

# **Google plus**

Extension Key: mn\_google\_plus

Language: en

Keywords: Google plus

Copyright 2000-2012, Mattias Nilsson, <tollepjaer@gmail.com>

This document is published under the Open Content License available from http://www.opencontent.org/opl.shtml

The content of this document is related to TYPO3

- a GNU/GPL CMS/Framework available from www.typo3.org



### **Table of Contents**

Google plus	1
Introduction	
What does it do?	3
Hears manual	4

FAQ	5
Configuration	6
Reference	6
To-Do list	7



### Introduction

### What does it do?

- A extension for presentation of a frontend user's Google plus information and activities plus a public profile search.

**TYP03** 3



### Users manual

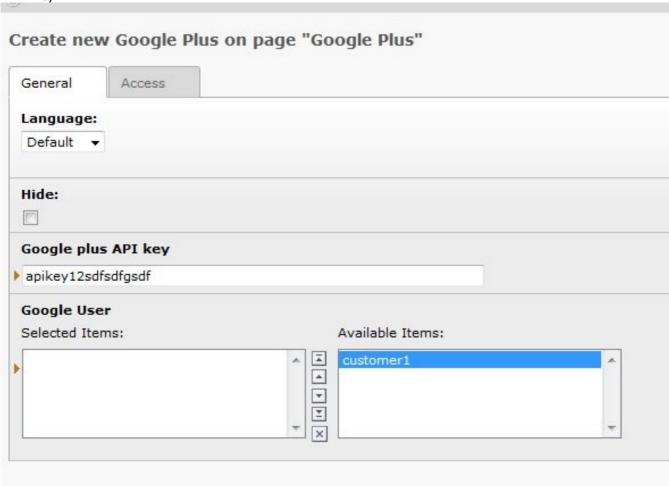
First of all an API for Google plus is required and this is acquired on Google API:s found here:

https://code.google.com/apis/ chose to activate Google+ API. After that you will get an "API key" for using in TYPO3. API Access AdSense Management API OFF Courtesy limit: 10,000 queries/day Reports Quotas OFF Malytics API Courtesy limit: 50,000 queries/day OFF Audit API Courtesy limit: 10,000 queries/day Blogger API Courtesy limit: 1,000 queries/day Request access... OFF Books API Courtesy limit: 1,000 queries/day Calendar API OFF Courtesy limit: 10,000 queries/day OFF Custom Search API Courtesy limit: 100 queries/day • Pricing OFF Freebase API Courtesy limit: 100,000 queries/day OFF Soogle Affiliate Network API Courtesy limit: 1,000 queries/day Soogle Cloud SQL Request access... OFF Google Cloud Storage Pricing OFF 5 Google Maps API v2 Courtesy limit: 25,000 queries/day • Pricing 0 OFF gle Maps API v3 Courtesy limit: 25,000 queries/day ON Soogle+ API Courtesy limit: 1,000 queries/day OFF Management Services API Courtesy limit: 100,000 queries/day OFF Identity Toolkit API Courtesy limit: 1,000,000 queries/day OFF Latitude API Courtesy limit: 1,000,000 queries/day OFF Moderator API Courtesy limit: 1,000,000 queries/day OFF Orkut REST API Courtesy limit: 1,000 queries/day OFF Page Speed Online API Courtesy limit: 250 queries/day Page Speed Service Request access... Diagon ADI OFF Courtoev limit: 1.000 quorioe/day • Pric

TYP03 🔽



The next step is to create a "Google Plus" record in TYPO3 backend. Just paste the the apikey into the field "Google plus API key".



- Then install extension on the desired page and include folders or pages containing the api record and frontend users on "Record Storage Page".
- And last but not least dont forget to include TS config for extension "Google Plus".
- Note that a frontend user also requires a Google plus id to work. A scheduler for collecting this user ids is under development.

#### **FAQ**

- Todo

TYP03 🔽



## Configuration

- Target group: Developers

#### Reference

- Reference (TypoScript)

Property:	Data type:	Description:	Default:
googlePlusApiUrl	string	Sets the url to the Google plus API	

#### **Default settings**

```
plugin.tx_mngoogleplus {
    settings {
          googlePlusApiUrl = https://www.googleapis.com/plus/v1/
       view {
               templateRootPath = {$plugin.tx_mngoogleplus.view.templateRootPath}
partialRootPath = {$plugin.tx_mngoogleplus.view.partialRootPath}
layoutRootPath = {$plugin.tx_mngoogleplus.view.layoutRootPath}
       persistence {
               storagePid = {$plugin.tx_mngoogleplus.persistence.storagePid}
               classes {
                       Tx_Extbase_Domain_Model_FrontendUser {
                              subclasses {
Tx_MnGooglePlus_Domain_Model_GoogleUser = Tx_MnGooglePlus_Domain_Model_GoogleUser
                       Tx_MnGooglePlus_Domain_Model_GoogleUser {
                              mapping {
                                tableName = fe_users
                                 recordType = Tx_MnGooglePlus_Domain_Model_GoogleUser
                         columns {
                              googlePlusId.mapOnProperty = google plus id
     }
```



## To-Do list

- Scheduler for collecting users id's from Google+.
- Backend module for a overview of relations between frontend users and Google +.
- TYPO3 Forge