

Social Network Integration

(google+ & twitter)

Beispiel-URL (mit demo-login und demo-buttons):

https://www.app-arena.com/app/iconsultants/login/modules/auth/login_demo.php

Auf dieser Seite werden die Antworten der Logins als JSON String angezeigt (bei twitter inline wird die Antwort alleine angezeigt, ansonsten wird die letzte Antwort immer unten angehängt).

g+ sendet sofort eine Antwort falls der User eingelogged ist und die App schon authorisiert hat.

Falls nicht sendet g+ nur einen auth error.

GOOGLE+

Um die google+ api zu nutzen benötigt man eine g+ App, in der zum Einen die google+ api unter „Services“ aktiviert ist, und zum Anderen muss eine „Client ID for web applications“ sowie ein „Key for browser apps“ erstellt worden sein.

Die Client ID wird im „sign in“-<button> Element eingetragen, genauso wie die Callback-Funktion, die die Authorisierung als Antwort bekommt, sowie die Scopes.

Google+ App erstellen (bezogen auf einen Google Account):

<https://code.google.com/apis/console/>

Eine g+ app kann einer g+ Fanpage zugewiesen werden.

Achtung! Wenn man die Fanpage in der App angibt, dann muss Google diese Zuordnung noch reviewen und approven!

GOOGLE+ METHODS

Hier ist nur ein Auszug aus der g+ API. Wenn noch weitere Services verwendet werden (in der App aktiviert werden), dann haben diese Services ihre eigenen APIs...

<https://developers.google.com/+api/latest/>

API Reference

This API reference is organized by resource type. Each resource type has one or more data representations and one or more methods. Your use of the Google+ APIs is subject to the [Google+ Platform Terms of Service](#).

People

For People Resource details, see the [resource representation](#) page.

Method	HTTP request	Description
URIs relative to https://www.googleapis.com/plus/v1 , unless otherwise noted		
get	GET <code>/people/<i>userId</i></code>	Get a person's profile. If your app uses scope https://www.googleapis.com/auth/plus.login , this method is guaranteed to return <code>ageRange</code> and <code>language</code> .
search	GET <code>/people</code>	Search all public profiles.
listByActivity	GET <code>/activities/<i>activityId</i>/people/<i>collection</i></code>	List all of the people in the specified collection for a particular activity.
list	GET <code>/people/<i>userId</i>/people/<i>collection</i></code>	List all of the people who this user has added to one or more circles. This list is limited to the circles that the user made visible to the requesting app. This method does not return circle information.

Activities

For Activities Resource details, see the [resource representation](#) page.

Method	HTTP request	Description
URIs relative to https://www.googleapis.com/plus/v1 , unless otherwise noted		
list	GET <code>/people/<i>userId</i>/activities/<i>collection</i></code>	List all of the activities in the specified collection for a particular user.
get	GET <code>/activities/<i>activityId</i></code>	Get an activity.
search	GET <code>/activities</code>	Search public activities.

Comments

For Comments Resource details, see the [resource representation](#) page.

Method	HTTP request	Description
URIs relative to https://www.googleapis.com/plus/v1 , unless otherwise noted		
list	GET <code>/activities/<i>activityId</i>/comments</code>	List all of the comments for an activity.
get	GET <code>/comments/<i>commentId</i></code>	Get a comment.

Moments

For Moments Resource details, see the [resource representation](#) page.

Method	HTTP request	Description
URIs relative to https://www.googleapis.com/plus/v1 , unless otherwise noted		
insert	POST <code>/people/<i>userId</i>/moments/<i>collection</i></code>	Record a moment representing a user's activity such as making a purchase or commenting on a blog. Writing moments involves specifying the type , which is a moment type , and posting that type of moment's required fields.
list	GET <code>/people/<i>userId</i>/moments/<i>collection</i></code>	List all of the moments that your app has written for the authenticated user.
remove	DELETE <code>/moments/<i>id</i></code>	Delete a moment that your app has written for the authenticated user.

Last updated February 26, 2013.

GOOGLE+ SCOPES

Google Services haben ihre Scopes, die man beispielsweise im "sign in" Button angeben kann bei der Authorisierung des Users. Bestätigt der User diese Scopes, dann können die den Scopes entsprechenden Google Services danach verwendet werden. Hier ein paar Beispiele:

<https://www.googleapis.com/auth/adexchange.seller> - View and manage your Ad Exchange data
<https://www.googleapis.com/auth/adexchange.seller.readonly> - View your Ad Exchange data
<https://www.googleapis.com/auth/adsense> - View and manage your AdSense data
<https://www.googleapis.com/auth/adsense.readonly> - View your AdSense data
<https://www.googleapis.com/auth/analytics> - View and manage your Google Analytics data
<https://www.googleapis.com/auth/analytics.readonly> - View your Google Analytics data
<https://www.googleapis.com/auth/bigquery> - View and manage your data in Google BigQuery
<https://www.googleapis.com/auth/blogger> - Manage your Blogger account
<https://www.googleapis.com/auth/books> - Manage your books
<https://www.googleapis.com/auth/calendar> - Manage your calendars
<https://www.googleapis.com/auth/compute.readonly> - View your Google Compute Engine resources
<https://www.googleapis.com/auth/coordinate> - View and manage your Google Maps Coordinate jobs
<https://www.googleapis.com/auth/coordinate.readonly> - View your Google Coordinate jobs
https://www.googleapis.com/auth/devstorage.read_only - View your data in Google Cloud Storage
https://www.googleapis.com/auth/devstorage.read_write - Manage your data in Google Cloud Storage
<https://www.googleapis.com/auth/dfareporting> - View and manage DoubleClick for Advertisers reports
<https://www.googleapis.com/auth/drive.apps.readonly> - View your Google Drive apps
<https://www.googleapis.com/auth/drive.readonly> - View the files and documents in your Google Drive
<https://www.googleapis.com/auth/drive.scripts> - Modify your Google Apps Script scripts' behavior
<https://www.googleapis.com/auth/freebase> - Sign in to Freebase with your account
<https://www.googleapis.com/auth/fusiontables.readonly> - View your Fusion Tables
<https://www.googleapis.com/auth/gan> - Manage your GAN data
<https://www.googleapis.com/auth/latitude.all.city> - Manage your city-level location and location history
<https://www.googleapis.com/auth/latitude.current.best> - Manage your best-available location
<https://www.googleapis.com/auth/latitude.current.city> - Manage your city-level location
<https://www.googleapis.com/auth/orkut> - Manage your Orkut activity
<https://www.googleapis.com/auth/orkut.readonly> - View your Orkut data
<https://www.googleapis.com/auth/plus.me> - Know who you are on Google
<https://www.googleapis.com/auth/prediction> - Manage your data in the Google Prediction API
<https://www.googleapis.com/auth/shoppingapi> - View your product data
<https://www.googleapis.com/auth/siteverification> - Manage the list of sites and domains you control
<https://www.googleapis.com/auth/taskqueue> - Manage your Tasks and Taskqueues
<https://www.googleapis.com/auth/taskqueue.consumer> - Consume Tasks from your Taskqueues
<https://www.googleapis.com/auth/tasks> - Manage your tasks
<https://www.googleapis.com/auth/tasks.readonly> - View your tasks
<https://www.googleapis.com/auth/urlshortener> - Manage your goo.gl short URLs
<https://www.googleapis.com/auth/userinfo.email> - View your email address
<https://www.googleapis.com/auth/userinfo.profile> - View basic information about your account
<https://www.googleapis.com/auth/youtube> - Manage your YouTube account
<https://www.googleapis.com/auth/youtube.readonly> - View your YouTube account
<https://www.googleapis.com/auth/youtube.upload> - Manage your YouTube videos

Eine andere Scopeliste findet sich hier:

<http://sag-3.blogspot.com/2013/04/list-google-oauth-scopes.html>

Google hat einen oauthplayground:

<https://developers.google.com/oauthplayground/?hl=de>

...hier kann man einige Scopes auswählen und dann die zugehörigen Services zugreifbar machen.

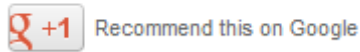
Die Scopes ermöglichen auch Zugriff auf andere Google Services, wie beispielsweise youtube, google drive oder google analytics...

GOOGLE+ BUTTONS

<https://developers.google.com/+/web/+1button/>

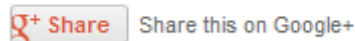
In google+ gibt es hauptsächlich 4 Buttons die für social integration genutzt werden können.

+1 like/comment/share



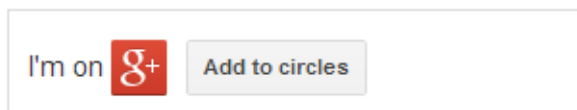
Der „+1“ Button ist ein Pendant zum FB-like Button. Wenn der User ihn betätigt dann kann er auch noch einen Kommentar zu dem „like“ hinzufügen. Die Empfehlungsmöglichkeit kann deaktiviert werden.

comment/share (no +1)

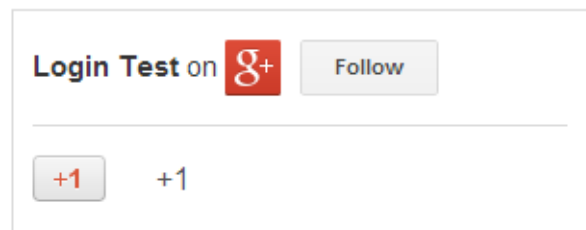


Der „Share“ Button hat nahezu dieselbe Funktionalität wie der „+1“ Button, allerdings kann der User hier einen Kommentar hinterlassen ohne den Inhalt zu „ liken“.

profile badge



page badge



Badges enthalten beispielsweise einen „+1“ Button, einen „Share“ Button, oder custom Buttons, oder auch alle zusammen in einem „Badge“. Es gibt Userprofilbezogene und Fanpagebezogene Badges.

Durch „Interactive Posts“ können Inhalte individueller angepasst werden als beim „Share“ Button. Anscheinend kann hiermit auch eine Art Multifriendselector gebastelt werden. Es können hier auch Callback-URLs verwendet werden. Interactive Posts können durch Verben, die die Verhaltensweise des Interactive Posts individualisieren, bestimmte Funktionalitäten realisieren.

(Interactive Posts sind erst einmal nicht im Beispiel enthalten aufgrund von cross-browser-/Konfigurations-Problemen)

Diese Mechanismen haben viele unterschiedliche Stylingoptionen. Von der Funktionalität her scheint es, als würde es hauptsächlich das g+-like („+1“), das Sharen und das Kommentieren an sich geben, wobei sich dies allerdings auch durch die „Interactive Posts“ relativ frei gestalten lassen könnte.

Der g+ sign in Button wird von Google standardmässig gestyled:



Angeblich kann man auch eine locale angeben um beispielsweise alle Buttons in einer gewissen Sprache rendern zu lassen, das hatte aber leider nicht funktioniert. Ansonsten sollten sich die g+ Übersetzungen auch wie bei Facebook nach der in g+ eingestellten Sprache richten.

Um g+ buttons automatisch rendern zu lassen wird vor Ende des body-tags folgendes Javascript eingefügt (mit einer onload-ready function, die nach laden der gapi aufgerufen wird):

```
<script type="text/javascript">(function() {var po = document.createElement('script'); po.type =  
'text/javascript'; po.async = true;po.src =  
'https://apis.google.com/js/client:plusone.js?onload=gapiAsyncInit';var s =  
document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(po, s);})();</script>
```


TWITTER

Twitter supported generell hauptsächlich Twitter-Funktionalitäten. Das heisst, es werden von Twitter gerenderte Buttons zur Verfügung gestellt um grundlegende Twitter Funktionen zu nutzen, Twitter stellt aber nicht direkt einen „Sign in“-Button zur Verfügung.

Da auf Twitter durch viele Open-Source Frameworks zugegriffen werden kann fiel die Entscheidung für das codebird-php Framework zum Einloggen von Usern (und möglicherweise auch später zum callen der API).

<https://github.com/mynetx/codebird-php>

Eine Twitter App erstellen (bezogen auf einen Twitter Account):

<https://dev.twitter.com/apps>

TWITTER METHODS

Die Twitter API Dokumentation ist [hier](#) zu finden:

<https://dev.twitter.com/docs>

Hauptsächlich werden bei Twitter REST API calls verwendet um auf Ressourcen zuzugreifen. Ressourcen sind bei Twitter z.B. User oder Tweets.

Hier ist eine kleine Auswahl an wichtigen Ressourcen (gekürzt)...

Timelines

Timelines are collections of Tweets, ordered with the most recent first.

Resource	Description
GET statuses/mentions_timeline	Returns the 20 most recent mentions (tweets containing a users's @screen_name) for the authenticating user. The timeline returned is the equivalent of the one seen when you view your mentions on twitter.com. This method can only return up to 800 tweets. See Working with Timelines for...
GET statuses/user_timeline	Returns a collection of the most recent Tweets posted by the user indicated by the screen_name or user_id parameters. User timelines belonging to protected users may only be requested when the authenticated user either "owns" the timeline or is an approved follower of the owner. The timeline...
GET statuses/home_timeline	Returns a collection of the most recent Tweets and retweets posted by the authenticating user and the users they follow. The home timeline is central to how most users interact with the Twitter service. Up to 800 Tweets are obtainable on the home timeline. It is more volatile for users that follow...
GET statuses/retweets_of_me	Returns the most recent tweets authored by the authenticating user that have been retweeted by others. This timeline is a subset of the user's GET statuses/user_timeline. See Working with Timelines for instructions on traversing timelines.

Tweets

Tweets are the atomic building blocks of Twitter, 140-character status updates with additional associated metadata. People tweet for a variety of reasons about a multitude of topics.

Resource	Description
GET statuses/retweets/:id	Returns up to 100 of the first retweets of a given tweet.
GET statuses/show/:id	Returns a single Tweet, specified by the id parameter. The Tweet's author will also be embedded within the tweet. See Embeddable Timelines, Embeddable Tweets, and GET statuses/oembed for tools to render Tweets according to Display Requirements.
POST statuses/destroy/:id	Destroys the status specified by the required ID parameter. The authenticating user must be the author of the specified status. Returns the destroyed status if successful.
POST statuses/update	Updates the authenticating user's current status, also known as tweeting. To upload an image to accompany the tweet, use POST statuses/update_with_media. For each update attempt, the update text is compared with the authenticating user's recent tweets. Any attempt that would result in duplication...
POST statuses/retweet/:id	Retweets a tweet. Returns the original tweet with retweet details embedded.
POST statuses/update_with_media	Updates the authenticating user's current status and attaches media for upload. In other words, it creates a Tweet with a picture attached. Unlike POST statuses/update, this method expects raw multipart data. Your POST request's Content-Type should be set to multipart/form-data with the media[]...
GET statuses/oembed	Returns information allowing the creation of an embedded representation of a Tweet on third party sites. See the oEmbed specification for information about the response format. While this endpoint allows a bit of customization for the final appearance of the embedded Tweet, be aware that the...

Search

Find relevant Tweets based on queries performed by your users.

Resource	Description
GET search/tweets	Returns a collection of relevant Tweets matching a specified query. Please note that Twitter's search service and, by extension, the Search API is not meant to be an exhaustive source of Tweets. Not all Tweets will be indexed or made available via the search interface. In API v1.1, the response...

Direct Messages

Direct Messages are short, non-public messages sent between two users. Access to Direct Messages is governed by the [The Application Permission Model](#).

Resource	Description
GET direct_messages	Returns the 20 most recent direct messages sent to the authenticating user. Includes detailed information about the sender and recipient user. You can request up to 200 direct messages per call, up to a maximum of 800 incoming DMs. Important: This method requires an access token with RWD (read,...
GET direct_messages/sent	Returns the 20 most recent direct messages sent by the authenticating user. Includes detailed information about the sender and recipient user. You can request up to 200 direct messages per call, up to a maximum of 800 outgoing DMs. Important: This method requires an access token with RWD (read,...
GET direct_messages/show	Returns a single direct message, specified by an id parameter. Like the /1.1/direct_messages.format request, this method will include the user objects of the sender and recipient. Important: This method requires an access token with RWD (read, write...
POST direct_messages/destroy	Destroys the direct message specified in the required ID parameter. The authenticating user must be the recipient of the specified direct message. Important: This method requires an access token with RWD (read, write...
POST direct_messages/new	Sends a new direct message to the specified user from the authenticating user. Requires both the user and text parameters and must be a POST. Returns the sent message in the requested format if successful.

Friends & Followers

Users follow their interests on Twitter through both one-way and mutual following relationships.

Resource	Description
GET friendships/no_retweets/ids	Returns a collection of user_ids that the currently authenticated user does not want to receive retweets from. Use POST friendships/update to set the "no retweets" status for a given user account on behalf of the current user.
GET friends/ids	Returns a cursored collection of user IDs for every user the specified user is following (otherwise known as their "friends"). At this time, results are ordered with the most recent following first — however, this ordering is subject to unannounced change and eventual consistency issues....
GET followers/ids	Returns a cursored collection of user IDs for every user following the specified user. At this time, results are ordered with the most recent following first — however, this ordering is subject to unannounced change and eventual consistency issues. Results are given in groups of 5,000 user...
GET friendships/lookup	Returns the relationships of the authenticating user to the comma-separated list of up to 100 screen_names or user_ids provided. Values for connections can be: following, following_requested, followed_by, none.
GET friendships/incoming	Returns a collection of numeric IDs for every user who has a pending request to follow the authenticating user.
GET friendships/outgoing	Returns a collection of numeric IDs for every protected user for whom the authenticating user has a pending follow request.
POST friendships/create	Allows the authenticating users to follow the user specified in the ID parameter. Returns the befriended user in the requested format when successful. Returns a string describing the failure condition when unsuccessful. If you are already friends with the user a HTTP 403 may be returned, though for...
POST friendships/destroy	Allows the authenticating user to unfollow the user specified in the ID parameter. Returns the unfollowed user in the requested format when successful. Returns a string describing the failure condition when unsuccessful. Actions taken in this method are asynchronous and changes will be eventually...
POST friendships/update	Allows one to enable or disable retweets and device notifications from the specified user.

Users

Users are at the center of everything Twitter: they follow, they favorite, and tweet & retweet.

Resource	Description
GET account/settings	Returns settings (including current trend, geo and sleep time information) for the authenticating user.
GET account/verify_credentials	Returns an HTTP 200 OK response code and a representation of the requesting user if authentication was successful; returns a 401 status code and an error message if not. Use this method to test if supplied user credentials are valid.
POST account/settings	Updates the authenticating user's settings.
POST account/update_delivery_device	Sets which device Twitter delivers updates to for the authenticating user. Sending none as the device parameter will disable SMS updates.
POST account/update_profile	Sets values that users are able to set under the "Account" tab of their settings page. Only the parameters specified will be updated.
POST account/update_profile_background_image	Updates the authenticating user's profile background image. This method can also be used to enable or disable the profile background image. Although each parameter is marked as optional, at least one of image, tile or use must be provided when making this request.
POST account/update_profile_colors	Sets one or more hex values that control the color scheme of the authenticating user's profile page on twitter.com. Each parameter's value must be a valid hexadecimal value, and may be either three or six characters (ex: #fff or #ffffff).
POST account/update_profile_image	Updates the authenticating user's profile image. Note that this method expects raw multipart data, not a URL to an image. This method asynchronously processes the uploaded file before updating the user's profile image URL. You can either update your local cache the next time you request the user's...
GET blocks/list	Returns a collection of user objects that the authenticating user is blocking. Important On October 15, 2012 this method will become censored by default, altering the default response format. See Using cursors to navigate collections for more details on how cursoring works.
GET blocks/ids	Returns an array of numeric user ids the authenticating user is blocking. Important On October 15, 2012 this method will become censored by default, altering the default response format. See Using cursors to navigate collections for more

Resource	Description
	details on how cursoring works.
POST blocks/create	Blocks the specified user from following the authenticating user. In addition the blocked user will not show in the authenticating users mentions or timeline (unless retweeted by another user). If a follow or friend relationship exists it is destroyed.
POST blocks/destroy	Un-blocks the user specified in the ID parameter for the authenticating user. Returns the un-blocked user in the requested format when successful. If relationships existed before the block was instated, they will not be restored.
GET users/lookup	Returns fully-hydrated user objects for up to 100 users per request, as specified by comma-separated values passed to the user_id and/or screen_name parameters. This method is especially useful when used in conjunction with collections of user IDs returned from GET friends/ids and GET followers/...

Places & Geo

Users tweet from all over the world. These methods allow you to attach location data to tweets and discover tweets & locations.

Resource	Description
GET geo/id/:place_id	Returns all the information about a known place.
GET geo/reverse_geocode	Given a latitude and a longitude, searches for up to 20 places that can be used as a place_id when updating a status. This request is an informative call and will deliver generalized results about geography.
GET geo/search	Search for places that can be attached to a statuses/update. Given a latitude and a longitude pair, an IP address, or a name, this request will return a list of all the valid places that can be used as the place_id when updating a status. Conceptually, a query can be made from the user's location...
GET geo/similar_places	Locates places near the given coordinates which are similar in name. Conceptually you would use this method to get a list of known places to choose from first. Then, if the desired place doesn't exist, make a request to POST geo/place to create a new one. The token contained in the response is the...
POST geo/place	Creates a new place object at the given latitude and longitude. Before creating a place you need to query GET geo/similar_places with the latitude, longitude and name of the place you wish to create.

OAuth

Twitter uses OAuth for authentication. Be sure and read about [Authentication & Authorization](#).

Resource	Description
GET oauth/authenticate	Allows a Consumer application to use an OAuth request_token to request user authorization. This method is a replacement of Section 6.2 of the OAuth 1.0 authentication flow for applications using the callback authentication flow. The method will use the currently logged in user as the account for...
GET oauth/authorize	Allows a Consumer application to use an OAuth Request Token to request user authorization. This method fulfills Section 6.2 of the OAuth 1.0 authentication flow. Desktop applications must use this method (and cannot use GET oauth/authenticate). Please use HTTPS for this method, and all other OAuth...
POST oauth/access_token	Allows a Consumer application to exchange the OAuth Request Token for an OAuth Access Token. This method fulfills Section 6.3 of the OAuth 1.0 authentication flow. The OAuth access token may also be used for xAuth operations. Please use HTTPS for this method, and all other OAuth token negotiation...
POST oauth/request_token	Allows a Consumer application to obtain an OAuth Request Token to request user authorization. This method fulfills Section 6.1 of the OAuth 1.0 authentication flow. It is strongly recommended you use HTTPS for all OAuth authorization steps. Usage Note: Only ASCII values are accepted for the...
POST oauth2/token	Allows a registered application to obtain an OAuth 2 Bearer Token, which can be used to make API requests on an application's own behalf, without a user context. This is called Application-only authentication. A Bearer Token may be invalidated using oauth2/invalidate_token. Once a Bearer Token has...
POST oauth2/invalidate_token	Allows a registered application to revoke an issued OAuth 2 Bearer Token by presenting its client credentials. Once a Bearer Token has been invalidated, new creation attempts will yield a different Bearer Token and usage of the invalidated token will no longer be allowed. As with all API v1.1...

TWITTER PERMISSIONS

Bei Twitter werden keine Scopes benutzt wie bei Facebook oder g+.

Die Twitter App erhält gewisse permissions aus ihrer Konfiguration.

Eine Twitter App kann entweder nur Leserechte bekommen, Lese- und Schreibrechte, oder Lese- und Schreibrechte und Rechte zum Zugriff auf direkte Nachrichten.

Hauptsächlich wird dann mit dem Access Token gearbeitet.

<https://dev.twitter.com/docs/application-permission-model>

Es gibt ausserdem eine explizite permission die unbedingt eingeschaltet werden sollte:

„Allow this application to be used to [Sign in with Twitter](#)“ (unter Application Type)

WEB INTENTS

Eine weitere Möglichkeit Twitter in eine Website zu integrieren sind Web Intents.

Hierbei werden ausschliesslich intent URLs verwendet um auf Twitter zuzugreifen.

Somit können die standard Aktionen an die eigene Website individuell angepasst werden.

<https://dev.twitter.com/docs/intents>

TWITTER BUTTONS

Der Twitter sign in Prozess wird einfach dadurch angestoßen, dass der User auf folgende Seite weitergeleitet wird:

twitter_auth.php

Diese kümmert sich dann mit Hilfe des codebird-php Frameworks um das aufrufen der korrekten Adresse...

Da hier erhebliche Xdomain Probleme auftreten, falls man versucht den User durch Javascript (respektive xhr-calls) über Twitter einzuloggen, wird hier einfach die Weiterleitung durchgeführt um das codebird-php Framework für den OAuth login zu verwenden. Es muss nur die Konfiguration angepasst werden (consumerkey & -secret, callback-url). Es ist auch eine „Weiterleitung“ durch ein Popup möglich.

Deshalb sind die login Buttons für Twitter frei stylebar:



Um twitter buttons für Twitter Funktionen automatisch rendern zu lassen wird vor Ende des body-tags folgendes Javascript eingefügt:

```
<script>!function(d,s,id){var
js,fjs=d.getElementsByTagName(s)[0],p=/^http:/.test(d.location)?'http':'https';if(!d.getElementById(i
d)){js=d.createElement(s);js.id=id;js.src=p+'//platform.twitter.com/widgets.js';fjs.parentNode.insert
Before(js,fjs);}(document, 'script', 'twitter-wjs'); </script>
```

Button Codeschnipsel können hier generiert werden:

<https://twitter.com/about/resources/buttons>

In Twitter gibt 4 Buttons für Userinteraktionen:

Der share Button zeigt auch die Anzahl der User an die den Button bereits zum twittern verwendet haben. D.h. wenn ein User den Button mehrmals betätigt dann wird nur um 1 hochgezählt.

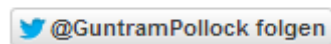
Die Userzahl wird nicht live geupdated. Wenn ein User den Button benutzt muss die Seite neu geladen werden damit der counter aktualisiert wird!

teile einen Link



Mit dem folgen Button kann man entweder einem user folgen oder er öffnet ein follower-Management Popup in dem der User seine follows verwalten kann, falls der Button inaktiv ist weil der User schon darauf geklickt hatte.

folgen



Mit dem Hashtag kann der User direkt auf ein tag tweeten.

Hashtag



Durch die Erwähnung kann der User direkt an einen anderen User tweeten.

Erwähnung



Die Buttons sind alle auf der Twitter Seite konfigurierbar. Die Seite erzeugt dann einen Codeschnipsel für den konfigurierten Button, den der developer dann in seinen Sourcecode pasten kann.

Achtung! Der <script ... Codeschnipsel wird nur einmal auf einer Seite benötigt und nicht bei jedem Button!