

DocCoreMVChannel

NITROMVC - COREBASE - NOTIONCUBESOFTWARE GMBH

Table of Contents

Symbol Reference	1
Classes	1
AsanaAPI Class	1
CoreChannel Class	2
CoreChannelController Class	3
CoreChannelController Methods	3
CoreChannelController::Event Method	4
CoreChannelController::Trigger Method	6
CoreService Class	8
CoreService Methods	9
Array Method	9
CoreService::Param Method	10
CoreService::Value Method	10
CoreService::convertToJson Method	10
CoreService::convertToParam Method	11
CoreService::firstUpper Method	12
CoreService::get Method	13
CoreService::post Method	14
CoreService::rawData Method	16
DropBoxAPI Class	16
FacebookAPI Class	17
GoogleCalendarAPI Class	18
GooglePlacesAPI Class	19
GoogleSheetsAPI Class	21
GoogleTranslateAPI Class	22
HubSpotAPI Class	23
RSSFeed Class	24
SalesForceAPI Class	25
SlackAPI Class	25
TrelloAPI Class	27
TwitterAPI Class	28
WrikeAPI Class	29
Variables	30
DropBoxAPI::m_LocalPath Variable	30
DropBoxAPI::m_RemotePath Variable	31
HubSpotAPI::m_GlobalAPIKey Variable	31
TrelloAPI::m_GlobalAPIKey Variable	31
TrelloAPI::m_GlobalToken Variable	31

hookurl Variable	32
targeturl Variable	32
Files	32
AsanaAPI.cpp	33
AsanaAPI.h	33
CoreChannel.cpp	33
CoreChannel.h	34
CoreChannelController.cpp	34
CoreChannelController.h	34
CoreService.cpp	34
CoreService.h	35
DropBoxAPI.cpp	35
DropBoxAPI.h	35
FacebookAPI.cpp	35
FacebookAPI.h	36
GoogleCalendarAPI.cpp	36
GoogleCalendarAPI.h	36
GooglePlacesAPI.cpp	36
GooglePlacesAPI.h	37
GoogleSheetsAPI.cpp	37
GoogleSheetsAPI.h	37
GoogleTranslateAPI.cpp	37
GoogleTranslateAPI.h	38
HubSpotAPI.cpp	38
HubSpotAPI.h	38
RSSFeed.cpp	38
RSSFeed.h	39
SalesForceAPI.cpp	39
SalesForceAPI.h	39
SlackAPI.cpp	39
SlackAPI.h	40
TrelloAPI.cpp	40
TrelloAPI.h	40
TwitterAPI.cpp	40
TwitterAPI.h	41
WrikeAPI.cpp	41
WrikeAPI.h	41

Index


















a

1 Symbol Reference


1.1 Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	AsanaAPI (see page 1)	
	CoreChannel (see page 2)	
	CoreChannelController (see page 3)	
	CoreService (see page 8)	
	DropBoxAPI (see page 16)	
	FacebookAPI (see page 17)	
	GoogleCalendarAPI (see page 18)	
	GooglePlacesAPI (see page 19)	
	GoogleSheetsAPI (see page 21)	
	GoogleTranslateAPI (see page 22)	
	HubSpotAPI (see page 23)	
	RSSFeed (see page 24)	
	SalesForceAPI (see page 25)	
	SlackAPI (see page 25)	
	TrelloAPI (see page 27)	
	TwitterAPI (see page 28)	
	WrikeAPI (see page 29)	

Legend

	Class
---	-------

1.1.1 AsanaAPI Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

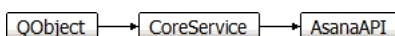
Date

19.01.2018

C++

```
class AsanaAPI : public CoreService;
```

Inheritance



Group

Classes (see page 1)

Legend

	Method
	protected

Members

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Service

Method

AsanaAPI

Class

AsanaAPI

1.1.2 CoreChannel Class

C++

```
class CoreChannel : public QObject;
```

Inheritance



Description

This is class CoreChannel.

Group

Classes ([↗](#) see page 1)

1.1.3 CoreChannelController Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

25.01.2018

C++

```
class CoreChannelController : public CoreApiController;
```

Inheritance**Group**

Classes ([↗](#) see page 1)

Legend

	Method
--	--------

Members**CoreChannelController Methods**

	Name	Description
	Event (↗ see page 4)	
	Trigger (↗ see page 6)	

CoreChannelController Methods

	Name	Description
	Event (↗ see page 4)	
	Trigger (↗ see page 6)	

Context

Controller

Method

CoreChannelController

Class

CoreChannelController

1.1.3.1 CoreChannelController Methods

1.1.3.1.1 CoreChannelController::Event Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

28.01.2018

Body Source

```
1: bool CoreChannelController::Event(const QString&type, const QString&title, const
QString&message, const QMap<QString, QString>&valuemap, const QString&jsonevent)
2: {
3:     SlackAPI slack;
4:     slack.postMessage("mails", title + ":" + message, type);
5:
6:     //CoreMail mail;
7:     //mail.send(title, message);
8:
9:     if (!targeturl.isEmpty())
10:    {
11:        CoreChannelController channel;
12:        channel.post(targeturl, QString("[{ \\Name\\":\\" + title +
QDateTime::currentDateTime().toString() + "\\}]]"));
13:    }
14:
15:    //slack.
16:    //foreach(const QString& channel, slack.list())
17:    //{
18:    //    h4().Text(channel);
19:    //}
20:
21:
22:    //HubSpotAPI hubspot;
23:    //foreach(const QString& task, hubspot.task())
24:    //{
25:    //    h4().Text(task);
26:    //}
27:
28:    //foreach(const QString& category, hubspot.categories())
29:    //{
30:    //    h5().Text(category);
31:    //}
32:
33:    //WrikeAPI wrike;
34:    //foreach(const QString& folder, wrike.folders())
35:    //{
36:    //    h4().Text(folder);
37:    //}
38:
39:    //foreach(const QString& folderid, wrike.folderIds())
40:    //{
41:    //    h5().Text(folderid);
42:    //    foreach(const QString& task, wrike.task(folderid))
43:    //    {
44:    //        h6().Text(task);
45:    //    }
46:
47:    //}
48:
49:    //AsanaAPI asana;
50:    //foreach(const QString& project, asana.projects())
51:    //{
52:    //    h4().Text(project);
53:
```



```

54:     //
55:     //}
56:
57:     //foreach(const QString& projectid, asana.projectIds())
58:     //{
59:     //    h5().Text(projectid);
60:     //    foreach(const QString& task, asana.task(projectid))
61:     //    {
62:     //        h6().Text(task);
63:     //    }
64:
65:     //}
66:     //SlackAPI slack;
67:
68:     //foreach(const QString& channel, slack.list())
69:     //{
70:     //    h4().Text(channel);
71:     //}
72:
73:     //TrelloAPI trello;
74:
75:     //foreach(const QString& board, trello.boards())
76:     //{
77:     //    h4().Text(board);
78:     //}
79:
80:     //foreach(const QString& board, trello.boardIds())
81:     //{
82:     //    h4().Text(board);
83:     //    foreach(const QString& list, trello.lists(board))
84:     //    {
85:     //        h5().Text(list);
86:     //    }
87:     //}
88:     //    foreach(const QString& list, trello.listIds())
89:     //    {
90:     //        h5().Text(list);
91:     //        foreach(const QString& card, trello.cards(list))
92:     //        {
93:     //            h6().Text(card);
94:     //        }
95:     //    }
96:
97:     //}
98:     //foreach(const QString& channelid, slack.channelIds())
99:     //{
100:    //    h4().Text(channelid);
101:
102:    //    foreach(const QString& message, slack.history(channelid))
103:    //    {
104:    //        h5().Text(message);
105:    //    }
106:    //}
107:
108:    return false;
109: }

```

C++

```

bool Event(const QString& type, const QString& title, const QString& message, const
QMap<QString, QString>& valuemap, const QString& jsonevent);

```

Class

CoreChannelController Class (see page 3)

Context

Controller

Method

Event

Class

CoreChannelController (see page 3)

1.1.3.1.2 CoreChannelController::Trigger Method**Author**

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

28.01.2018

Body Source

```

1: void CoreChannelController::Trigger()
2: {
3:
4:     DropBoxAPI dropbox;
5:
6:     dropbox.setLocalPath(application()->serverPath());
7:     dropbox.syncFolder("/App_Data", "/Table", QStringList()<<".txt");
8:     dropbox.syncFolder("/Content/Img", "/Img");
9:     dropbox.syncFolder("/App_Data",
"/Doc", QStringList()<<".pdf" << ".html" << ".csv" << ".xlsx" << ".dox" << ".pptx");
10:
11:
12:     //QString type;
13:     //QString title;
14:     QString message;
15:     //QMap<QString, QString> valuemap;
16:     //QString jsonevent;
17:
18:     //WrikeAPI wrike;
19:     //foreach(const QString& folder, wrike.folders())
20:     //{
21:     //    message += folder+"-";
22:     //}
23:
24:     //foreach(const QString& folderid, wrike.folderIds())
25:     //{
26:     //    foreach(const QString& task, wrike.task(folderid))
27:     //    {
28:     //        message += task+"-";
29:     //
30:     //        application()->insertEventCard("Wrike", wrike.folderName(folderid), task);
31:     //    }
32:     //}
33:
34:     //AsanaAPI asana;
35:     //foreach(const QString& project, asana.projects())
36:     //{
37:     //    message += project+"-";
38:     //}
39:
40:     //foreach(const QString& projectid, asana.projectIds())
41:     //{
42:     //    foreach(const QString& task, asana.task(projectid))
43:     //    {
44:     //        message += task+"-";
45:     //
46:     //        application()->insertEventCard("Asana", asana.projectName(projectid), task);
47:     //    }
48:     //}
49:     //}
50:

```

```

51:     SlackAPI slack;
52:
53:     foreach(const QString& channel, slack.list())
54:     {
55:         message += channel+ "-";
56:     }
57:     foreach(const QString& channelid, slack.channelIds())
58:     {
59:         foreach(const QString& msg, slack.history(channelid))
60:         {
61:             message += msg + "-";
62:
63:             application()->insertEventCard("Slack", slack.channelName(channelid), msg);
64:         }
65:     }
66:
67:     TrelloAPI trello;
68:
69:     foreach(const QString& board, trello.boards())
70:     {
71:         message += board+ "-";
72:     }
73:
74:     foreach(const QString& boardid, trello.boardIds())
75:     {
76:         foreach(const QString& list, trello.lists(boardid))
77:         {
78:             message += list+ "-";
79:         }
80:
81:         foreach(const QString& listid, trello.listIds())
82:         {
83:             foreach(const QString& card, trello.cards(listid))
84:             {
85:                 message += card+ "-";
86:
87:                 application()->insertEventCard(trello.boardName(boardid),
trello.listName(listid), card);
88:             }
89:         }
90:     }
91:     //
92:     //CoreMail mail;
93:     //mail.open("sslout.df.eu", "system@notioncube-software.com", "system",
"!g3vfsdgvADDF");
94:     //mail.send("mail@notioncube-software.com", "TP", "Trigger", message);
95:
96:     application()->swapEventCards();
97:
98: }

```

C++

```
void Trigger();
```

Class

CoreChannelController Class ([see page 3](#))

Context

Controller

Method

Trigger

Class

CoreChannelController ([see page 3](#))

1.1.4 CoreService Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

11.01.2018

C++

```
class CoreService : public QObject;
```

Inheritance



Group

Classes (see page 1)

Legend

	Method
	protected

Members

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Core

Method

CoreService

Class

CoreService

1.1.4.1 CoreService Methods

1.1.4.1.1 Array Method

1.1.4.1.1.1 CoreService::Array Method (const QString&)

Body Source

```
1: QString Array(const QString& value)
2: {
3:     return "[" + Value(value) + "];"
4: }
```

C++

```
QString Array(const QString& value);
```

Description

This is Array, a member of class CoreService.

Class

CoreService Class ([see page 8](#))

1.1.4.1.1.2 CoreService::Array Method (const QStringList&)

Body Source

```
1: QString Array(const QStringList& valuelist)
2: {
3:     QString values;
4:     foreach(const QString& value, valuelist)
5:     {
6:         if (values.isEmpty()) values = Value(value);
7:         else values += ","+Value(value);
8:     }
9:     return "[" + values + "];"
10: }
```

C++

```
QString Array(const QStringList& valuelist);
```

Description

This is Array, a member of class CoreService.

Class

CoreService Class ([see page 8](#))

1.1.4.1.2 CoreService::Param Method

Body Source

```
1: QString Param(const QString& value)
2: {
3:     return "+" + value;
4: }
```

C++

```
QString Param(const QString& value);
```

Description

This is Param, a member of class CoreService.

Class

CoreService Class ([see page 8](#))

1.1.4.1.3 CoreService::Value Method

Body Source

```
1: QString Value(QString value)
2: {
3:     return "\"" + value.replace("\\", "\\") + "\"";
4: }
```

C++

```
QString Value(QString value);
```

Description

This is Value, a member of class CoreService.

Class

CoreService Class ([see page 8](#))

1.1.4.1.4 CoreService::convertToJson Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

11.01.2018

Body Source

```
1: QString CoreService::convertToJson(const QMap<QString, QString>&values)
2: {
3:     QString jsonparam;
4:
5:     if (!values.isEmpty())
6:     {
7:         jsonparam = "{ ";
8:
9:         bool first = true;
10:        for(QString key:values.keys())
11:        {
12:            if (!first) jsonparam += ",";
13:
14:            if (values[key].startsWith("[")
```

```

15:         {
16:             jsonparam += " \"" + key + "\": " + values[key] + ";
17:         }
18:         else
19:         {
20:
21:             if (values[key] == "<><><>")
22:                 jsonparam += " \"" + key + "\": \"\";
23:             else if (values[key]=="true"|| values[key] == "false")
24:                 jsonparam += " \"" + key + "\": " + values[key] + ";
25:             else
26:                 jsonparam += " \"" + key + "\": \"\" +
QString(values[key]).replace("\\"", "'") + "\"";
27:         }
28:         first = false;
29:     }
30:     jsonparam += " }";
31: }
32:
33: return jsonparam;
34: }

```

C++

```
QString convertToJson(const QMap<QString, QString>& values);
```

Class

CoreService Class (see page 8)

Context

Core

Method

convertToJson

Class

CoreService (see page 8)

1.1.4.1.5 CoreService::convertToParam Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

11.01.2018

Body Source

```

1: QString CoreService::convertToParam(const QMap<QString, QString>&params)
2: {
3:     QString urlparam;
4:     if (!params.isEmpty())
5:     {
6:         for(QString key:params.keys())
7:         {
8:             if (urlparam.isEmpty())
9:             {
10:                 if (params[key] == "<><><>")
11:                     urlparam = "?" + key + "=";
12:                 else
13:                     urlparam = "?" + key + "=" + QUrl::toPercentEncoding(params[key]);
14:             }
15:             else
16:             {

```

```
17:         if (params[key] == "<><><>")
18:             urlparam = urlparam + "&" + key + "=";
19:         else
20:             urlparam = urlparam + "&" + key + "=" +
QUrl::toPercentEncoding(params[key]);
21:     }
22: }
23: }
24: return urlparam;
25: }
```

C++

```
QString convertToParam(const QMap<QString, QString>& params);
```

Class

CoreService Class (see page 8)

Context

Core

Method

convertToParam

Class

CoreService (see page 8)

1.1.4.1.6 CoreService::firstUpper Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

11.01.2018

Body Source

```
1: QString CoreService::firstUpper(const QString&name)
2: {
3:     if (name.isEmpty())
4:         return QObject::emptyString();
5:     if (name.length() == 1)
6:         return name.toUpper();
7:
8:     QString result;
9:     bool uppercase = true;
10:    for (qint32 i = 0; i<name.length(); i++)
11:    {
12:        if (name[i].isLetterOrNumber())
13:        {
14:            if (uppercase) result += name.mid(i, 1).toUpper();
15:            else result += name.mid(i, 1).toLower();
16:            uppercase = false;
17:        }
18:        else
19:        {
20:            uppercase = true;
21:        }
22:    }
23:
24:    return result;
25:
26: }
```


C++

```
QString firstUpper(const QString& name);
```

Class

CoreService Class (see page 8)

Context

Core

Method

firstUpper

Class

CoreService (see page 8)

1.1.4.1.7 CoreService::get Method

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

9.2.2017

Body Source

```
1: QString CoreService::get(const QString& path)
2: {
3:     print(""); print("");
4:     print("Get:" + path);
5:
6:     QString params;
7:     if (!m_APIKey.isEmpty())
8:     {
9:         if (path.contains("?")) params = "&" + m_APIKeyName+"="+m_APIKey; else params =
"?" + m_APIKeyName + "=" + m_APIKey;
10:
11:         if (!m_APIToken.isEmpty())
12:         {
13:             params += "&" + m_APITokenName + "=" + m_APIToken;
14:         }
15:     }
16:
17:     if (!m_APIParams.isEmpty())
18:     {
19:         if (path.contains("?") || params.contains("?")) params += "&" +
m_APIParams.join("&"); else params += "?" + m_APIParams.join("&");
20:     }
21:     print("Get:" + QString(m_Https ? "https://" : "http://") + m_Server + m_APIPath +
path + params);
22:
23:     QNetworkRequest request(QUrl((m_Https ? "https://" : "http://") + m_Server +
m_APIPath + path + params));
24:     //request.setRawHeader(QString("Host").toLocal8Bit(), m_Server.toLocal8Bit());
25:     if (!m_Agent.isEmpty()) request.setRawHeader(QString("User-Agent").toLocal8Bit(),
m_Agent.toLocal8Bit());
26:     if (m_JSON) request.setRawHeader(QString("Content-Type").toLocal8Bit(),
QString("application/json").toLocal8Bit());
27:     if (m_JSON) request.setRawHeader(QString("Accept").toLocal8Bit(),
QString("application/json").toLocal8Bit());
28:     if (!m_AuthorizationBase64.isEmpty())
request.setRawHeader(QString("Authorization").toLocal8Bit(), QString("Basic " +
m_AuthorizationBase64).toLocal8Bit());
```

```

29:     if (!m_Bearer.isEmpty())
request.setRawHeader(QString("Authorization").toLocal8Bit(), QString("Bearer " +
m_Bearer).toLocal8Bit());
30:     if (!m_RawName.isEmpty()) request.setRawHeader(QString(m_RawName).toLocal8Bit(),
QString(m_RawHeader).toLocal8Bit());
31:
32:     QNetworkReply *reply = m_NetworkManager->get(request);
33:     reply->ignoreSslErrors();
34:     QEventLoop loop;
35:     connect(reply, SIGNAL(finished()), &loop, SLOT(quit()));
36:     loop.exec();
37:     print("Finished");
38:
39:     m_RawData = QByteArray();
40:     if (reply&&reply->error() == QNetworkReply::NoError)
41:     {
42:         m_RawData = reply->readAll();
43:         print("Reply:");
44:         print(m_RawData);
45:         if (reply) delete reply;
46:
47:         return QString(m_RawData);
48:     }
49:
50:     m_RawData = reply->readAll();
51:     print("Reply:");
52:     print(m_RawData);
53:
54:     print("Error:");
55:     print(reply->errorString());
56:
57:
58:     if (reply) delete reply;
59:     return oObject::emptyString();
60: }

```

C++

```
QString get(const QString& path);
```

Class

CoreService Class (see page 8)

1.1.4.1.8 CoreService::post Method**Author**

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

9.2.2017

Body Source

```

1: QString CoreService::post(const QString& path, QString &senddata)
2: {
3:     const QByteArray postdata = senddata.toLatin1();
4:     print(""); print("");
5:     print("Post:" + path);
6:     print("Data:");
7:     print(postdata);
8:
9:     QString params;
10:    if (!m_APIKey.isEmpty())
11:    {
12:        if (path.contains("?")) params = "&" + m_APIKeyName + "=" + m_APIKey; else

```

```

params = "?" + m_APIKeyName + "=" + m_APIKey;
13:
14:     if (!m_APIToken.isEmpty())
15:     {
16:         params += "&" + m_APITokenName + "=" + m_APIToken;
17:     }
18: }
19: if (!m_APIParams.isEmpty())
20: {
21:     if (path.contains("?") || params.contains("?")) params += "&" +
m_APIParams.join("&"); else params += "?" + m_APIParams.join("&");
22: }
23:
24: QNetworkRequest request(QUrl((m_Https?"https://":"http://") + m_Server + m_APIPath
+ path+params));
25: //request.setRawHeader(QString("Host").toLocal8Bit(), m_Server.toLocal8Bit());
26: if (!m_Agent.isEmpty()) request.setRawHeader(QString("User-Agent").toLocal8Bit(),
m_Agent.toLocal8Bit());
27: if (m_JSON) request.setRawHeader(QString("Content-Type").toLocal8Bit(),
QString("application/json").toLocal8Bit());
28: if (m_JSON) request.setRawHeader(QString("Accept").toLocal8Bit(),
QString("application/json").toLocal8Bit());
29: if (!m_AuthorizationBase64.isEmpty())
request.setRawHeader(QString("Authorization").toLocal8Bit(), QString("Basic " +
m_AuthorizationBase64).toLocal8Bit());
30: if (!m_Bearer.isEmpty())
request.setRawHeader(QString("Authorization").toLocal8Bit(), QString("Bearer " +
m_Bearer).toLocal8Bit());
31: if (!m_RawName.isEmpty()) request.setRawHeader(m_RawName.toLocal8Bit(),
m_RawHeader.toLocal8Bit());
32:
33: request.setRawHeader(QString("Content-Length").toLocal8Bit(),
QString("%1").arg(postdata.length()).toLocal8Bit());
34:
35: QNetworkReply *reply = m_NetworkManager->post(request, postdata);
36: reply->ignoreSslErrors();
37: QEventLoop loop;
38: connect(reply, SIGNAL(finished()), &loop, SLOT(quit()));
39: loop.exec();
40: print("Finished");
41:
42: m_RawData = QByteArray();
43: if (reply&&reply->error() == QNetworkReply::NoError)
44: {
45:     m_RawData = reply->readAll();
46:     if (!m_Raw) print("Reply:");
47:     if (!m_Raw) print(m_RawData);
48:     if (reply) delete reply;
49:     return QString(m_RawData);
50: }
51:
52: m_RawData = reply->readAll();
53: print("Reply:");
54: print(m_RawData);
55:
56:
57: print("Error:");
58: print(reply->errorString());
59:
60: if (reply) delete reply;
61: return oObject::emptyString();
62: }

```

C++

```
QString post(const QString& path, QString & senddata);
```

Class

CoreService Class (see page 8)

1.1.4.1.9 CoreService::rawData Method

Body Source

```

1: const QByteArray& rawData()
2: {
3:     return m_RawData;
4: }

```

C++

```
const QByteArray& rawData();
```

Description

This is rawData, a member of class CoreService.

Class

CoreService Class (see page 8)

1.1.5 DropBoxAPI Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

19.01.2018

C++

```
class DropBoxAPI : public CoreService;
```

Inheritance



Group

Classes (see page 1)

Legend

	Method
	protected

Members

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	

	rawData (see page 16)	
---	---	--

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Service

Method

DropBoxAPI

Class

DropBoxAPI

1.1.6 FacebookAPI Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

11.01.2018

C++

```
class FacebookAPI : public CoreService;
```



Inheritance



Group

Classes ([see page 1](#))

Legend

	Method
	protected

Members

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	

⇒🔗	Value (🔗 see page 10)	
⇒🔗	convertToJson (🔗 see page 10)	
⇒🔗	convertToParam (🔗 see page 11)	
⇒🔗	firstUpper (🔗 see page 12)	
⇒🔗	get (🔗 see page 13)	
⇒🔗	post (🔗 see page 14)	
⇒🔗	rawData (🔗 see page 16)	

CoreService Methods

	Name	Description
⇒🔗	Array(const QString&) (🔗 see page 9)	
⇒🔗	Param (🔗 see page 10)	
⇒🔗	Value (🔗 see page 10)	
⇒🔗	convertToJson (🔗 see page 10)	
⇒🔗	convertToParam (🔗 see page 11)	
⇒🔗	firstUpper (🔗 see page 12)	
⇒🔗	get (🔗 see page 13)	
⇒🔗	post (🔗 see page 14)	
⇒🔗	rawData (🔗 see page 16)	

Context

Service

Method

FacebookAPI

Class

FacebookAPI

1.1.7 GoogleCalendarAPI Class

Author

Tobias Post

Company

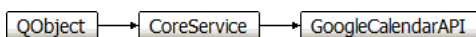
NOTIONCUBE Software GmbH

Date

11.01.2018



C++

```
class GoogleCalendarAPI : public CoreService;
```

Inheritance**Group**

Classes (🔗 see page 1)




Legend

	Method
	protected

Members**CoreService Methods**

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Service

Method

GoogleCalendarAPI

Class

GoogleCalendarAPI

1.1.8 GooglePlacesAPI Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

11.01.2018

C++

```
class GooglePlacesAPI : public CoreService;
```

Inheritance



Group

Classes (see page 1)

Legend

	Method
	protected

Members

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Service

Method

GooglePlacesAPI

Class

GooglePlacesAPI

1.1.9 GoogleSheetsAPI Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

11.01.2018

C++

```
class GoogleSheetsAPI : public CoreService;
```

Inheritance



Group

Classes (see page 1)

Legend

	Method
	protected

Members

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Service

Method

GoogleSheetsAPI

Class

GoogleSheetsAPI

1.1.10 GoogleTranslateAPI Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

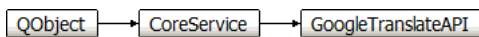
Date

11.01.2018

C++

```
class GoogleTranslateAPI : public CoreService;
```

Inheritance



Group

Classes (see page 1)

Legend

	Method
	protected

Members

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	

☞	convertToJson (🔗 see page 10)	
☞	convertToParam (🔗 see page 11)	
☞	firstUpper (🔗 see page 12)	
☞	get (🔗 see page 13)	
☞	post (🔗 see page 14)	
☞🔒	rawData (🔗 see page 16)	

Context

Service

Method

GoogleTranslateAPI

Class

GoogleTranslateAPI

1.1.11 HubSpotAPI Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

19.01.2018

C++

```
class HubSpotAPI : public CoreService;
```

Inheritance



Group

Classes (🔗 see page 1)

Legend

☞	Method
☞🔒	protected


Members

CoreService Methods

	Name	Description
☞🔒	Array(const QString&) (🔗 see page 9)	
☞🔒	Param (🔗 see page 10)	
☞🔒	Value (🔗 see page 10)	
☞	convertToJson (🔗 see page 10)	
☞	convertToParam (🔗 see page 11)	
☞	firstUpper (🔗 see page 12)	
☞	get (🔗 see page 13)	
☞	post (🔗 see page 14)	

	rawData (see page 16)
---	---

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Service

Method

HubSpotAPI

Class

HubSpotAPI

1.1.12 RSSFeed Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

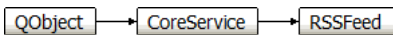
Date

11.01.2018

C++

```
class RSSFeed : public CoreService;
```



Inheritance



Group



Classes ([see page 1](#))

Legend

	Method
	protected

Members

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	

Value (see page 10)	
convertToJson (see page 10)	
convertToParam (see page 11)	
firstUpper (see page 12)	
get (see page 13)	
post (see page 14)	
rawData (see page 16)	

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Service

Method

RSSFeed

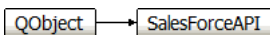
Class

RSSFeed

1.1.13 SalesforceAPI Class

C++

```
class SalesforceAPI : public QObject;
```

Inheritance**Description**

This is class SalesforceAPI.

Group

Classes (see page 1)

1.1.14 SlackAPI Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

11.01.2018

C++

```
class SlackAPI : public CoreService;
```

Inheritance



Group

Classes (see page 1)

Legend

	Method
	protected

Members

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Service

Method

SlackAPI

Class

SlackAPI

1.1.15 TrelloAPI Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

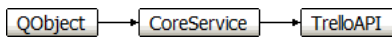
Date

11.01.2018

C++

```
class TrelloAPI : public CoreService;
```

Inheritance



Group

Classes (see page 1)

Legend

	Method
	protected

Members

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Service

Method

TrelloAPI

Class

TrelloAPI

1.1.16 TwitterAPI Class

Author

Tobias Post

Company

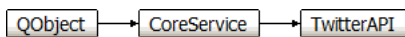
NOTIONCUBE Software GmbH

Date

11.01.2018

C++

```
class TwitterAPI : public CoreService;
```

Inheritance**Group**Classes ([see page 1](#))**Legend**

	Method
	protected

Members**CoreService Methods**

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	

☞	convertToJson (🔗 see page 10)	
☞	convertToParam (🔗 see page 11)	
☞	firstUpper (🔗 see page 12)	
☞	get (🔗 see page 13)	
☞	post (🔗 see page 14)	
☞🔒	rawData (🔗 see page 16)	

Context

Service

Method

TwitterAPI

Class

TwitterAPI

1.1.17 WrikeAPI Class

Author

Tobias Post

Company

NOTIONCUBE Software GmbH

Date

19.01.2018

C++

```
class WrikeAPI : public CoreService;
```

Inheritance



Group

Classes (🔗 see page 1)

Legend

☞	Method
☞🔒	protected

Members

CoreService Methods

	Name	Description
☞🔒	Array(const QString&) (🔗 see page 9)	
☞🔒	Param (🔗 see page 10)	
☞🔒	Value (🔗 see page 10)	
☞	convertToJson (🔗 see page 10)	
☞	convertToParam (🔗 see page 11)	
☞	firstUpper (🔗 see page 12)	
☞	get (🔗 see page 13)	
☞	post (🔗 see page 14)	

	rawData (see page 16)	
---	---	--

CoreService Methods

	Name	Description
	Array(const QString&) (see page 9)	
	Param (see page 10)	
	Value (see page 10)	
	convertToJson (see page 10)	
	convertToParam (see page 11)	
	firstUpper (see page 12)	
	get (see page 13)	
	post (see page 14)	
	rawData (see page 16)	

Context

Service

Method

WrikeAPI

Class

WrikeAPI

1.2 Variables

The following table lists variables in this documentation.

Variables

Name	Description
DropBoxAPI::m_LocalPath (see page 30)	
DropBoxAPI::m_RemotePath (see page 31)	
HubSpotAPI::m_GlobalAPIKey (see page 31)	
TrelloAPI::m_GlobalAPIKey (see page 31)	
TrelloAPI::m_GlobalToken (see page 31)	
hooksurl (see page 32)	
targeturl (see page 32)	

1.2.1 DropBoxAPI::m_LocalPath Variable

C++

```
QString DropBoxAPI::m_LocalPath;
```

Description

This is variable DropBoxAPI::m_LocalPath.

Group

Variables ([see page 30](#))

1.2.2 DropBoxAPI::m_RemotePath Variable

C++

```
QString DropBoxAPI::m_RemotePath;
```

Description

This is variable DropBoxAPI::m_RemotePath.

Group

Variables ([↗](#) see page 30)

1.2.3 HubSpotAPI::m_GlobalAPIKey Variable

C++

```
QString HubSpotAPI::m_GlobalAPIKey;
```

Description

This is variable HubSpotAPI::m_GlobalAPIKey.

Group

Variables ([↗](#) see page 30)

1.2.4 TrelloAPI::m_GlobalAPIKey Variable

C++

```
QString TrelloAPI::m_GlobalAPIKey;
```

Description

This is variable TrelloAPI::m_GlobalAPIKey.

Group

Variables ([↗](#) see page 30)

1.2.5 TrelloAPI::m_GlobalToken Variable

C++

```
QString TrelloAPI::m_GlobalToken;
```

Description

This is variable TrelloAPI::m_GlobalToken.

Group

Variables ([↗](#) see page 30)

1.2.6 hooksurl Variable

C++

```
QString hooksurl;
```

Description

This is variable hooksurl.

Group

Variables ([↗](#) see page 30)

1.2.7 targeturl Variable

C++

```
QString targeturl;
```

Description

This is variable targeturl.

Group

Variables ([↗](#) see page 30)

1.3 Files

The following table lists files in this documentation.

Files

Name	Description
AsanaAPI.cpp (↗ see page 33)	
AsanaAPI.h (↗ see page 33)	
CoreChannel.cpp (↗ see page 33)	
CoreChannel.h (↗ see page 34)	
CoreChannelController.cpp (↗ see page 34)	
CoreChannelController.h (↗ see page 34)	
CoreService.cpp (↗ see page 34)	
CoreService.h (↗ see page 35)	
DropBoxAPI.cpp (↗ see page 35)	
DropBoxAPI.h (↗ see page 35)	
FacebookAPI.cpp (↗ see page 35)	
FacebookAPI.h (↗ see page 36)	
GoogleCalendarAPI.cpp (↗ see page 36)	
GoogleCalendarAPI.h (↗ see page 36)	
GooglePlacesAPI.cpp (↗ see page 36)	
GooglePlacesAPI.h (↗ see page 37)	
GoogleSheetsAPI.cpp (↗ see page 37)	

GoogleSheetsAPI.h (see page 37)	
GoogleTranslateAPI.cpp (see page 37)	
GoogleTranslateAPI.h (see page 38)	
HubSpotAPI.cpp (see page 38)	
HubSpotAPI.h (see page 38)	
RSSFeed.cpp (see page 38)	
RSSFeed.h (see page 39)	
SalesForceAPI.cpp (see page 39)	
SalesForceAPI.h (see page 39)	
SlackAPI.cpp (see page 39)	
SlackAPI.h (see page 40)	
TrelloAPI.cpp (see page 40)	
TrelloAPI.h (see page 40)	
TwitterAPI.cpp (see page 40)	
TwitterAPI.h (see page 41)	
WrikeAPI.cpp (see page 41)	
WrikeAPI.h (see page 41)	

1.3.1 AsanaAPI.cpp

This is file AsanaAPI.cpp.

Group

Files ([see page 32](#))

1.3.2 AsanaAPI.h

This is file AsanaAPI.h.


Classes

	Name	Description
	AsanaAPI (see page 1)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.3 CoreChannel.cpp

This is file CoreChannel.cpp.

Group

Files ([see page 32](#))

1.3.4 CoreChannel.h

This is file CoreChannel.h.


Classes

	Name	Description
	CoreChannel (see page 2)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.5 CoreChannelController.cpp

This is file CoreChannelController.cpp.

Group

Files ([see page 32](#))

Variables

Name	Description
hooksurl (see page 32)	
targeturl (see page 32)	

1.3.6 CoreChannelController.h

This is file CoreChannelController.h.


Classes

	Name	Description
	CoreChannelController (see page 3)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.7 CoreService.cpp

This is file CoreService.cpp.

Group

Files ([see page 32](#))

1.3.8 CoreService.h

This is file CoreService.h.


Classes

	Name	Description
	CoreService (see page 8)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.9 DropBoxAPI.cpp

This is file DropBoxAPI.cpp.

Group

Files ([see page 32](#))

Variables

Name	Description
DropBoxAPI::m_LocalPath (see page 30)	
DropBoxAPI::m_RemotePath (see page 31)	

1.3.10 DropBoxAPI.h

This is file DropBoxAPI.h.


Classes

	Name	Description
	DropBoxAPI (see page 16)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.11 FacebookAPI.cpp

This is file FacebookAPI.cpp.

Group

Files ([see page 32](#))

1.3.12 FacebookAPI.h

This is file FacebookAPI.h.

Classes

	Name	Description
	FacebookAPI (see page 17)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.13 GoogleCalendarAPI.cpp

This is file GoogleCalendarAPI.cpp.

Group

Files ([see page 32](#))

1.3.14 GoogleCalendarAPI.h

This is file GoogleCalendarAPI.h.


Classes

	Name	Description
	GoogleCalendarAPI (see page 18)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.15 GooglePlacesAPI.cpp

This is file GooglePlacesAPI.cpp.

Group

Files ([see page 32](#))

1.3.16 GooglePlacesAPI.h

This is file GooglePlacesAPI.h.


Classes

	Name	Description
	GooglePlacesAPI (see page 19)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.17 GoogleSheetsAPI.cpp

This is file GoogleSheetsAPI.cpp.


Group

Files ([see page 32](#))

1.3.18 GoogleSheetsAPI.h

This is file GoogleSheetsAPI.h.


Classes

	Name	Description
	GoogleSheetsAPI (see page 21)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.19 GoogleTranslateAPI.cpp

This is file GoogleTranslateAPI.cpp.

Group

Files ([see page 32](#))

1.3.20 GoogleTranslateAPI.h

This is file GoogleTranslateAPI.h.


Classes

	Name	Description
	GoogleTranslateAPI (see page 22)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.21 HubSpotAPI.cpp

This is file HubSpotAPI.cpp.

Group

Files ([see page 32](#))

Variables

Name	Description
HubSpotAPI::m_GlobalAPIKey (see page 31)	

1.3.22 HubSpotAPI.h

This is file HubSpotAPI.h.


Classes

	Name	Description
	HubSpotAPI (see page 23)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.23 RSSFeed.cpp

This is file RSSFeed.cpp.

Group

Files ([see page 32](#))

1.3.24 RSSFeed.h

This is file RSSFeed.h.


Classes

	Name	Description
	RSSFeed (see page 24)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.25 SalesforceAPI.cpp

This is file SalesforceAPI.cpp.

Group

Files ([see page 32](#))

1.3.26 SalesforceAPI.h

This is file SalesforceAPI.h.


Classes

	Name	Description
	SalesForceAPI (see page 25)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.27 SlackAPI.cpp

This is file SlackAPI.cpp.

Group

Files ([see page 32](#))

1.3.28 SlackAPI.h

This is file SlackAPI.h.


Classes

	Name	Description
	SlackAPI (see page 25)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.29 TrelloAPI.cpp

This is file TrelloAPI.cpp.

Group

Files ([see page 32](#))

Variables

Name	Description
TrelloAPI::m_GlobalAPIKey (see page 31)	
TrelloAPI::m_GlobalToken (see page 31)	

1.3.30 TrelloAPI.h

This is file TrelloAPI.h.


Classes

	Name	Description
	TrelloAPI (see page 27)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.31 TwitterAPI.cpp

This is file TwitterAPI.cpp.

Group

Files ([see page 32](#))

1.3.32 TwitterAPI.h

This is file TwitterAPI.h.


Classes

	Name	Description
	TwitterAPI (see page 28)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

1.3.33 WrikeAPI.cpp

This is file WrikeAPI.cpp.

Group

Files ([see page 32](#))

1.3.34 WrikeAPI.h

This is file WrikeAPI.h.


Classes

	Name	Description
	WrikeAPI (see page 29)	

Group

Files ([see page 32](#))

Legend

	Class
---	-------

Index

A

AsanaAPI 1
 AsanaAPI class 1
 about AsanaAPI class 1
 AsanaAPI.cpp 33
 AsanaAPI.h 33

C

Classes 1
 CoreChannel 2
 CoreChannel class 2
 about CoreChannel class 2
 CoreChannel.cpp 33
 CoreChannel.h 34
 CoreChannelController 3
 CoreChannelController class 3
 Event 4
 Trigger 6
 about CoreChannelController class 3
 CoreChannelController.cpp 34
 CoreChannelController.h 34
 CoreChannelController::Event 4
 CoreChannelController::Trigger 6
 CoreService 8
 CoreService class 8
 Array 9
 Param 10
 Value 10
 about CoreService class 8
 convertToJson 10
 convertToParam 11
 firstUpper 12
 get 13
 post 14
 rawData 16
 CoreService.cpp 34
 CoreService.h 35
 CoreService::Array 9
 CoreService::Param 10

CoreService::Value 10
 CoreService::convertToJson 10
 CoreService::convertToParam 11
 CoreService::firstUpper 12
 CoreService::get 13
 CoreService::post 14
 CoreService::rawData 16

D

DropBoxAPI 16
 DropBoxAPI class 16
 about DropBoxAPI class 16
 DropBoxAPI.cpp 35
 DropBoxAPI.h 35
 DropBoxAPI::m_LocalPath 30
 DropBoxAPI::m_LocalPath variable 30
 DropBoxAPI::m_RemotePath 31
 DropBoxAPI::m_RemotePath variable 31

F

FacebookAPI 17
 FacebookAPI class 17
 about FacebookAPI class 17
 FacebookAPI.cpp 35
 FacebookAPI.h 36
 Files 32

G

GoogleCalendarAPI 18
 GoogleCalendarAPI class 18
 about GoogleCalendarAPI class 18
 GoogleCalendarAPI.cpp 36
 GoogleCalendarAPI.h 36
 GooglePlacesAPI 19
 GooglePlacesAPI class 19
 about GooglePlacesAPI class 19
 GooglePlacesAPI.cpp 36
 GooglePlacesAPI.h 37
 GoogleSheetsAPI 21
 GoogleSheetsAPI class 21
 about GoogleSheetsAPI class 21
 GoogleSheetsAPI.cpp 37

GoogleSheetsAPI.h 37
GoogleTranslateAPI 22
GoogleTranslateAPI class 22
 about GoogleTranslateAPI class 22
GoogleTranslateAPI.cpp 37
GoogleTranslateAPI.h 38

H

HubSpotAPI 23
HubSpotAPI class 23
 about HubSpotAPI class 23
HubSpotAPI.cpp 38
HubSpotAPI.h 38
HubSpotAPI::m_GlobalAPIKey 31
HubSpotAPI::m_GlobalAPIKey variable 31
RSSFeed 24
RSSFeed class 24
 about RSSFeed class 24
RSSFeed.cpp 38
RSSFeed.h 39
SalesForceAPI 25
SalesForceAPI class 25
 about SalesForceAPI class 25
SalesForceAPI.cpp 39
SalesForceAPI.h 39
SlackAPI 25
SlackAPI class 25
 about SlackAPI class 25
SlackAPI.cpp 39
SlackAPI.h 40
TrelloAPI 27
TrelloAPI class 27
 about TrelloAPI class 27
TrelloAPI.cpp 40
TrelloAPI.h 40
TrelloAPI::m_GlobalAPIKey 31
TrelloAPI::m_GlobalAPIKey variable 31
TrelloAPI::m_GlobalToken 31
TrelloAPI::m_GlobalToken variable 31
TwitterAPI 28
TwitterAPI class 28
 about TwitterAPI class 28

TwitterAPI.cpp 40
TwitterAPI.h 41
Variables 30
WrikeAPI 29
WrikeAPI class 29
 about WrikeAPI class 29
WrikeAPI.cpp 41
WrikeAPI.h 41
hooksurl 32
hooksurl variable 32

T

targeturl 32
targeturl variable 32