_HTTP_HEADER_CONTENT_DISPOSITION _____

Declaration BROOK_HTTP_HEADER_CONTENT_DISPOSITION = 'Content-Disposition';

BROOK_HTTP_HEADER_ETAG _____

Declaration BROOK_HTTP_HEADER_ETAG = 'ETag';

BROOK_HTTP_HEADER_STATUS _____

Declaration BROOK_HTTP_HEADER_STATUS = 'Status';

BROOK_HTTP_HEADER_TRANSFER_ENCODING _____

Declaration BROOK_HTTP_HEADER_TRANSFER_ENCODING = 'Transfer-Encoding';

BROOK_HTTP_HEADER_X_POWERED_BY _____

Declaration BROOK_HTTP_HEADER_X_POWERED_BY = 'X-Powered-By';

BROOK_HTTP_HEADER_CHARSET _____

Declaration BROOK_HTTP_HEADER_CHARSET = '; charset=';

BROOK_HTTP_CONTENT_TYPE_TEXT_PLAIN _____

Declaration BROOK_HTTP_CONTENT_TYPE_TEXT_PLAIN = 'text/plain';

BROOK_HTTP_CONTENT_TYPE_TEXT_HTML _____

Declaration BROOK_HTTP_CONTENT_TYPE_TEXT_HTML = 'text/html';

BROOK_HTTP_CONTENT_TYPE_TEXT_CSS _____

Declaration BROOK_HTTP_CONTENT_TYPE_TEXT_CSS = 'text/css';

BROOK_HTTP_CONTENT_TYPE_TEXT_XML _____

Declaration BROOK_HTTP_CONTENT_TYPE_TEXT_XML = 'text/xml';

BROOK_HTTP_CONTENT_TYPE_TEXT_CVS _____

Declaration BROOK_HTTP_CONTENT_TYPE_TEXT_CVS = 'text/cvs';

BROOK_HTTP_CONTENT_TYPE_TEXT_RICHTEXT _____

Declaration BROOK_HTTP_CONTENT_TYPE_TEXT_RICHTEXT = 'text/richtext';

BROOK_HTTP_CONTENT_TYPE_TEXT_RTF _____

Declaration BROOK_HTTP_CONTENT_TYPE_TEXT_RTF = 'text/rtf';

BROOK_HTTP_CONTENT_TYPE_MULTIPART_FORM_DATA _____

Declaration BROOK_HTTP_CONTENT_TYPE_MULTIPART_FORM_DATA = 'multipart/form-data';

BROOK_HTTP_CONTENT_TYPE_MULTIPART_FORM_DATA_BOUNDARY

Declaration BROOK_HTTP_CONTENT_TYPE_MULTIPART_FORM_DATA_BOUNDARY =
BROOK_HTTP_CONTENT_TYPE_MULTIPART_FORM_DATA + ' ; boundary=';

BROOK_HTTP_CONTENT_TYPE_APP_JSON

Declaration BROOK_HTTP_CONTENT_TYPE_APP_JSON = 'application/json';

BROOK_HTTP_CONTENT_TYPE_APP_XML

Declaration BROOK_HTTP_CONTENT_TYPE_APP_XML = 'application/xml';

BROOK_HTTP_CONTENT_TYPE_APP_RDF_XML

Declaration BROOK_HTTP_CONTENT_TYPE_APP_RDF_XML = 'application/rdf+xml';

BROOK_HTTP_CONTENT_TYPE_APP_X_WWW_FORM_URLENCODED

Declaration BROOK_HTTP_CONTENT_TYPE_APP_X_WWW_FORM_URLENCODED =
'application/x-www-form-urlencoded';

BROOK_HTTP_CONTENT_TYPE_APP_JAVASCRIPT

Declaration BROOK_HTTP_CONTENT_TYPE_APP_JAVASCRIPT = 'application/javascript';

BROOK_HTTP_CONTENT_TYPE_APP_OCTET_STREAM

Declaration BROOK_HTTP_CONTENT_TYPE_APP_OCTET_STREAM = 'application/octet-stream';

BROOK_HTTP_CONTENT_TYPE_APP_PDF

Declaration BROOK_HTTP_CONTENT_TYPE_APP_PDF = 'application/pdf';

BROOK_HTTP_CONTENT_TYPE_APP_ZIP

Declaration BROOK_HTTP_CONTENT_TYPE_APP_ZIP = 'application/zip';

BROOK_HTTP_CONTENT_TYPE_APP_MSWORD

Declaration BROOK_HTTP_CONTENT_TYPE_APP_MSWORD = 'application/msword';

BROOK_HTTP_CONTENT_TYPE_APP_MS_EXCEL

Declaration BROOK_HTTP_CONTENT_TYPE_APP_MS_EXCEL = 'application/vnd.ms-excel';

BROOK_HTTP_CONTENT_TYPE_APP_MS_POWERPOINT

Declaration BROOK_HTTP_CONTENT_TYPE_APP_MS_POWERPOINT = 'application/vnd.ms-powerpoint';

BROOK_HTTP_CONTENT_TYPE_APP_MP4 _____

Declaration BROOK_HTTP_CONTENT_TYPE_APP_MP4 = 'application/mp4';

BROOK_HTTP_CONTENT_TYPE_APP_MPEG4_GENERIC _____

Declaration BROOK_HTTP_CONTENT_TYPE_APP_MPEG4_GENERIC = 'application/mpeg4-generic';

BROOK_HTTP_CONTENT_TYPE_IMAGE_PNG _____

Declaration BROOK_HTTP_CONTENT_TYPE_IMAGE_PNG = 'image/png';

BROOK_HTTP_CONTENT_TYPE_IMAGE_JPEG _____

Declaration BROOK_HTTP_CONTENT_TYPE_IMAGE_JPEG = 'image/jpeg';

BROOK_HTTP_CONTENT_TYPE_IMAGE_GIF _____

Declaration BROOK_HTTP_CONTENT_TYPE_IMAGE_GIF = 'image/gif';

BROOK_HTTP_CONTENT_TYPE_IMAGE_TIFF _____

Declaration BROOK_HTTP_CONTENT_TYPE_IMAGE_TIFF = 'image/tiff';

BROOK_HTTP_CONTENT_TYPE_AUDIO_MPEG _____

Declaration BROOK_HTTP_CONTENT_TYPE_AUDIO_MPEG = 'audio/mpeg';

BROOK_HTTP_CONTENT_TYPE_VIDEO_MPEG _____

Declaration BROOK_HTTP_CONTENT_TYPE_VIDEO_MPEG = 'video/mpeg';

BROOK_HTTP_CONTENT_TYPE_VIDEO_QUICKTIME _____

Declaration BROOK_HTTP_CONTENT_TYPE_VIDEO_QUICKTIME = 'video/quicktime';

BROOK_HTTP_CACHE_CONTROL_NO_CACHE _____

Declaration BROOK_HTTP_CACHE_CONTROL_NO_CACHE = 'no-cache';

BROOK_HTTP_CACHE_CONTROL_NO_STORE _____

Declaration BROOK_HTTP_CACHE_CONTROL_NO_STORE = 'no-store';

BROOK_HTTP_CACHE_CONTROL_MAX_AGE _____

Declaration BROOK_HTTP_CACHE_CONTROL_MAX_AGE = 'max-age=';

BROOK_HTTP_CACHE_CONTROL_MAX_STALE _____

Declaration BROOK_HTTP_CACHE_CONTROL_MAX_STALE = 'max-stale';

BROOK_HTTP_CACHE_CONTROL_MIN_FRESH _____

Declaration BROOK_HTTP_CACHE_CONTROL_MIN_FRESH = 'min-fresh';

BROOK_HTTP_CACHE_CONTROL_NO_TRANSFORM _____

Declaration BROOK_HTTP_CACHE_CONTROL_NO_TRANSFORM = 'no-transform';

BROOK_HTTP_CACHE_CONTROL_ONLY_IF_CACHED _____

Declaration BROOK_HTTP_CACHE_CONTROL_ONLY_IF_CACHED = 'only-if-cached';

BROOK_HTTP_CACHE_CONTROL_CACHE_EXTENSION _____

Declaration BROOK_HTTP_CACHE_CONTROL_CACHE_EXTENSION = 'cache-extension';

BROOK_HTTP_CACHE_CONTROL_PUBLIC _____

Declaration BROOK_HTTP_CACHE_CONTROL_PUBLIC = 'public';

BROOK_HTTP_CACHE_CONTROL_PRIVATE _____

Declaration BROOK_HTTP_CACHE_CONTROL_PRIVATE = 'private';

BROOK_HTTP_CACHE_CONTROL_MUST_REVALIDATE _____

Declaration BROOK_HTTP_CACHE_CONTROL_MUST_REVALIDATE = 'must-revalidate';

BROOK_HTTP_CACHE_CONTROL_PROXY_REVALIDATE _____

Declaration BROOK_HTTP_CACHE_CONTROL_PROXY_REVALIDATE = 'proxy-revalidate';

BROOK_HTTP_CACHE_CONTROL_S_MAXAGE _____

Declaration BROOK_HTTP_CACHE_CONTROL_S_MAXAGE = 's-maxage';

BROOK_HTTP_CONTENT_DISPOSITION_INLINE _____

Declaration BROOK_HTTP_CONTENT_DISPOSITION_INLINE = 'inline';

BROOK_HTTP_CONTENT_DISPOSITION_ATTACHMENT _____

Declaration BROOK_HTTP_CONTENT_DISPOSITION_ATTACHMENT = 'attachment';

BROOK_HTTP_CONTENT_ENCODING_DEFLATE _____

Declaration BROOK_HTTP_CONTENT_ENCODING_DEFLATE = 'deflate';

BROOK_HTTP_CONTENT_ENCODING_GZIP _____

Declaration BROOK_HTTP_CONTENT_ENCODING_GZIP = 'gzip';

BROOK_HTTP_CONTENT_ENCODING_SDCH _____

Declaration BROOK_HTTP_CONTENT_ENCODING_SDCH = 'sdch';

BROOK_HTTP_CONTENT_ENCODING_X_GZIP _____

Declaration BROOK_HTTP_CONTENT_ENCODING_X_GZIP = 'x-gzip';

BROOK_HTTP_TRANSFER_ENCODING_CHUNKED _____

Declaration BROOK_HTTP_TRANSFER_ENCODING_CHUNKED = 'chunked';

BROOK_HTTP_CHARSET_ISO_8859_1 _____

Declaration BROOK_HTTP_CHARSET_ISO_8859_1 = 'iso-8859-1';

BROOK_HTTP_CHARSET_UTF_8 _____

Declaration BROOK_HTTP_CHARSET_UTF_8 = 'utf-8';

BROOK_SRV_ENV_DOCUMENT_ROOT _____

Declaration BROOK_SRV_ENV_DOCUMENT_ROOT = 'DOCUMENT_ROOT';

BROOK_SRV_ENV_SERVER_ADDR _____

Declaration BROOK_SRV_ENV_SERVER_ADDR = 'SERVER_ADDR';

BROOK_SRV_ENV_SERVER_ADMIN _____

Declaration BROOK_SRV_ENV_SERVER_ADMIN = 'SERVER_ADMIN';

BROOK_SRV_ENV_SERVER_SOFTWARE _____

Declaration BROOK_SRV_ENV_SERVER_SOFTWARE = 'SERVER_SOFTWARE';

BROOK_SRV_ENV_SERVER_NAME _____

Declaration BROOK_SRV_ENV_SERVER_NAME = 'SERVER_NAME';

BROOK_SRV_ENV_GATEWAY_INTERFACE _____

Declaration BROOK_SRV_ENV_GATEWAY_INTERFACE = 'GATEWAY_INTERFACE';

BROOK_SRV_ENV_SERVER_PROTOCOL _____

Declaration BROOK_SRV_ENV_SERVER_PROTOCOL = 'SERVER_PROTOCOL';

BROOK_SRV_ENV_SERVER_PORT _____

Declaration BROOK_SRV_ENV_SERVER_PORT = 'SERVER_PORT';

BROOK_SRV_ENV_REQUEST_METHOD _____

Declaration BROOK_SRV_ENV_REQUEST_METHOD = 'REQUEST_METHOD';

BROOK_SRV_ENV_REQUEST_URI _____

Declaration BROOK_SRV_ENV_REQUEST_URI = 'REQUEST_URI';

BROOK_SRV_ENV_PATH_INFO _____

Declaration BROOK_SRV_ENV_PATH_INFO = 'PATH_INFO';

BROOK_SRV_ENV_PATH_TRANSLATED _____

Declaration BROOK_SRV_ENV_PATH_TRANSLATED = 'PATH_TRANSLATED';

BROOK_SRV_ENV_SCRIPT_FILENAME _____

Declaration BROOK_SRV_ENV_SCRIPT_FILENAME = 'SCRIPT_FILENAME';

BROOK_SRV_ENV_SCRIPT_NAME _____

Declaration BROOK_SRV_ENV_SCRIPT_NAME = 'SCRIPT_NAME';

BROOK_SRV_ENV_QUERY_STRING _____

Declaration BROOK_SRV_ENV_QUERY_STRING = 'QUERY_STRING';

BROOK_SRV_ENV_REMOTE_HOST _____

Declaration BROOK_SRV_ENV_REMOTE_HOST = 'REMOTE_HOST';

BROOK_SRV_ENV_REMOTE_ADDR _____

Declaration BROOK_SRV_ENV_REMOTE_ADDR = 'REMOTE_ADDR';

BROOK_SRV_ENV_REMOTE_PORT _____

Declaration BROOK_SRV_ENV_REMOTE_PORT = 'REMOTE_PORT';

BROOK_SRV_ENV_AUTH_TYPE _____

Declaration BROOK_SRV_ENV_AUTH_TYPE = 'AUTH_TYPE';

BROOK_SRV_ENV_REMOTE_USER _____

Declaration BROOK_SRV_ENV_REMOTE_USER = 'REMOTE_USER';

BROOK_SRV_ENV_REMOTE_IDENT _____

Declaration BROOK_SRV_ENV_REMOTE_IDENT = 'REMOTE_IDENT';

BROOK_SRV_ENV_CONTENT_TYPE _____

Declaration BROOK_SRV_ENV_CONTENT_TYPE = 'CONTENT_TYPE';

BROOK_SRV_ENV_CONTENT_LENGTH _____

Declaration BROOK_SRV_ENV_CONTENT_LENGTH = 'CONTENT_LENGTH';

BROOK_SRV_ENV_X_FORWARDED_FOR _____

Declaration BROOK_SRV_ENV_X_FORWARDED_FOR = 'X_FORWARDED_FOR';

BROOK_CLT_ENV_HTTP_COOKIE _____

Declaration BROOK_CLT_ENV_HTTP_COOKIE = 'HTTP_COOKIE';

BROOK_CLT_ENV_HTTP_REFERER _____

Declaration BROOK_CLT_ENV_HTTP_REFERER = 'HTTP_REFERER';

BROOK_CLT_ENV_HTTP_USER_AGENT _____

Declaration BROOK_CLT_ENV_HTTP_USER_AGENT = 'HTTP_USER_AGENT';

BROOK_CLT_ENV_HTTP_IF_MODIFIED_SINCE _____

Declaration BROOK_CLT_ENV_HTTP_IF_MODIFIED_SINCE = 'HTTP_IF_MODIFIED_SINCE';

BROOK_CLT_ENV_HTTP_HOST _____

Declaration BROOK_CLT_ENV_HTTP_HOST = 'HTTP_HOST';

BROOK_CLT_ENV_HTTP_CONNECTION _____

Declaration BROOK_CLT_ENV_HTTP_CONNECTION = 'HTTP_CONNECTION';

BROOK_CLT_ENV_HTTP_CACHE_CONTROL _____

Declaration BROOK_CLT_ENV_HTTP_CACHE_CONTROL = 'HTTP_CACHE_CONTROL';

BROOK_CLT_ENV_HTTP_ACCEPT _____

Declaration BROOK_CLT_ENV_HTTP_ACCEPT = 'HTTP_ACCEPT';

BROOK_CLT_ENV_HTTP_ACCEPT_ENCODING _____

Declaration BROOK_CLT_ENV_HTTP_ACCEPT_ENCODING = 'HTTP_ACCEPT_ENCODING';

BROOK_CLT_ENV_HTTP_ACCEPT_LANGUAGE _____

Declaration BROOK_CLT_ENV_HTTP_ACCEPT_LANGUAGE = 'HTTP_ACCEPT_LANGUAGE';

BROOK_CLT_ENV_HTTP_ACCEPT_CHARSET _____

Declaration BROOK_CLT_ENV_HTTP_ACCEPT_CHARSET = 'HTTP_ACCEPT_CHARSET';

BROOK_CLT_ENV_HTTP_IF_NONE_MATCH _____

Declaration BROOK_CLT_ENV_HTTP_IF_NONE_MATCH = 'HTTP_IF_NONE_MATCH';

BROOK_CLT_ENV_HTTP_X_REQUESTED_WITH _____

Declaration BROOK_CLT_ENV_HTTP_X_REQUESTED_WITH = 'HTTP_X_REQUESTED_WITH';

BROOK_HTTP_RESPONSE_TEMPLATE_NOT_FOUND _____

Declaration BROOK_HTTP_RESPONSE_TEMPLATE_NOT_FOUND = '
<html><head><title>Page not
found</title></head><body>' + '
<h1>404 - Page not
found</h1></body></html>';

BROOK_HTTP_RESPONSE_TEMPLATE_INTERNAL_SERVER_ERROR _____

Declaration BROOK_HTTP_RESPONSE_TEMPLATE_INTERNAL_SERVER_ERROR =
'<html><head><title>Internal server error</title></head><body>' +
'<h1>500 - Internal server error</h1>' + '
<p style="color: red;" >Error:
@error</p>' +
'</body></html>';

13.4 Variables

BROOK_HTTP_CLIENT_DEFAULT_LIBRARY _____

Declaration BROOK_HTTP_CLIENT_DEFAULT_LIBRARY: string = ES;

Chapter 14

Unit BrookHTTPUtils

14.1 Description

Brook HTTP Utils unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

14.2 Uses

- `BrookHTTPClient(12)`
- `BrookMessages(16)`
- `BrookHTTPConsts(13)`
- `BrookConsts(6)`
- `BrookUtils(24)`
- `HTTPDefs`
- `SysUtils`
- `FPJSON`
- `JSONParser`

14.3 Overview

BrookDateTimeToGMT Converts TDateTime to a GMT format.

BrookSameURL Compare two URLs ignoring a possible final slash.

BrookStatusCodeToReasonPhrase Returns the reason phrase corresponding to a status code.

BrookReasonPhraseToStatusCode Returns the status code corresponding to a reason phrase.

BrookGetAcceptEncodingSet Returns a set of HTTP AcceptEncoding header.

BrookGetAcceptEncoding Returns a string of HTTP AcceptEncoding.

BrookMimeTypeFromFileExt Returns a MIME type by file extension.

BrookMimeTypeFromFileName Returns a MIME type by file name.

BrookFileExtFromMimeType Returns a file extension by MIME type.

BrookExtractUrlFileName Extracts the file name of a URL.

BrookExtractUrlFileName Extracts the file name of a URL and escapes it.

BrookRequestMethodToStr Returns the string corresponding to a TBrookRequestMethod.

BrookStrToRequestMethod Returns the TBrookRequestMethod corresponding to a string.

BrookHttpRequest Perform HTTP requests.

BrookHttpRequest Perform HTTP requests returning the response as TJSONData.

BrookHttpRequest Perform HTTP requests returning the response as TJSONArray.

BrookHttpRequest Perform HTTP requests returning the response as TJSONObject.

BrookHttpRequest Perform HTTP requests passing the data as TJSONData.

BrookHttpRequest Perform HTTP requests passing the data as TJSONObject.

14.4 Functions and Procedures

BrookDateTimeToGMT

Declaration `function BrookDateTimeToGMT(const ADateTime: TDateTime): string;`

Description Converts TDateTime to a GMT format.

BrookSameURL

Declaration `function BrookSameURL(AURL1, AURL2: string): Boolean;`

Description Compare two URLs ignoring a possible final slash.

BrookStatusCodeToReasonPhrase

Declaration `function BrookStatusCodeToReasonPhrase(const AStatusCode: Word): string;`

Description Returns the reason phrase corresponding to a status code.

BrookReasonPhraseToStatusCode

Declaration `function BrookReasonPhraseToStatusCode(const AReasonPhrase: string): Word;`

Description Returns the status code corresponding to a reason phrase.

BrookGetAcceptEncodingSet

Declaration `function BrookGetAcceptEncodingSet(const AAcceptEncoding: ShortString):
TBrookAcceptEncodingSet;`

Description Returns a set of HTTP AcceptEncoding header.

BrookGetAcceptEncoding

Declaration `function BrookGetAcceptEncoding(const AAcceptEncoding:
TBrookAcceptEncodingSet): string;`

Description Returns a string of HTTP AcceptEncoding.

BrookMimeTypeFromFileExt

Declaration `function BrookMimeTypeFromFileExt(const AValue: string): string;`

Description Returns a MIME type by file extension.

BrookMimeTypeFromFileName

Declaration `function BrookMimeTypeFromFileName(const AValue: string): string;`

Description Returns a MIME type by file name.

BrookFileExtFromMimeType

Declaration `function BrookFileExtFromMimeType(const AValue: string): string;`

Description Returns a file extension by MIME type.

BrookExtractUrlFileName

Declaration `function BrookExtractUrlFileName(const AUrl: string): string;`

Description Extracts the file name of a URL.

BrookExtractUrlFileName

Declaration `function BrookExtractUrlFileName(const AUrl: string; const
AEscapeQueryString: Boolean): string;`

Description Extracts the file name of a URL and escapes it.

BrookRequestMethodToStr

Declaration `function BrookRequestMethodToStr(const AMethod: TBrookRequestMethod):
string;`

Description Returns the string corresponding to a TBrookRequestMethod.

BrookStrToRequestMethod

Declaration `function BrookStrToRequestMethod(const AMethod: string):
TBrookRequestMethod;`

Description Returns the TBrookRequestMethod corresponding to a string.

BrookHttpRequest

Declaration `function BrookHttpRequest(const AUrl: string; const AMethod:
TBrookRequestMethod = rmGet; const AHttpClientLibrary: string = ES):
TBrookHTTPResult;`

Description Perform HTTP requests. (allows all request methods)

BrookHttpRequest

Declaration `function BrookHttpRequest(const AUrl: string; out AResponse: TJSONData;
const AMethod: TBrookRequestMethod = rmGet; const AHttpClientLibrary:
string = ES): TBrookHTTPResult;`

Description Perform HTTP requests returning the response as TJSONData. (allows request methods: GET, HEAD, OPTIONS and TRACE)

BrookHttpRequest

Declaration `function BrookHttpRequest(const AUrl: string; out AResponse: TJSONArray;
const AMethod: TBrookRequestMethod = rmGet; const AHttpClientLibrary:
string = ES): TBrookHTTPResult;`

Description Perform HTTP requests returning the response as TJSONArray. (allows request methods: GET, HEAD, OPTIONS and TRACE)

BrookHttpRequest

Declaration `function BrookHttpRequest(const AUrl: string; out AResponse: TJSONObject;
const AMethod: TBrookRequestMethod = rmGet; const AHttpClientLibrary:
string = ES): TBrookHTTPResult;`

Description Perform HTTP requests returning the response as TJSONObject. (allows request methods: GET, HEAD, OPTIONS and TRACE)

BrookHttpRequest

Declaration `function BrookHttpRequest(var AData: TJSONData; const AUrl: string; const
AMethod: TBrookRequestMethod = rmPost; const AHttpClientLibrary: string =
ES; const AEncodeData: Boolean = True): TBrookHTTPResult;`

Description Perform HTTP requests passing the data as TJSONData. (allows request methods: POST, PUT and DELETE)

BrookHttpRequest

Declaration `function BrookHttpRequest(var AData: TJSONObject; const AUrl: string;
const AMethod: TBrookRequestMethod = rmPost; const AHttpClientLibrary:
string = ES; const AEncodeData: Boolean = True): TBrookHTTPResult;`

Description Perform HTTP requests passing the data as TJSONObject. (allows request methods: POST, PUT and DELETE)

14.5 Types

TBrookAcceptEncodingSet

Declaration `TBrookAcceptEncodingSet = set of (aeDeflate, aeGzip, aeSdch, aeXGzip);`

Description Defines a set to represent the AcceptEncoding HTTP header.

Chapter 15

Unit BrookJSONHelper

15.1 Description

Brook JSON Helper unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

15.2 Uses

- `BrookConsts(6)`
- `FPJSON`
- `SysUtils`
- `MD5`
- `SHA1`

15.3 Overview

`TBrookJSONDataHelper` Class Adds features to the `TJSONData` class.

`TBrookJSONObjectHelper` Class Adds features to the `TJSONObject` class.

15.4 Classes, Interfaces, Objects and Records

TBrookJSONDataHelper Class helper for TJSONData

Hierarchy

TBrookJSONDataHelper > TObject

Description

Adds features to the TJSONData class.

Properties

AsChar	<code>public property AsChar: Char read GetAsChar write SetAsChar;</code> Get or set the JSONData as a Char.
AsSmallInt	<code>public property AsSmallInt: SmallInt read GetAsSmallInt write SetAsSmallInt;</code> Get or set the JSONData as a SmallInt.
AsTime	<code>public property AsTime: TTime read GetAsTime write SetAsTime;</code> Get or set the JSONData as a TTime.
AsDate	<code>public property AsDate: TDate read GetAsDate write SetAsDate;</code> Get or set the JSONData as a TDate.
AsDateTime	<code>public property AsDateTime: TDateTime read GetAsDateTime write SetAsDateTime;</code> Get or set the JSONData as a TDateTime.
AsTrimStr	<code>public property AsTrimStr: string read GetAsTrimStr write SetAsTrimStr;</code> Get or set the JSONData as a trimmed string.
AsTrimJS	<code>public property AsTrimJS: string read GetAsTrimJS write SetAsTrimJS;</code> Get or set the JSONData as a trimmed JSON string.
AsLowerStr	<code>public property AsLowerStr: string read GetAsLowerStr write SetAsLowerStr;</code> Get or set the JSONData as a lower string.
AsLowerJS	<code>public property AsLowerJS: string read GetAsLowerJS write SetAsLowerJS;</code> Get or set the JSONData as a lower JSON string.
AsUpperStr	<code>public property AsUpperStr: string read GetAsUpperStr write SetAsUpperStr;</code> Get or set the JSONData as a upper string.
AsUpperJS	<code>public property AsUpperJS: string read GetAsUpperJS write SetAsUpperJS;</code> Get or set the JSONData as a upper JSON string.

AsQuotedStr public property AsQuotedStr: string read GetAsQuotedStr write SetAsQuotedStr;
Get or set the JSONData as a quoted string.

AsMD5 public property AsMD5: string read GetAsMD5 write SetAsMD5;
Get or set the JSONData as a MD5 string.

AsSHA1 public property AsSHA1: string read GetAsSHA1 write SetAsSHA1;
Get or set the JSONData as a SHA1 string.

IsEmpty public property IsEmpty: Boolean read GetIsEmpty;
Get if the JSONData is an empty string.

IsBlank public property IsBlank: Boolean read GetIsBlank;
Get if the JSONData is a blank string.

TBrookJSONObjectHelper Class helper for TJSONObject ---

Hierarchy

TBrookJSONObjectHelper > TObject

Description

Adds features to the TJSONObject class.

Properties

Times public property Times[AName: string]: TTime read GetTimes write SetTimes;
Get or set a value as TTime.

Dates public property Dates[AName: string]: TTime read GetDates write SetDates;
Get or set a value as TDate.

DateTimes public property DateTimes[AName: string]: TTime read GetDateTimes write SetDateTimes;
Get or set a value as TDateTime.

Methods

Exists

Declaration public function Exists(const AName: string): Boolean;

Description Checks if a name exists.

Chapter 16

Unit BrookMessages

16.1 Description

Brook Messages unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

16.2 Uses

- `BrookClasses(4)`
- `BrookConsts(6)`
- `BrookException(11)`
- `Classes`
- `SysUtils`

16.3 Overview

`EBrookMessage` Class Handles exceptions for `TBrookMessage(16.4)`.

`EBrookMessages` Class Handles exceptions for `TBrookMessages(16.4)`.

`TBrookMessage` Class Handles strings for translation of messages.

`TBrookMessages` Class Registers the available language in the framework.

`BrookSetDefaultLanguage` Translates the framework messages according to the default language.

16.4 Classes, Interfaces, Objects and Records

EBrookMessage Class

Hierarchy

EBrookMessage > EBrook(11.4) > Exception

Description

Handles exceptions for TBrookMessage(16.4).

EBrookMessages Class

Hierarchy

EBrookMessages > EBrook(11.4) > Exception

Description

Handles exceptions for TBrookMessages(16.4).

TBrookMessage Class

Hierarchy

TBrookMessage > TBrookObject(4.4) > TObject

Description

Handles strings for translation of messages.

Methods

Register

Declaration `public class procedure Register;`

Description Register the broker class.

Translate

Declaration `public class procedure Translate; virtual; abstract;`

Description Translates the framework messages.

CountryCode

Declaration `public class function CountryCode: ShortString; virtual; abstract;`

Description Returns the country code, exemple: en-US.

CountryName

Declaration `public class function CountryName: string; virtual; abstract;`

Description Returns the country name, exemple: United States.

TBrookMessages Class

Hierarchy

TBrookMessages > TBrookObject(4.4) > TObject

Description

Registers the available language in the framework.

Properties

Items `public property Items[const AIndex: Integer]: TBrookMessageClass read GetItems
write SetItems;`

The list of translation items.

Methods

Create

Declaration `public constructor Create;`

Description Creates an instance of a TBrookMessages(16.4) class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of TBrookMessages(16.4) class.

RegisterService

Declaration `public class procedure RegisterService;`

Description Registers the service provided by this class.

UnregisterService

Declaration `public class procedure UnregisterService;`

Description Unregisters the service provided by this class.

Service

Declaration `public class function Service: TBrookMessages;`

Description Return a instance of this class.

Count

Declaration `public function Count: Integer;`

Description Returns the number of registered translations.

SetLanguage

Declaration `public procedure SetLanguage(const ACountryCode: ShortString);`

Description Translates the framework messages acoording to the country code.

Find

Declaration `public function Find(const ACountryCode: ShortString): TBrookMessageClass;`

Description Finds a translation item by its country code.

ItemByCountryCode

Declaration `public function ItemByCountryCode(const ACountryCode: ShortString):
TBrookMessageClass;`

Description Returns a translation item by its country code.

Add

Declaration `public procedure Add(AClass: TBrookMessageClass);`

Description Adds a translation item.

Remove

Declaration `public procedure Remove(AClass: TBrookMessageClass);`

Description Removes a translation item.

16.5 Functions and Procedures

BrookSetDefaultLanguage

Declaration `procedure BrookSetDefaultLanguage;`

Description Translates the framework messages acoording to the default language.

16.6 Types

TBrookMessageClass

Declaration `TBrookMessageClass = class of TBrookMessage;`

Description Is a metaclass for TBrookMessage(16.4) class.

TBrookMessagesClass

Declaration `TBrookMessagesClass = class of TBrookMessages;`

Description Is a metaclass for TBrookMessages(16.4) class.

16.7 Variables

SBrookInvalidRequestMethodError

Declaration `SBrookInvalidRequestMethodError: string = ES;`

SBrookItemNotFoundError

Declaration `SBrookItemNotFoundError: string = ES;`

SBrookFileNotFoundError

Declaration `SBrookFileNotFoundError: string = ES;`

SBrookNoRequestMethodError

Declaration `SBrookNoRequestMethodError: string = ES;`

SBrookNoApplicationRegisteredError

Declaration `SBrookNoApplicationRegisteredError: string = ES;`

SBrookApplicationAlreadyRegisteredError

Declaration `SBrookApplicationAlreadyRegisteredError: string = ES;`

SBrookRegiterTBrookActionError

Declaration `SBrookRegiterTBrookActionError: string = ES;`

SBrookActionAlreadyRegisteredError

Declaration `SBrookActionAlreadyRegisteredError: string = ES;`

SBrookDefaultActionAlreadyRegisteredError _____

Declaration SBrookDefaultActionAlreadyRegisteredError: string = ES;

SBrookPatternAlreadyRegisteredError _____

Declaration SBrookPatternAlreadyRegisteredError: string = ES;

SBrookRouterAlreadyRegisteredError _____

Declaration SBrookRouterAlreadyRegisteredError: string = ES;

SBrookNoRouterRegisteredError _____

Declaration SBrookNoRouterRegisteredError: string = ES;

SBrookMessagesAlreadyRegisteredError _____

Declaration SBrookMessagesAlreadyRegisteredError: string = ES;

SBrookNoMessagesRegisteredError _____

Declaration SBrookNoMessagesRegisteredError: string = ES;

SBrookNoRouteRegisteredError _____

Declaration SBrookNoRouteRegisteredError: string = ES;

SBrookCfgFileNotFoundError _____

Declaration SBrookCfgFileNotFoundError: string = ES;

SBrookNilJSONParamError _____

Declaration SBrookNilJSONParamError: string = ES;

SBrookNilParamError _____

Declaration SBrookNilParamError: string = ES;

SBrookDataBaseAlreadyRegisteredError _____

Declaration SBrookDataBaseAlreadyRegisteredError: string = ES;

SBrookNoDataBaseRegisteredError _____

Declaration SBrookNoDataBaseRegisteredError: string = ES;

SBrookNoQueryBrokerClassRegisteredError _____

Declaration SBrookNoQueryBrokerClassRegisteredError: string = ES;

SBrookEmptyTableNameError _____

Declaration SBrookEmptyTableNameError: string = ES;

SBrookEmptyLibraryNameError _____

Declaration SBrookEmptyLibraryNameError: string = ES;

SBrookEmptyCountryCodeError _____

Declaration SBrookEmptyCountryCodeError: string = ES;

SBrookMethodNotAllowedError _____

Declaration SBrookMethodNotAllowedError: string = ES;

SBrookIncompatibleTypesError _____

Declaration SBrookIncompatibleTypesError: string = ES;

Chapter 17

Unit BrookQuery

17.1 Description

Brook Query unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

17.2 Uses

- `BrookClasses(4)`
- `BrookDataBase(7)`
- `BrookException(11)`
- `BrookMessages(16)`
- `BrookConsts(6)`
- `BrookDBConsts(9)`
- `DB`
- `SQLdb`
- `FPJSON`
- `JSONParser`
- `SysUtils`

- **Classes**
- **Variants**

17.3 Overview

EBrookQuery Class Handles exceptions for **TBrookQuery**(17.4).

TBrookQuery Class Handles database queries.

17.4 Classes, Interfaces, Objects and Records

EBrookQuery Class

Hierarchy

EBrookQuery > EBrook(11.4) > Exception

Description

Handles exceptions for **TBrookQuery**(17.4).

TBrookQuery Class

Hierarchy

TBrookQuery > TBrookObject(4.4) > TObject

Description

Handles database queries.

Properties

SQL	public property SQL: TStrings read GetSQL; Is the SQL statement.
Fields	public property Fields: TFields read GetFields; Are the query fields.
Params	public property Params: TParams read GetParams; Are the query parameters.
BOF	public property BOF: Boolean read GetBOF; Checks if the cursor is at the first register.
EOF	public property EOF: Boolean read GetEOF; Checks if the cursor is at the last register.

Bookmark	public property Bookmark: TBookmark read GetBookmark write GotoBookmark; Returns or creates a bookmark.
Modified	public property Modified: Boolean read GetModified; Returns True if the content of the query has changed.
State	public property State: TDataSetState read GetState; Get the current query state.
Active	public property Active: Boolean read GetActive write SetActive; Checks if the query is active.
DataSet	public property DataSet: TDataSet read GetDataSet; Get the dataset of the query.
FieldDefs	public property FieldDefs: TFieldDefs read GetFieldDefs; Get the fieldefs of the query.
DataSource	public property DataSource: TDataSource read GetDataSource write SetDataSource; Get or set a master query datasource.
DataBase	public property DataBase: TBrookDataBase read GetDataBase write SetDataBase; Get or set a query database.
DateAsString	public property DateAsString: Boolean read FDateAsString write FDateAsString; Enables the date saving as string.
AsJSON	public property AsJSON: TJSONStringType read GetAsJSON write SetAsJSON; Returns a JSON string with the query registers.
Rows	public property Rows: TJSONArray read GetRows; Returns a JSON array with the query registers.
Row	public property Row: TJSONObject read GetRow; Returns the current register in a JSON object.
Position	public property Position: LongInt read GetPosition write SetPosition; Get the position of the current register.

Methods

Init

Declaration `public constructor Init(ADataBase: TBrookDataBase); virtual; abstract;`

Description Creates an instance of a TBrookQuery(17.4) class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of TBrookQuery(17.4) class.

Create

Declaration `public class function Create(ADataBase: TBrookDataBase): TBrookQuery;`

Description Creates an instance of a TBrookQuery(17.4) class.

InitBrokerClass

Declaration `public class procedure InitBrokerClass;`

Description Initializes the class broker.

BrokerClass

Declaration `public class function BrokerClass: TBrookQueryClass; virtual;`

Description Registers the broker class.

GetJSONType

Declaration `public class function GetJSONType(const AFieldClass: TFieldClass): ShortString;`

Description Returns a JSON type according to a field class.

GetJSONType

Declaration `public class function GetJSONType(const AField: TField): ShortString;`

Description Returns a JSON type according to a field type.

JSONToFields

Declaration `public class procedure JSONToFields(AJSON: TJSONObject; AFields: TFields; const ADateAsString: Boolean);`

Description Set the content of a JSON object to fields.

FieldsToJSON

Declaration `public class procedure FieldsToJSON(AFields: TFields; AJSON: TJSONObject;
const ADateAsString: Boolean);`

Description Set the content of fields to a JSON object.

DataSetToJSON

Declaration `public class procedure DataSetToJSON(ADataset: TDataSet; AJSON: TJSONArray;
const ADateAsString: Boolean);`

Description Set the content of a dataset to a JSON array.

JSONToParams

Declaration `public class procedure JSONToParams(AJSON: TJSONObject; AParams: TParams;
AFieldDefs: TFieldDefs; const ADateAsString, AAutoCreateParams: Boolean);`

Description Set the content of a JSON object to params.

FieldDefsToSchema

Declaration `public class procedure FieldDefsToSchema(AFieldDefs: TFieldDefs; ASchema:
TJSONObject); overload;`

Description Set the content of fieldsdefs to a JSON object.

FieldDefsToSchema

Declaration `public class procedure FieldDefsToSchema(AFieldDefs: TFieldDefs; out
ASchema: TJSONStringType); overload;`

Description Set the content of fieldsdefs to a JSON string.

GetJSONAttributes

Declaration `public class procedure GetJSONAttributes(AJSON: TJSONObject; out ANames:
string; out AValues: Variant; const ADelimiter: Char = SC);`

Description Get the attributes of a JSON object and converts into a variant.

GetSchema

Declaration `public procedure GetSchema(out ASchema: TJSONObject); overload;`

Description Returns a JSON object with the query columns.

GetSchema

Declaration `public function GetSchema: TJSONStringType; overload;`

Description Returns a JSON string with the query columns.

Locate

Declaration `public class function Locate(ADataset: TDataSet; const AJSON: TJSONObject;
const AOptions: TLocateOptions): Boolean; overload;`

Description Locates a register from a JSON object.

GetRows

Declaration `public function GetRows(out AJSON: TJSONArray): TBrookQuery;`

Description Get all the rows of a query in a JSON array.

GetRow

Declaration `public function GetRow(out AJSON: TJSONObject): TBrookQuery;`

Description Get the current register in a JSON object.

SetRows

Declaration `public function SetRows(AJSON: TJSONArray): TBrookQuery;`

Description Inserts registers from a JSON array.

SetRow

Declaration `public function SetRow(AJSON: TJSONObject): TBrookQuery;`

Description Inserts one register from a JSON object.

Bind

Declaration `public function Bind(AJSON: TJSONObject): TBrookQuery;`

Description Binds a JSON object to the parameters.

ApplyUpdates

Declaration `public function ApplyUpdates: TBrookQuery; virtual; abstract;`

Description Applies all the changes stored in the buffer.

CancelUpdates

Declaration `public function CancelUpdates: TBrookQuery; virtual; abstract;`

Description Cancels all the updates stored in the buffer.

Apply

Declaration `public function Apply(const ARetaining: Boolean = False): TBrookQuery;
virtual; abstract;`

Description Applies the updates stored in the buffer and commits the transaction.

Undo

Declaration `public function Undo(const ARetaining: Boolean = False): TBrookQuery;
virtual; abstract;`

Description Undoes the updates stored in the buffer and rollbacks the transaction.

Commit

Declaration `public function Commit(const ARetaining: Boolean = False): TBrookQuery;
virtual; abstract;`

Description Commits the transaction.

Rollback

Declaration `public function Rollback(const ARetaining: Boolean = False): TBrookQuery;
virtual; abstract;`

Description Rollbacks the transaction.

Append

Declaration `public function Append(AJSON: TJSONObject): TBrookQuery;`

Description Adds a JSON object to the end of registers.

Insert

Declaration `public function Insert(AJSON: TJSONObject): TBrookQuery;`

Description Inserts a JSON object in the current position.

Edit

Declaration `public function Edit(AJSON: TJSONObject): TBrookQuery;`

Description Edits the current register by means of a JSON object.

Cancel

Declaration public function Cancel: TBrookQuery;

Description Cancels editions in the query.

Delete

Declaration public function Delete: TBrookQuery;

Description Deletes the current register.

Open

Declaration public function Open: TBrookQuery;

Description Opens the query.

Close

Declaration public function Close: TBrookQuery;

Description Closes the query.

Refresh

Declaration public function Refresh: TBrookQuery;

Description Refreshes the query.

First

Declaration public function First: TBrookQuery;

Description Goes to the query first register.

Prior

Declaration public function Prior: TBrookQuery;

Description Goes to the query previous register.

Next

Declaration public function Next: TBrookQuery;

Description Goes to the query next register.

Last

Declaration public function Last: TBrookQuery;

Description Goes to the query last register.

Post

Declaration public function Post: TBrookQuery;

Description Applies editions to the query.

Execute

Declaration public function Execute: TBrookQuery; virtual; abstract;

Description Executes the query.

RowsAffected

Declaration public function RowsAffected: TRowCount; virtual; abstract;

Description Get the number of changed registers.

Locate

Declaration public function Locate(const AJSON: TJSONObject; const AOptions: TLocateOptions = []): Boolean; overload;

Description Locates a register by means of JSON object.

Locate

Declaration public function Locate(const AKeyFields: string; const AKeyValues: Variant; const AOptions: TLocateOptions = []): Boolean; overload;

Description Locates a register passing a key and a value.

Param

Declaration public function Param(const AName: string): TParam; virtual; abstract;

Description Get a query parameter.

Field

Declaration public function Field(const AName: string): TField; virtual; abstract;

Description Get a query field.

FieldDef

Declaration `public function FieldDef(const AName: string): TFieldDef; virtual;
abstract;`

Description Get a query fielddef.

Count

Declaration `public function Count: Int64;`

Description Get the number of registers.

GetBookmark

Declaration `public function GetBookmark: TBookmark;`

Description Creates a bookmark.

GotoBookmark

Declaration `public procedure GotoBookmark(ABookmark: TBookmark);`

Description Goes to a bookmark.

17.5 Types

TBrookQueryClass

Declaration `TBrookQueryClass = class of TBrookQuery;`

Description Is a metaclass for TBrookQuery(17.4) class.

Chapter 18

Unit BrookRequestHelper

18.1 Description

Brook Request Helper unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

18.2 Uses

- `BrookUtils(24)`
- `BrookRouter(21)`
- `BrookConsts(6)`
- `BrookHTTPConsts(13)`
- `HttpDefs`
- `SysUtils`
- `Classes`
- `CustWeb`

18.3 Overview

`TBrookRequestHelper` Class Adds features to `TRequest` class.

18.4 Classes, Interfaces, Objects and Records

TBrookRequestHelper Class helper for TRequest

Hierarchy

TBrookRequestHelper > TObject

Description

Adds features to TRequest class.

Properties

Path public property Path[const AIndex: Integer]: string read GetPath;
Get the current path, script name and pathinfo included.

Methods

SetScheme

Declaration public class procedure SetScheme(const AScheme: ShortString);

Description Set the URL scheme (HTTP or HTTPS) in GetFullURL function.

GetScheme

Declaration public class function GetScheme: ShortString;

Description Get the current URL scheme (HTTP or HTTPS).

GetPath

Declaration public function GetPath(const AIndex: Integer): string;

Description Get the path level passing the respective index. Exemple:
BrookGetPathLevel('/a/b/c/', 1) = b.

IsGet

Declaration public function IsGet: Boolean;

Description Checks if the request was done under the GET method.

IsPost

Declaration public function IsPost: Boolean;

Description Checks if the request was done under the POST method.

IsPut

Declaration `public function IsPut: Boolean;`

Description Checks if the request was done under the PUT method.

IsDelete

Declaration `public function IsDelete: Boolean;`

Description Checks if the request was done under the DELETE method.

IsHead

Declaration `public function IsHead: Boolean;`

Description Checks if the request was done under the HEAD method.

IsOptions

Declaration `public function IsOptions: Boolean;`

Description Checks if the request was done under the OPTIONS method.

IsAjax

Declaration `public function IsAjax: Boolean;`

Description Checks if the request was done by means of Ajax.

IsXhr

Declaration `public function IsXhr: Boolean;`

Description Checks if the request was done by means of Xhr.

IsForm

Declaration `public function IsForm: Boolean;`

Description Checks if the request was done by means of a HTML form.

IsFormData

Declaration `public function IsFormData: Boolean;`

Description Checks if the request was done by means of a HTML data form.

OverriddenMethod

Declaration public function OverriddenMethod: ShortString;

Description Get X-HTTP-Method-Override header

GetHostWithPort

Declaration public function GetHostWithPort: string;

Description Get the URL and the port separated by a ":", exemple: http://localhost:80

GetPath

Declaration public function GetPath: string;

Description Get the current path.

GetFullURL

Declaration public function GetFullURL: string;

Description Get the full URL, schema included.

GetURLWithScriptName

Declaration public function GetURLWithScriptName: string;

Description Get the full URL, schema and script name included.

GetIP

Declaration public function GetIP: ShortString;

Description Get the IP of the requester.

GetDocumentRoot

Declaration public function GetDocumentRoot: string;

Description Get the server path where HTML documents can be placed.

Variable

Declaration public function Variable(const AName: string): string;

Description Get a environment variable given its name.

Variables

Declaration public procedure Variables(out AValues: TStrings);

Description Get a list with all environment variables.

Chapter 19

Unit BrookResponseHelper

19.1 Description

Brook Response Helper unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

19.2 Uses

- `BrookMessages(16)`
- `BrookConsts(6)`
- `BrookHTTPConsts(13)`
- `BrookHTTPUtils(14)`
- `BrookException(11)`
- `HTTPDefs`
- `Classes`
- `SysUtils`

19.3 Overview

`EBrookResponseHelper` Class Handles exceptions for `TBrookResponseHelper(19.4)`.

`TBrookResponseHelper` Class Adds features to `TResponse` class.

19.4 Classes, Interfaces, Objects and Records

EBrookResponseHelper Class

Hierarchy

EBrookResponseHelper > EBrook(11.4) > Exception

Description

Handles exceptions for TBrookResponseHelper(19.4).

TBrookResponseHelper Class helper for TResponse

Hierarchy

TBrookResponseHelper > TObject

Description

Adds features to TResponse class.

Properties

Header public property Header[AName: string]: string read GetHeader write SetHeader;
Get or set a HTTP header.

Status public property Status: Integer read GetStatus write SetStatus;
Get or set a HTTP status code.

Methods

IsEmpty

Declaration public function IsEmpty: Boolean;

Description Is this a empty response?

IsInformational

Declaration public function IsInformational: Boolean;

Description Is this an informational response?

IsOK

Declaration public function IsOK: Boolean;

Description Is this a 200 OK response?

IsSuccessful

Declaration `public function IsSuccessful: Boolean;`

Description Is this a 2xx successful response?

IsRedirection

Declaration `public function IsRedirection: Boolean;`

Description Is this a 3xx redirection response?

IsRedirect

Declaration `public function IsRedirect: Boolean;`

Description Is this a specific redirect response? (301, 302, 303, 307)

IsForbidden

Declaration `public function IsForbidden: Boolean;`

Description Is this a forbidden response?

IsNotFound

Declaration `public function IsNotFound: Boolean;`

Description Is this a 404 not found response?

IsClientError

Declaration `public function IsClientError: Boolean;`

Description Is this a client error response?

IsServerError

Declaration `public function IsServerError: Boolean;`

Description Is this a server error response?

AddContentDisposition

Declaration `public procedure AddContentDisposition(const AContentType: ShortString;
const AFileName: TFileName = ES; const ADispositionType: ShortString =
BROOK_HTTP_CONTENT_DISPOSITION_ATTACHMENT; const AContentDescription:
ShortString = ES; const AModificationDate: TDateTime = NullDate);`

Description Defines a content disposition for a file.

Chapter 20

Unit BrookRESTActions

20.1 Description

Brook REST Actions unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

20.2 Uses

- `BrookDBAction(8)`
- `BrookUtils(24)`
- `BrookHTTPConsts(13)`
- `HTTPDefs`
- `FPJSON`
- `SysUtils`

20.3 Overview

`TBrookOptionsAction` Class Displays the schema of the resource.

`TBrookRetrieveAction` Class Displays all the contents of the resource.

`TBrookShowAction` Class Displays the content of a specific resource.

TBrookCreateAction Class Creates a new resource.

TBrookUpdateAction Class Updates a specific resource.

TBrookDestroyAction Class Destroy a specific resource.

20.4 Classes, Interfaces, Objects and Records

TBrookOptionsAction Class

Hierarchy

TBrookOptionsAction > TBrookDBAction(8.4) > TBrookCustomDBAction(8.4) > TBrookAction(1.4) > TBrookObject(4.4) > TObject

Description

Displays the schema of the resource.

Methods

Register

Declaration `public class procedure Register(const ATableName, APattern: string; const AMethod: TBrookRequestMethod = rmOptions; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table.

Register

Declaration `public class procedure Register(const ATableName, APattern, AIgnoredFields: string; const AMethod: TBrookRequestMethod = rmOptions; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table and defining the fields that will be ignored for persistence purposes.

Request

Declaration `public procedure Request(ARequest: TRequest; AResponse: TResponse); override;`

Description Is triggered by a OPTIONS HTTP request method.

Execute

Declaration `public function Execute: Boolean; virtual;`

Description Executes the action. If there are schema in the resource, they are returned, if not, it returns a '{ "error": "No schema." }' JSON.

TBrookRetrieveAction Class ---

Hierarchy

TBrookRetrieveAction > TBrookDBAction(8.4) > TBrookCustomDBAction(8.4) > TBrookAction(1.4) > TBrookObject(4.4) > TObject

Description

Displays all the contents of the resource.

Methods

Register

Declaration `public class procedure Register(const ATableName, APattern: string; const AMethod: TBrookRequestMethod = rmGet; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table.

Register

Declaration `public class procedure Register(const ATableName, APattern, AIgnoredFields: string; const AMethod: TBrookRequestMethod = rmGet; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table and defining the fields that will be ignored for persistence purposes.

Request

Declaration `public procedure Request(ARequest: TRequest; AResponse: TResponse); override;`

Description Is triggered by a request of any HTTP method.

Execute

Declaration `public function Execute: Boolean; virtual;`

Description Executes the action. If there are contents in the resource, they are returned, if not, it returns a 404 status code.

TBrookShowAction Class ---

Hierarchy

TBrookShowAction > TBrookDBAction(8.4) > TBrookCustomDBAction(8.4) > TBrookAction(1.4) > TBrookObject(4.4) > TObject

Description

Displays the content of a specific resource.

Methods

Register

Declaration `public class procedure Register(const ATableName, APattern: string; const AMethod: TBrookRequestMethod = rmGet; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table.

Register

Declaration `public class procedure Register(const ATableName, APattern, AIgnoredFields: string; const AMethod: TBrookRequestMethod = rmGet; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table and defining the fields that will be ignored for persistence purposes.

Request

Declaration `public procedure Request(ARequest: TRequest; AResponse: TResponse); override;`

Description Is triggered by a request of any HTTP method.

Execute

Declaration `public function Execute: Boolean; virtual;`

Description Executes the action. If there are contents in the resource, they are returned, if not, it returns a 404 status code.

TBrookCreateAction Class

Hierarchy

TBrookCreateAction > TBrookDBAction(8.4) > TBrookCustomDBAction(8.4) > TBrookAction(1.4) > TBrookObject(4.4) > TObject

Description

Creates a new resource.

Methods

Register

Declaration `public class procedure Register(const ATableName, APattern: string; const AMethod: TBrookRequestMethod = rmPost; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table.

Register

Declaration `public class procedure Register(const ATableName, APattern, AIgnoredFields: string; const AMethod: TBrookRequestMethod = rmPost; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table and defining the fields that will be ignored for persistence purposes.

Request

Declaration `public procedure Request(ARequest: TRequest; AResponse: TResponse); override;`

Description Is triggered by a request of any HTTP method.

Execute

Declaration `public function Execute: Boolean; virtual;`

Description Executes the action. The 201 status code is always returned.

TBrookUpdateAction Class

Hierarchy

TBrookUpdateAction > TBrookDBAction(8.4) > TBrookCustomDBAction(8.4) > TBrookAction(1.4) > TBrookObject(4.4) > TObject

Description

Updates a specific resource.

Methods

Register

Declaration `public class procedure Register(const ATableName, APattern: string; const AMethod: TBrookRequestMethod = rmPut; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table.

Register

Declaration `public class procedure Register(const ATableName, APattern, AIgnoredFields: string; const AMethod: TBrookRequestMethod = rmPut; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table and defining the fields that will be ignored for persistence purposes.

Request

Declaration `public procedure Request(ARequest: TRequest; AResponse: TResponse); override;`

Description Is triggered by a request of any HTTP method.

Execute

Declaration `public function Execute: Boolean; virtual;`

Description Executes the action. If the edition is successful, the 204 status code is returned, if not, the code is 201.

TBrookDestroyAction Class

Hierarchy

TBrookDestroyAction > TBrookDBAction(8.4) > TBrookCustomDBAction(8.4) > TBrookAction(1.4) > TBrookObject(4.4) > TObject

Description

Destroy a specific resource.

Methods

Register

Declaration `public class procedure Register(const ATableName, APattern: string; const AMethod: TBrookRequestMethod = rmDelete; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table.

Register

Declaration `public class procedure Register(const ATableName, APattern, AIgnoredFields: string; const AMethod: TBrookRequestMethod = rmDelete; const ADefault: Boolean = False); overload;`

Description Registers an action linking the request to a database table and defining the fields that will be ignored for persistence purposes.

Request

Declaration `public procedure Request(ARequest: TRequest; AResponse: TResponse);
override;`

Description Is triggered by a request of any HTTP method.

Execute

Declaration `public function Execute: Boolean; virtual;`

Description Executes the action. If the deletion is successful, the 204 status code is returned, if not, the code is 404.

Chapter 21

Unit BrookRouter

21.1 Description

Brook Router unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

21.2 Uses

- BrookClasses(4)
- BrookException(11)
- BrookAction(1)
- BrookUtils(24)
- BrookConsts(6)
- BrookMessages(16)
- BrookHTTPConsts(13)
- HTTPDefs
- FPJSON
- Classes
- SysUtils
- StrUtils

21.3 Overview

E BrookRoutes Class Handles exceptions for **T BrookRoutes**(21.4).

E BrookRouter Class Handles exceptions for **T BrookRouter**(21.4).

T BrookRoute record Defines a route item.

T BrookRoutes Class Defines a list of routes.

T BrookRouter Class Provides features for the route handling.

21.4 Classes, Interfaces, Objects and Records

E BrookRoutes Class

Hierarchy

E BrookRoutes > **E Brook**(11.4) > Exception

Description

Handles exceptions for **T BrookRoutes**(21.4).

E BrookRouter Class

Hierarchy

E BrookRouter > **E Brook**(11.4) > Exception

Description

Handles exceptions for **T BrookRouter**(21.4).

T BrookRoute record

Description

Defines a route item.

Fields

ActionClass **ActionClass:** **T BrookActionClass;**
 Specifies the class of the action to be called.

Method **Method:** **T BrookRequestMethod;**
 Specifies a HTTP request method of the action to be called.

Pattern **Pattern:** **string;**
 Specifies the patter of the action to be called.

TBrookRoutes Class

Hierarchy

TBrookRoutes > TBrookObject(4.4) > TObject

Description

Defines a list of routes.

Properties

Items public property Items[const AIndex: Integer]: PBrookRoute read GetItems write SetItems;

The list of routes.

List public property List: TFPList read FList;

The router pointers list.

Methods

Create

Declaration public constructor Create;

Description Creates an instance of a TBrookRoutes(21.4) class.

Destroy

Declaration public destructor Destroy; override;

Description Frees an instance of TBrookRoutes(21.4) class.

Count

Declaration public function Count: Integer;

Description Returns the number of registered routes.

Add

Declaration public function Add(AActionClass: TBrookActionClass; const APattern: string; const AMethod: TBrookRequestMethod; const ADefault: Boolean): Integer;

Description Adds a route item.

GetDefaultActionClass

Declaration `public procedure GetDefaultActionClass(out AClass: TBrookActionClass; out AIndex: Integer);`

Description Get the default action class.

PatternByActionClass

Declaration `public function PatternByActionClass(AClass: TBrookActionClass): string;`

Description Get the registered pattern of a class.

ActionClassByPattern

Declaration `public function ActionClassByPattern(const APattern: string): TBrookActionClass;`

Description Get the action class from a patter.

ActionClassByClassName

Declaration `public function ActionClassByClassName(const AName: string): TBrookActionClass;`

Description Get an action class from its class name.

TBrookRouter Class

Hierarchy

TBrookRouter > TBrookObject(4.4) > TObject

Description

Provides features for the route handling.

Properties

Routes `public property Routes: TBrookRoutes read FRoutes write FRoutes;`
List of available routes.

AfterExecuteAction `public property AfterExecuteAction: TBrookExecuteActionEvent read FAfterExecuteAction write FAfterExecuteAction;`
Is triggered after the router executes a action.

AfterMatchPattern `public property AfterMatchPattern: TBrookMatchPatternEvent read FAfterMatchPattern write FAfterMatchPattern;`
Is triggered after the router matches a pattern.

AfterRoute `public property AfterRoute: TBrookRouteEvent read FAfterRoute
write FAfterRoute;`
Is triggered after the router is routing.

BeforeExecuteAction `public property BeforeExecuteAction: TBrookExecuteActionEvent read
FBeforeExecuteAction write FBeforeExecuteAction;`
Is triggered before the router executes a action.

BeforeMatchPattern `public property BeforeMatchPattern: TBrookMatchPatternEvent read
FBeforeMatchPattern write FBeforeMatchPattern;`
Is triggered before the router matches a pattern.

BeforeRoute `public property BeforeRoute: TBrookRouteEvent read FBeforeRoute
write FBeforeRoute;`
Is triggered before the router is routing.

OnExecuteAction `public property OnExecuteAction: TBrookExecuteActionEvent read
FOnExecuteAction write FOnExecuteAction;`
Is triggered when the router executes a action.

Methods

Create

Declaration `public constructor Create; virtual;`

Description Creates an instance of a TBrookRouter(21.4) class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of TBrookRouter(21.4) class.

GetServiceClass

Declaration `public class function GetServiceClass: TBrookRouterClass;`

Description Return the service class provided by this class.

RegisterService

Declaration `public class procedure RegisterService;`

Description Registers the service provided by this class.

UnregisterService

Declaration `public class procedure UnregisterService;`

Description Unregisters the service provided by this class.

Service

Declaration `public class function Service: TBrookRouter;`

Description Return a instance of this class.

RootUrl

Declaration `public class function RootUrl: string;`

Description Return the root URL.

RootUrl

Declaration `public class function RootUrl(ARequest: TRequest): string;`

Description Return the root URL passing TRequest as param.

MethodNotAllowed

Declaration `public class procedure MethodNotAllowed(AResponse: TResponse);`

Description Sends the HTTP "NotAllowed" status code to the response.

UrlFor

Declaration `public function UrlFor(AActionClass: TBrookActionClass; const AParams: array of string): string; overload;`

Description Creates an URL for an action informing an array of parameters. Exemple:

```
procedure TMyAction.Get;
begin
    // When calling with http://localhost/cgi-bin/cgi1/foo/myvalue
    // the output will be /cgi-bin/cgi1/foo/myvalue
    Write(UrlFor(TMyAction, ['myvalue']));
end;

initialization
    TMyAction.Register('/foo/:myvar')
```

UrlFor

Declaration `public function UrlFor(AClassName: string; const AParams: array of string): string; overload;`

Description Creates an URL for an action passing an array of parameters however informing the class name as string

UrlFor

Declaration `public function UrlFor(AActionClass: TBrookActionClass; const AParams: TJSONData): string; overload;`

Description Creates an URL for an action passing a JSON data.

UrlFor

Declaration `public function UrlFor(AClassName: string; const AParams: TJSONData): string; overload;`

Description Creates an URL for an action passing a JSON data however informing the class name as string.

Canonicalize

Declaration `public function Canonicalize(ARequest: TRequest; AResponse: TResponse): Boolean;`

Description Adds an slash to the end of the URL if does not exist.

MatchPattern

Declaration `public function MatchPattern(APattern, APathInfo: string; out ARedirect: Boolean; out ANames, AValues: TBrookArrayOfString): Boolean; virtual;`

Description Checks if the given parameters match with a registered route.

Route

Declaration `public procedure Route(ARequest: TRequest; AResponse: TResponse); virtual;`

Description Runs the route processing.

21.5 Types

TBrookRoutesClass

Declaration `TBrookRoutesClass = class of TBrookRoutes;`

Description Is a metaclass for TBrookRoutes(21.4) class.

TBrookRouterClass

Declaration TBrookRouterClass = class of TBrookRouter;

Description Is a metaclass for TBrookRouter(21.4) class.

PBrookRoute

Declaration PBrookRoute = ^TBrookRoute;

Description Defines a pointer to the route item.

TBrookMatchPatternEvent

Declaration TBrookMatchPatternEvent = procedure(ASender: TObject; APattern, APathInfo: string; out ARedirect: Boolean; out ANames, AValues: TBrookArrayOfString) of object;

Description Is a type to *MatchPattern event.

TBrookRouteEvent

Declaration TBrookRouteEvent = procedure(ASender: TObject; ARequest: TRequest; AResponse: TResponse) of object;

Description Is a type to *Route event.

TBrookExecuteActionEvent

Declaration TBrookExecuteActionEvent = procedure(ASender: TObject; AAction: TBrookAction; ARequest: TRequest; AResponse: TResponse; ARoute: TBrookRoute; var AHandled: Boolean) of object;

Description Is a type to *ExecuteAction event.

Chapter 22

Unit BrookSession

22.1 Description

Brook Session unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

22.2 Uses

- `BrookClasses(4)`
- `BrookUtils(24)`
- `BrookException(11)`
- `BrookConsts(6)`
- `FPJSON`
- `JSONParser`
- `Classes`
- `SysUtils`
- `HTTPDefs`
- `DateUtils`

22.3 Overview

EBrookSession Class Handles exceptions for **TBrookSession**(22.4).

TBrookSession Class Defines features to the section handling.

22.4 Classes, Interfaces, Objects and Records

EBrookSession Class

Hierarchy

EBrookSession > EBrook(11.4) > Exception

Description

Handles exceptions for **TBrookSession**(22.4).

TBrookSession Class

Hierarchy

TBrookSession > TBrookObject(4.4) > TObject

Description

Defines features to the section handling.

Properties

CookieName public property CookieName: string read FCookieName write FCookieName;
Set the session cookie name.

Directory public property Directory: string read FDirectory write FDirectory;
Set the name of directory session.

Expired public property Expired: Boolean read IsExpired;
Returns **True** if the session has expired.

SID public property SID: string read FSID;
Get the ID session.

Started public property Started: Boolean read IsStarted;
Checks if the session has started.

FileName public property FileName: TFileName read FFileName write FFileName;
The session file name.

FilePrefix `public property FilePrefix: ShortString read FFilePrefix write FFilePrefix;`
 The session file prefix.

Fields `public property Fields: TJSONObject read FFields;`
 The fields session.

TimeOut `public property TimeOut: Integer read FTimeOut write FTimeOut;`
 The remaining seconds for the session end.

HttpOnly `public property HttpOnly: Boolean read FHttpOnly write FHttpOnly;`
 Informs if the session cookie is accessible only by HTTP requests, if `True`, the JavaScript access is not allowed.

Methods

Create

Declaration `public constructor Create; virtual;`

Description Creates an instance of a `TBrookSession(22.4)` class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of `TBrookSession(22.4)` class.

IsExpired

Declaration `public function IsExpired: Boolean; virtual;`

Description Returns `True` if the session has expired.

GenerateID

Declaration `public function GenerateID: string; virtual;`

Description Creates an ID for the session.

Start

Declaration `public procedure Start(ARequest: TRequest); virtual;`

Description Starts the session.

Finish

Declaration `public procedure Finish(AResponse: TResponse); virtual;`

Description Terminates the session.

Expire

Declaration `public procedure Expire(ARequest: TRequest; AResponse: TResponse); virtual;`

Description Expires the session.

DeleteFiles

Declaration `public procedure DeleteFiles;`

Description Deletes the session files.

Exists

Declaration `public function Exists(const AName: string): Boolean;`

Description Checks if a name exists in fields.

DeleteOldFiles

Declaration `public procedure DeleteOldFiles(const ABeforeOf: TDateTime);`

Description Deletes the session files except for the current session file.

22.5 Types

TBrookSessionClass

Declaration `TBrookSessionClass = class of TBrookSession;`

Description Is a metaclass for TBrookSession(22.4) class.

Chapter 23

Unit BrookTable

23.1 Description

Brook Table unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Plase see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

23.2 Uses

- `BrookClasses`(4)
- `BrookDataBase`(7)
- `BrookQuery`(17)
- `BrookException`(11)
- `BrookMessages`(16)
- `BrookConsts`(6)
- `BrookDBConsts`(9)
- `DB`
- `SQLdb`
- `FPJSON`
- `SysUtils`

23.3 Overview

EBrookTable Class Handles exceptions for **TBrookTable**(23.4).

TBrookTable Class Handles database tables.

23.4 Classes, Interfaces, Objects and Records

EBrookTable Class

Hierarchy

EBrookTable > EBrook(11.4) > Exception

Description

Handles exceptions for **TBrookTable**(23.4).

TBrookTable Class

Hierarchy

TBrookTable > TBrookObject(4.4) > TObject

Description

Handles database tables.

Properties

Name	public property Name: string read FName write FName; Defines the table name.
BOF	public property BOF: Boolean read GetBOF; Checks if the cursor is at the first register.
EOF	public property EOF: Boolean read GetEOF; Checks if the cursor is at the last register.
Bookmark	public property Bookmark: TBookmark read GetBookmark write GotoBookmark; Returns or creates a bookmark.
Modified	public property Modified: Boolean read GetModified; Returns True if the content of the table has changed.
State	public property State: TDataSetState read GetState; Get the current table state.

Active	public property Active: Boolean read GetActive write SetActive; Checks if the table is active.
DataSet	public property DataSet: TDataSet read GetDataSet; Get the dataset of the table.
FieldDefs	public property FieldDefs: TFieldDefs read GetFieldDefs; Get the fieldefs of the table.
DataSource	public property DataSource: TDataSource read GetDataSource write SetDataSource; Get or set a master table datasource.
DataBase	public property DataBase: TBrookDataBase read GetDataBase write SetDataBase; Get or set a table database.
DateAsString	public property DateAsString: Boolean read GetDateAsString write SetDateAsString; Enables the date saving as string.
AsJSON	public property AsJSON: TJSONStringType read GetAsJSON write SetAsJSON; Returns a JSON string with the table registers.
Rows	public property Rows: TJSONArray read GetRows; Returns a JSON array with the table registers.
Row	public property Row: TJSONObject read GetRow; Returns the current register in a JSON object.
Columns	public property Columns[const AName: string]: TField read GetColumns; The list of table columns.
Prepared	public property Prepared: Boolean read IsPrepared; Checks if a statement preparation is set.
Fields	public property Fields: TFields read GetFields; Are the table fields.
Params	public property Params: TParams read GetParams; Are the table parameters.
Position	public property Position: LongInt read GetPosition write SetPosition; Get the position of the current register.

Methods

Create

Declaration `public constructor Create(ADataBase: TBrookDataBase; const ATableName: string = ES); virtual;`

Description Creates an instance of a TBrookTable(23.4) class.

Create

Declaration `public constructor Create(const ATableName: string = ES); reintroduce;`

Description Creates an instance of a TBrookTable(23.4) class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of TBrookTable(23.4) class.

GetConditions

Declaration `public class procedure GetConditions(AJSON: TJSONObject; out AConditions: string);`

Description Receives an JSON object with parameters and creates a "SQL where" condition.

GetSchema

Declaration `public procedure GetSchema(out ASchema: TJSONObject); overload;`

Description Returns a JSON object with the table columns.

GetSchema

Declaration `public function GetSchema: TJSONStringType; overload;`

Description Returns a JSON string with the table columns.

CreateFields

Declaration `public function CreateFields(AJSON: TJSONObject): TBrookTable;`

Description Creates fielddefs from a JSON Object.

GetRows

Declaration `public function GetRows(out AJSON: TJSONArray): TBrookTable;`

Description Get all the rows of a query in a JSON array.

GetRow

Declaration `public function GetRow(out AJSON: TJSONObject): TBrookTable;`

Description Get the current register in a JSON object.

SetRows

Declaration `public function SetRows(AJSON: TJSONArray): TBrookTable;`

Description Inserts registers from a JSON array.

SetRow

Declaration `public function SetRow(AJSON: TJSONObject): TBrookTable;`

Description Inserts one register from a JSON object.

Bind

Declaration `public function Bind(AJSON: TJSONObject): TBrookTable;`

Description Binds a JSON object to the parameters.

ApplyUpdates

Declaration `public function ApplyUpdates: TBrookTable;`

Description Applies all the changes stored in the buffer.

CancelUpdates

Declaration `public function CancelUpdates: TBrookTable;`

Description Cancels all the updates stored in the buffer.

Apply

Declaration `public function Apply(const ARetaining: Boolean = False): TBrookTable;`

Description Applies the updates stored in the buffer and commits the transaction.

Undo

Declaration `public function Undo(const ARetaining: Boolean = False): TBrookTable;`

Description Undoes the updates stored in the buffer and rollbacks the transaction.

Commit

Declaration `public function Commit(const ARetaining: Boolean = False): TBrookTable;`

Description Commits the transaction.

Rollback

Declaration `public function Rollback(const ARetaining: Boolean = False): TBrookTable;`

Description Rolls back the transaction.

Append

Declaration `public function Append(AJSON: TJSONObject): TBrookTable;`

Description Adds a JSON object to the end of registers.

Insert

Declaration `public function Insert(AJSON: TJSONObject): TBrookTable;`

Description Inserts a JSON object in the current position.

Edit

Declaration `public function Edit(AJSON: TJSONObject): TBrookTable;`

Description Edits the current register by means of a JSON object.

Cancel

Declaration `public function Cancel: TBrookTable;`

Description Cancels editions in the table.

Delete

Declaration `public function Delete: TBrookTable;`

Description Deletes the current register.

Open

Declaration `public function Open: TBrookTable;`

Description Opens the table.

Close

Declaration public function Close: TBrookTable;

Description Closes the table.

Refresh

Declaration public function Refresh: TBrookTable;

Description Refreshes the table.

First

Declaration public function First: TBrookTable;

Description Goes to the table first register.

Prior

Declaration public function Prior: TBrookTable;

Description Goes to the table previous register.

Next

Declaration public function Next: TBrookTable;

Description Goes to the table next register.

Last

Declaration public function Last: TBrookTable;

Description Goes to the table last register.

Clear

Declaration public function Clear: TBrookTable;

Description Clears the SQL statements and close the table.

Reset

Declaration public function Reset: TBrookTable;

Description Clears the SQL statements, close the table and clears the fielddefs.

Post

Declaration public function Post: TBrookTable;

Description Applies editions to the query.

Locate

Declaration public function Locate(AJSON: TJSONObject; const AOptions: TLocateOptions = []): Boolean; overload;

Description Locates a register by means of JSON object.

Locate

Declaration public function Locate(const AKeyFields: string; const AKeyValues: Variant; const AOptions: TLocateOptions = []): Boolean; overload;

Description Locates a register passing a key and a value.

Find

Declaration public function Find(AJSON: TJSONObject; const AOptions: TLocateOptions = []): TBrookTable;

Description Finds a register by means of JSON object.

Get

Declaration public function Get(const AKeyValue: Variant): TBrookTable; overload;

Description Gets a register passing a key value.

Get

Declaration public function Get(const AKeyName: string; const AKeyValue: Variant): TBrookTable; overload;

Description Gets a register passing a key and a value.

Field

Declaration public function Field(const AName: string): TField;

Description Get a table field.

Param

Declaration public function Param(const AName: string): TParam;

Description Get a table parameter.

FieldDef

Declaration `public function FieldDef(const AName: string): TFieldDef;`

Description Get a table fielddef.

Conditions

Declaration `public function Conditions(AJSON: TJSONObject): TBrookTable;`

Description Receives an JSON object with parameters and creates a "SQL where" condition.

Count

Declaration `public function Count: Int64;`

Description Get the number of registers.

RowsAffected

Declaration `public function RowsAffected: TRowsCount;`

Description Get the number of changed registers.

Prepare

Declaration `public function Prepare: TBrookTable;`

Description Composes a SQL statement.

Unprepare

Declaration `public function Unprepare: TBrookTable;`

Description Clears a SQL statement preparation.

Select

Declaration `public function Select(const AFields: string): TBrookTable;`

Description Composes a "SELECT" statement.

Where

Declaration `public function Where(const AConditions: string): TBrookTable; overload;`

Description Composes a "WHERE" statement.

Where

Declaration `public function Where(const AConditions: string; const AArgs: array of const): TBrookTable; overload;`

Description Composes a parametrized "WHERE" statement.

OrderBy

Declaration `public function OrderBy(const AFields: string): TBrookTable;`

Description Composes a "ORDER BY" statement.

Empty

Declaration `public function Empty: Boolean;`

Description Checks if table is empty.

GetBookmark

Declaration `public function GetBookmark: TBookmark;`

Description Creates a bookmark.

GotoBookmark

Declaration `public procedure GotoBookmark(ABookmark: TBookmark);`

Description Goes to a bookmark.

23.5 Types

TBrookTableClass

Declaration `TBrookTableClass = class of TBrookTable;`

Description Is a metaclass for TBrookTable(23.4) class.

Chapter 24

Unit BrookUtils

24.1 Description

Brook Utils unit.

Copyright (C) 2013 Silvio Clecio.

<http://brookframework.org>

All contributors: Please see the file CONTRIBUTORS.txt, included in this distribution.

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

24.2 Uses

- `BrookException(11)`
- `BrookMessages(16)`
- `BrookConsts(6)`
- `BrookHTTPConsts(13)`
- `HTTPDefs`
- `FPJSON`
- `CustWeb`
- `SysUtils`

24.3 Overview

`TBrookSettings` record Stores the Brook configurations.

BrookStartsChar Check whether a string starts with a given character.

BrookEndsChar Check whether a string ends with a given character.

BrookExtractPathLevels Get the next pathinfo level.

BrookGetPathLevel Get the path level passing the respective index.

BrookGetPathLevels Get the path from the level correspondent to the index to the last level.

BrookMatchMethod Checks if a string is equivalent a enumerator representing a HTTP request method.

BrookDeleteFiles Deletes files according to their creation dates and file names.

BrookFileDate Get the datetime of a file.

BrookFileSetDate Set the datetime of a file.

BrookJSONCopy Copy the content of a JSON Object to another.

BrookDumpStack Writes a backtrace of the current exception.

BrookExcludeHTTPPathDelimiter Ensures URL ends with delimiter.

24.4 Classes, Interfaces, Objects and Records

TBrookSettings record

Description

Stores the Brook configurations.

Fields

Mapped	Mapped: Boolean; Enables the mapping of HTTP request methods.
Charset	Charset: ShortString; Set the default application Charset.
ContentType	ContentType: ShortString; Set the default application Content-Type.
Language	Language: ShortString; Set the default application Language.
Page404	Page404: string; Set the 404 HTML page. The string will be sent as is.

Page404File	Page404File: <code>string</code> ; Set the 404 HTML page file. The file content will be sent. This has higher precedence than <code>TBrookSettings.Page404</code> so when both are set, this will be processed first and only if the file is not found or cannot be read the system will fallback to <code>TBrookSettings.Page404</code>
Page500	Page500: <code>string</code> ; Set the 500 HTML page. The string will be sent as is.
Page500File	Page500File: <code>string</code> ; Set the 500 HTML page file. The file content will be sent. This has higher precedence than <code>TBrookSettings.Page500</code> so when both are set, this will be processed first and only if the file is not found or cannot be read the system will fallback to <code>TBrookSettings.Page500</code>
DirectoryForUploads	DirectoryForUploads: <code>string</code> ; Set the default directory for uploads.
DeleteUploadedFiles	DeleteUploadedFiles: <code>Boolean</code> ; Defines if the temporary uploaded files will be deleted.
KeepUploadedNames	KeepUploadedNames: <code>Boolean</code> ; Keeps the original name of the uploaded files.
AcceptsJSONContent	AcceptsJSONContent: <code>Boolean</code> ; Defines if the application allows JSON formatted requests.
Configuration	Configuration: <code>string</code> ; Set a configuration for the application or for its object members.
RootUrl	RootUrl: <code>string</code> ; Set the default root URL. This is used by methods such as <code>TBrookAction.UrlFor</code> , <code>TBrookActionHelper.LinkTo</code> , <code>TBrookActionHelper.ButtonTo</code> etc. By default, Brook assumes <code>SCRIPT_NAME</code> as root URL.
Port	Port: <code>Word</code> ; Set the default application port.
OnError	OnError: <code>TOnShowRequestException</code> ; Handles the application exceptions.

24.5 Functions and Procedures

BrookStartsChar

Declaration `function BrookStartsChar(const Ch: Char; const S: string): Boolean;`

Description Check whether a string starts with a given character.

BrookEndsChar

Declaration `function BrookEndsChar(const Ch: Char; const S: string): Boolean;`

Description Check whether a string ends with a given character.

BrookExtractPathLevels

Declaration `procedure BrookExtractPathLevels(S: string; var R: string; out ALvl: string; out AEndDelim: Boolean; const ADelimiter: Char = US);`

Description Get the next pathinfo level.

BrookGetPathLevel

Declaration `function BrookGetPathLevel(const APath: string; const AIndex: SizeInt = 0; const ADelimiter: Char = US): string;`

Description Get the path level passing the respective index. Exemple:

`BrookGetPathLevel('/a/b/c/', 1) = b.`

BrookGetPathLevels

Declaration `function BrookGetPathLevels(const APath: string; const AIndex: SizeInt = 0; const ADelimiter: Char = US): string;`

Description Get the path from the level correspondent to the index to the last level. Exemple:

`BrookGetPathLevels('/a/b/c/', 1) = b/c/.`

BrookMatchMethod

Declaration `function BrookMatchMethod(const ABrookMethod: TBrookRequestMethod; const AMethod: string): Boolean;`

Description Checks if a string is equivalent a enumerator representing a HTTP request method.

BrookDeleteFiles

Declaration `procedure BrookDeleteFiles(APath: string; const ABeforeOf: TDateTime; const ASkippedFile: TFileName = ES; const AContains: string = ES);`

Description Deletes files according to their creation dates and file names.

BrookFileDate

Declaration `function BrookFileDate(const AFileName: TFileName): TDateTime;`

Description Get the datetime of a file.

BrookFileSetDate

Declaration `function BrookFileSetDate(const AFileName: TFileName; const ADateTime: TDateTime): LongInt;`

Description Set the datetime of a file.

BrookJSONCopy

Declaration `procedure BrookJSONCopy(ASrc, ADest: TJSONObject);`

Description Copy the content of a JSON Object to another.

BrookDumpStack

Declaration `function BrookDumpStack(const AEOL: ShortString = BR): string;`

Description Writes a backtrace of the current exception.

BrookExcludeHTTPPathDelimiter

Declaration `function BrookExcludeHTTPPathDelimiter(const AUrl: string): string;`

Description Ensures URL ends with delimiter.

24.6 Types

TBrookArrayOfString

Declaration `TBrookArrayOfString = array of string;`

Description Defines a array of strings.

TBrookRequestMethod

Declaration `TBrookRequestMethod = (...);`

Description Defines an enumerator to represent the HTTP request methods.

Values

- `rmUnknown`
- `rmAll`
- `rmGet`
- `rmPost`
- `rmPut`
- `rmDelete`
- `rmHead`
- `rmOptions`
- `rmTrace`

24.7 Variables

BrookSettings

Declaration BrookSettings: TBrookSettings = (Mapped: False; Charset: BROOK_HTTP_CHARSET_UTF_8; ContentType: BROOK_HTTP_CONTENT_TYPE_TEXT_HTML; Language: BROOK_DEFAULT_LANGUAGE; Page404: BROOK_HTTP_RESPONSE_TEMPLATE_NOT_FOUND; Page404File: ES; Page500: BROOK_HTTP_RESPONSE_TEMPLATE_INTERNAL_SERVER_ERROR; Page500File: ES; DirectoryForUploads: ES; DeleteUploadedFiles: False; KeepUploadedNames: True; AcceptsJSONContent: False; Configuration: ES; RootUrl: ES; Port: 0; OnError: nil;);

Description Global variable to store Brook settings.