Sitecore Forms Extensions Documentation

Release 1.4.3

Bart Verdonck

Nov 01, 2018

Welcome, happy to see your are (considering) using the Sitecore Forms Extensions module to give your Sitecore 9 forms a feature boost!

With this module you will be able to prefill forms, get a file upload field, get better integration with EXM, a Google Recaptcha, additional validators and much more.

These pages serve as the documentation for the module. They are splitted into 2 sections:

- **Content Editor Documentation** Here you will find how to use the module once installed and configured.
- **Developer Documentation** This part contains installation instructions and info on how to add customizations on the module that fit your solution.

If you are missing stuff in this documentation. Feel free to send me a private message on Sitecore Slack (@bverdonck) or send me a tweet on twitter (@_onelittlespark).

Found a bug? Please reported it on Github

Have fun with the module!

CHAPTER 1

Form Bindings (Prefilling)

CHAPTER 2

Installation and Setup

2.1 Download

The latest version can be found in the download section on Github.

There are 3 distribution packages available:

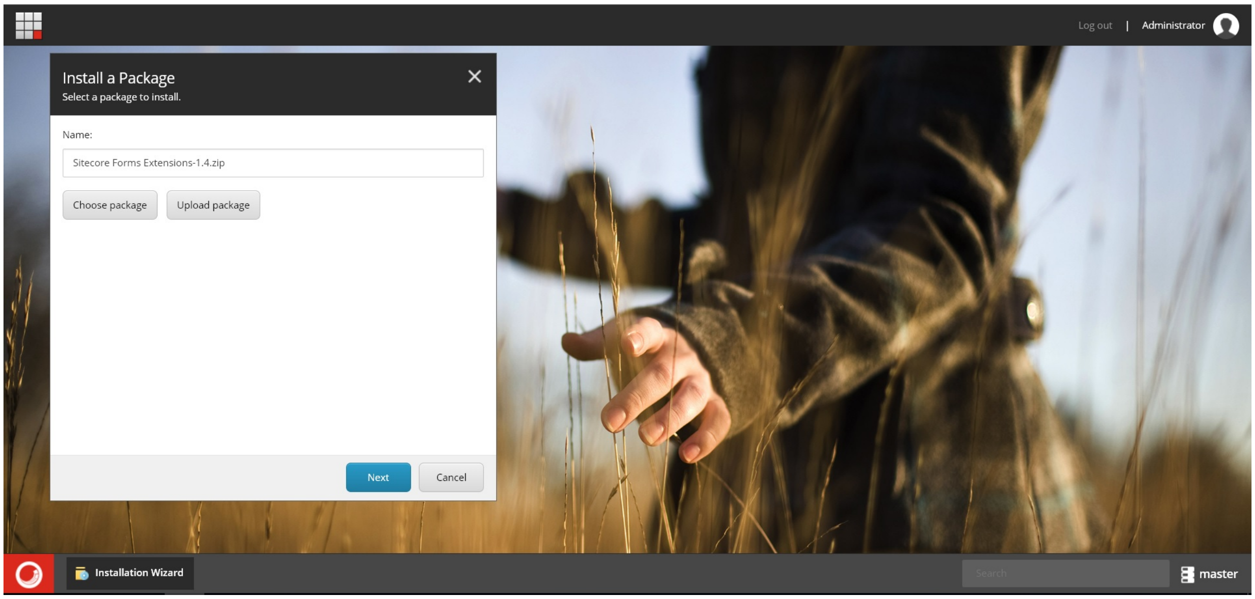
Sitecore Forms Extensions-<version>.zip Sitecore Package to install via Sitecore's installation wizard. This is the most common option.

Sitecore Forms Extensions-<version>.scwpd.zip Use this package for initial install on Azure PaaS via ARM templates.

Sitecore Forms Extensions-<version>-nodb.scwpd.zip Use this package for redeployment via ARM on Azure PaaS, this package will only install DLL's and file, sitecore items are excluded from this package.

2.2 Install

Just install the package in Sitecore via installation wizard.



Install a Package

Select a package to install.

Name: Sitecore Forms Extensions-1.4.zip

2.3 Configuration

Most functionalities of the libray work out of the box. However, the fileupload and captcha features require some additional configuration.

2.3.1 File Upload

The file upload requires to set a location to store the uploaded files. A file upload storage provider must be added in configuration.

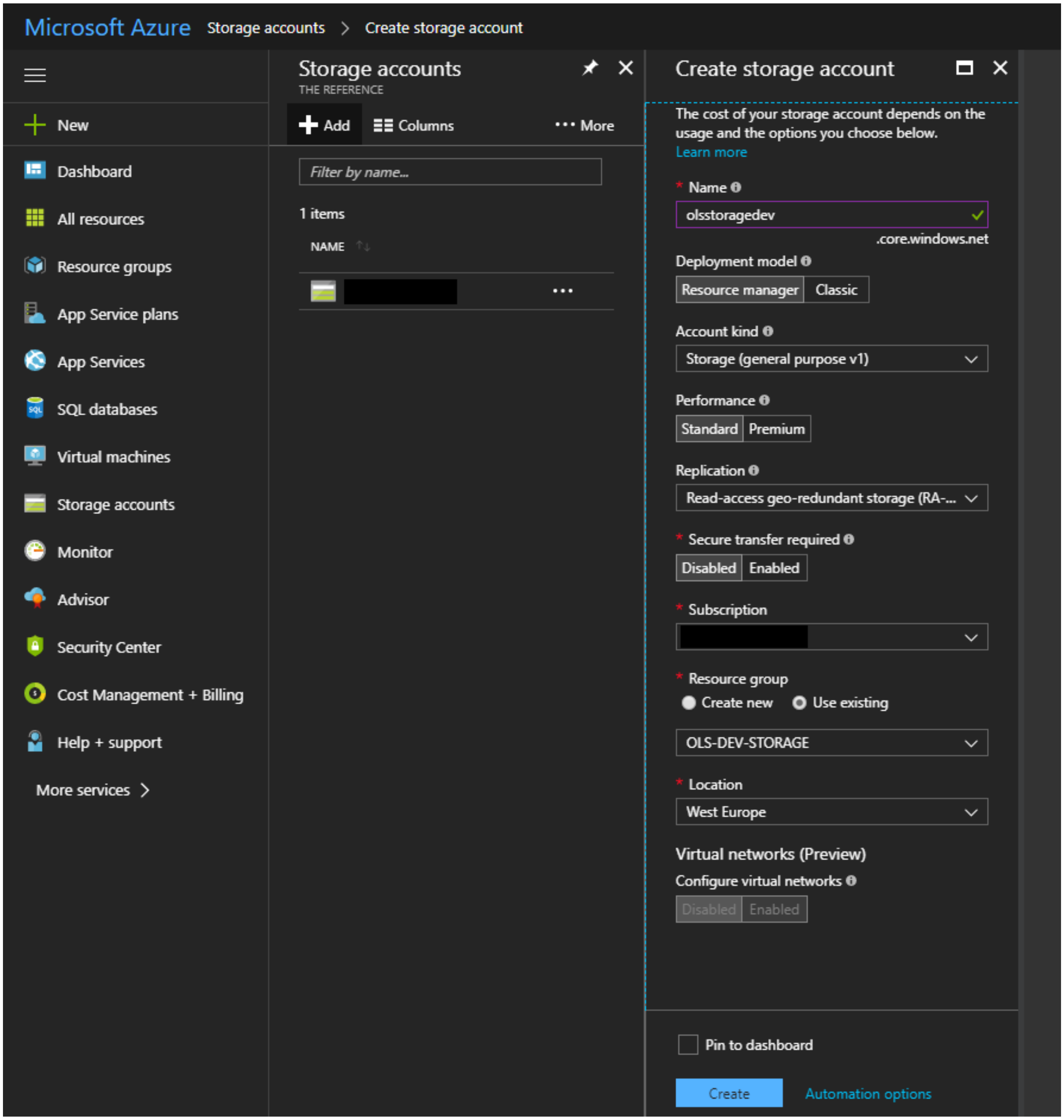
There are currently 2 store locations available:

Configure Azure Blob Storage

Setup Storage Account in Azure

First, in the case you don’t have an Azure Blob Storage in your solution already, you will need to create one.

To do so, go into azure portal. Click New and choose Storage Account.



[redacted]

Create storage account

The cost of your storage account depends on the usage and the options you choose below.
Learn more

* Name: olsstoragedev .core.windows.net

Deployment model: Resource manager | Classic

Account kind: Storage (general purpose v1)

Performance: Standard | Premium

Replication: Read-access geo-redundant storage (RA-...

* Secure transfer required: Disabled | Enabled

* Subscription: [redacted]

* Resource group: ○ Create new ● Use existing
OLS-DEV-STORAGE

* Location: West Europe

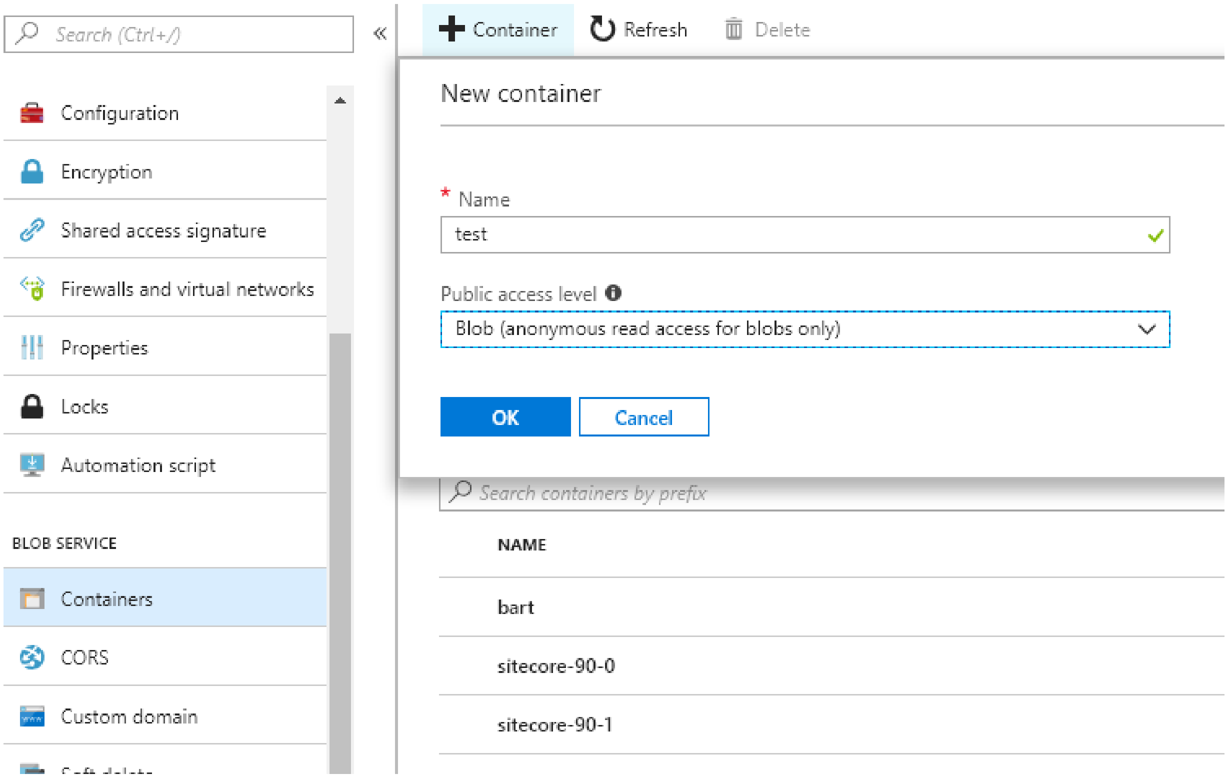
Virtual networks (Preview)
Configure virtual networks: Disabled | Enabled

□ Pin to dashboard

Next, create a storage container.

Choose public access level blob if you want easy access to the uploaded files by link. (All files will be renamed with a random GUID for security.)

Note: This is not required for the module to work



New container

* Name: test

Public access level: Blob (anonymous read access for blobs only)

NAME
bart
sitecore-90-0
sitecore-90-1

Add config file to your solution

Next, we will tell the module how to connect to the storage account.

Create a file *Feature.FormsExtentions.AzureStorageProvider.config* and add it in your website folder under App_Config/Environment

```
<?xml version="1.0"?>
<configuration xmlns:patch="http://www.sitecore.net/xmlconfig/"  xmlns:role="http://
↪www.sitecore.net/xmlconfig/role/">
  <sitecore>
    <formExtensions>
      <fileUploadStorageProvider type="Feature.FormsExtensions.Business.FileUpload.
↪AzureBlobStorageFileUploadStorageProvider, Feature.FormsExtensions">
        <connectionString></connectionString>
        <blobContainer></blobContainer>
        <folder>formextentions/{formName}/{fieldName}/{language}</folder>
      </fileUploadStorageProvider>
    </formExtensions>
  </sitecore>
</configuration>
```

Go to your storage account on Azure, browse to Access Keys, and copy either one of the connectionstrings. Put this connectionstring in the config.

Enter the name you have chosen for your container in the blobcontainer part of the configuration.

Finally, you will notice the folder attribute in the configuration. You can leave this empty or put in any desired folder structure that should be followed to store your files in.

All forms will follow the same config for upload of the files. You cannot create individual storages for individual forms. This is a design choice, so that content editors do not have to wory about where to store files.

However, the folder structure does support 3 variables that can be used:

- {formName} This handle will get replaced by the name of the form.
- {fieldName} Each form-upload field can be named in the forms-editor. This handle will be replaced by that name.
- {language} If you want to seperate the content by language, use this handle.

Note that none of these handles are required. They are all optional, including the order.

Local Blob Storage

blablabla

2.3.2 Google Captcha

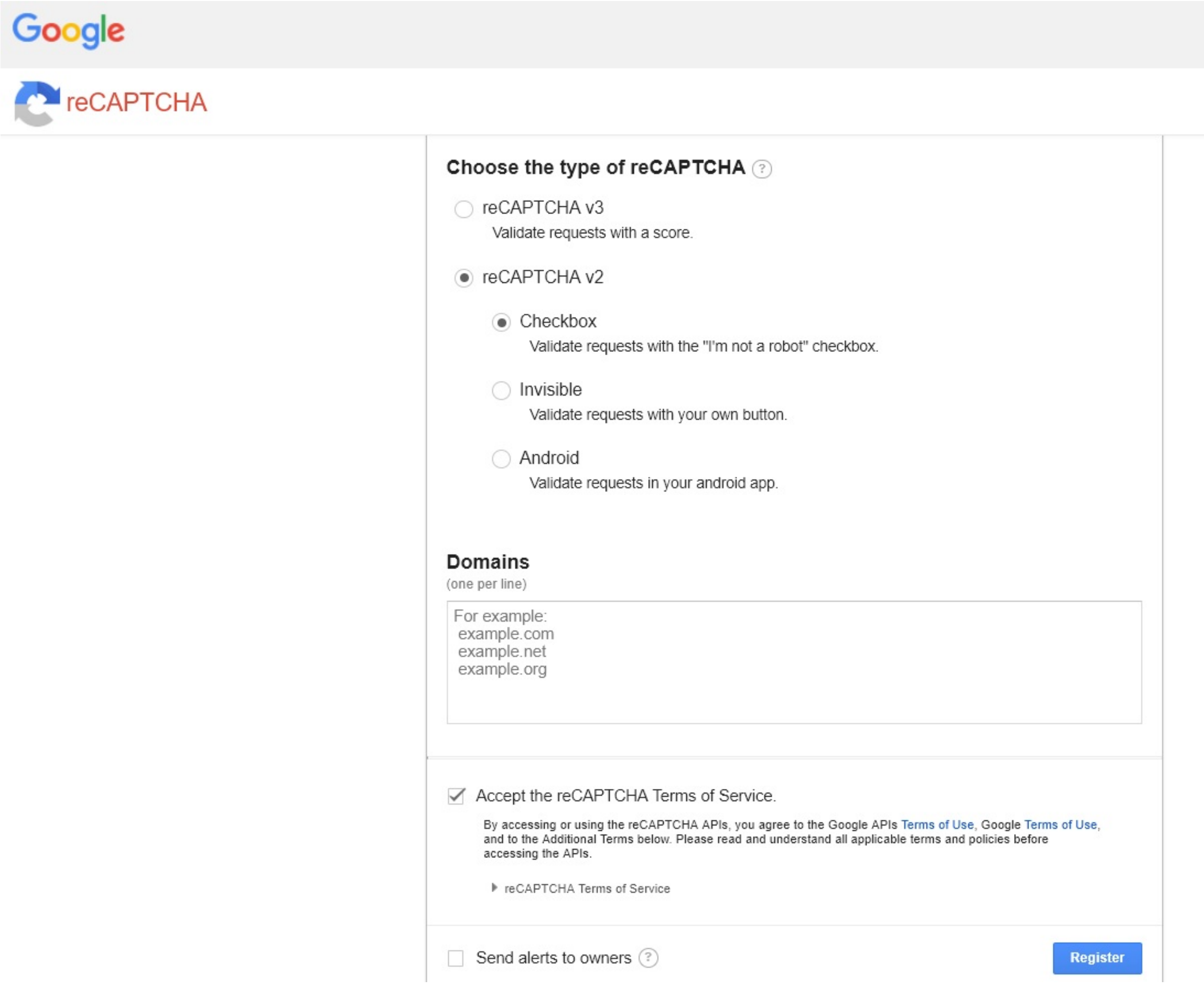
For Google Captcha to work, we need to set the public and private key in the configuration.

Configure Google Captcha v2

Setup Google Account

First we need to configure our account in Google Recaptcha Portal.

https://www.google.com/recaptcha/admin#list



Choose the type of reCAPTCHA ?

○ reCAPTCHA v3
Validate requests with a score.

● reCAPTCHA v2

● Checkbox
Validate requests with the "I'm not a robot" checkbox.

○ Invisible
Validate requests with your own button.

○ Android
Validate requests in your android app.

Domains
(one per line)

For example:
example.com
example.net
example.org

☑ Accept the reCAPTCHA Terms of Service.

By accessing or using the reCAPTCHA APIs, you agree to the Google APIs Terms of Use, Google Terms of Use, and to the Additional Terms below. Please read and understand all applicable terms and policies before accessing the APIs.

▸ reCAPTCHA Terms of Service

□ Send alerts to owners ?

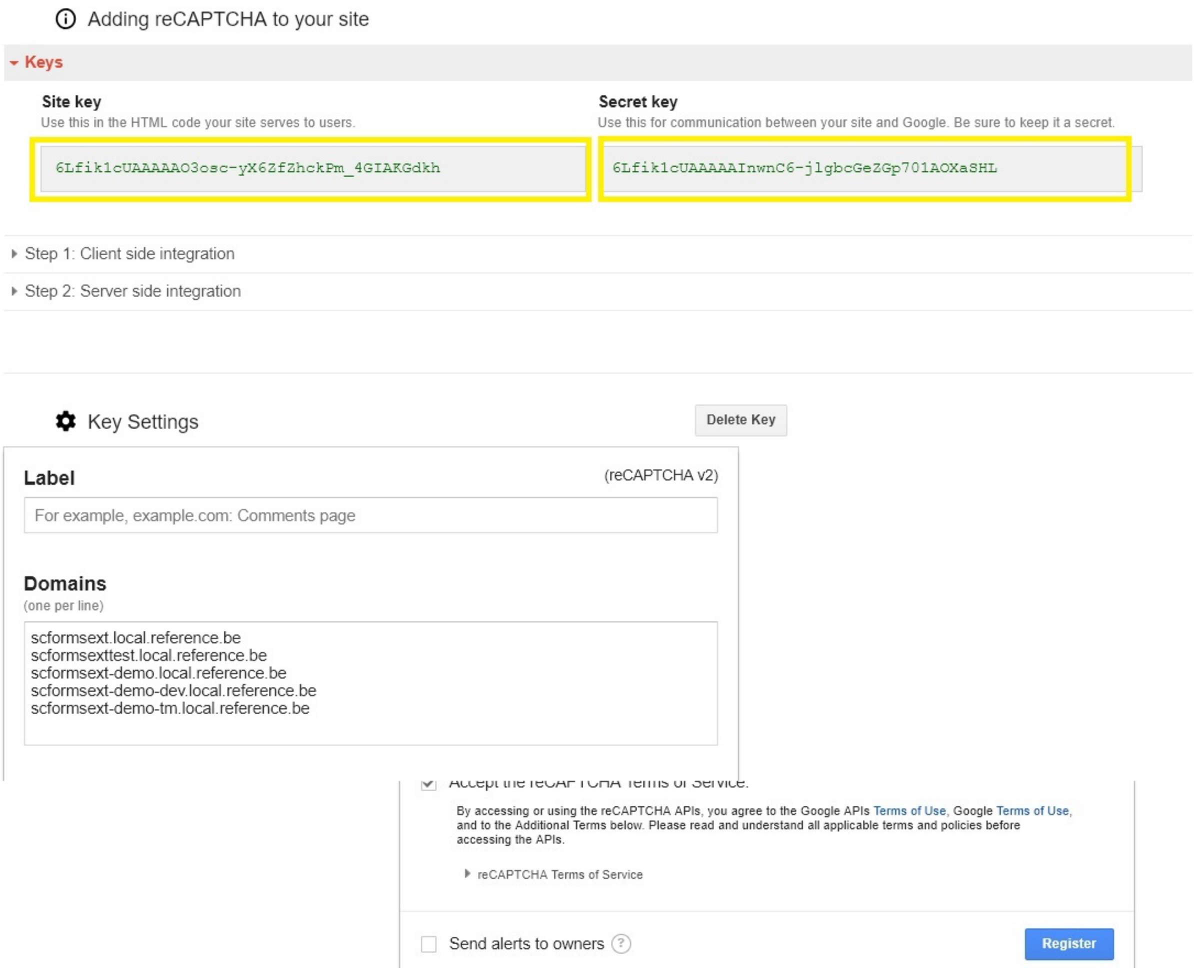
Add Config File to your Solution

Create a file *Feature.FormsExtentions.GoogleRecaptcha.config* and add it in your website folder under App_Config/Environment

```
<?xml version="1.0"?>
<configuration xmlns:patch="http://www.sitecore.net/xmlconfig/"  xmlns:role="http://
↪www.sitecore.net/xmlconfig/role/">
  <sitecore role:require="Standalone or ContentManagement or DedicatedDispatch">
    <settings>
      <setting name="GoogleCaptchaPublicKey" value="6Lfik1cUAAAAAO3osc-yX6ZfZhckPm_
↪4GIAKGdkh" />
      <setting name="GoogleCaptchaPrivateKey" value="6Lfik1cUAAAAAInwnC6-
↪jlgbcGeZGp701AOXaSHL" />
    </settings>
  </sitecore>
</configuration>
```

The setting “GoogleCaptchaPublicKey” should contain the site key.

The setting “GoogleCaptchaPrivateKey” should contain the secret key.



Adding reCAPTCHA to your site

Keys

Site key
Use this in the HTML code your site serves to users.

```
6Lfik1cUAAAAAO3osc-yX6ZfZhckPm_4GIAKGdkh
```

Secret key
Use this for communication between your site and Google. Be sure to keep it a secret.

```
6Lfik1cUAAAAAInwnC6-jlgbcGeZGp701AOXaSHL
```

Step 1: Client side integration

Step 2: Server side integration

Key Settings

Label (reCAPTCHA v2)

For example, example.com: Comments page

Domains
(one per line)

```
scformsext.local.reference.be
scformsexttest.local.reference.be
scformsext-demo.local.reference.be
scformsext-demo-dev.local.reference.be
scformsext-demo-tm.local.reference.be
```

☑ Accept the reCAPTCHA Terms of Service.

By accessing or using the reCAPTCHA APIs, you agree to the Google APIs Terms of Use, Google Terms of Use, and to the Additional Terms below. Please read and understand all applicable terms and policies before accessing the APIs.

reCAPTCHA Terms of Service

□ Send alerts to owners