

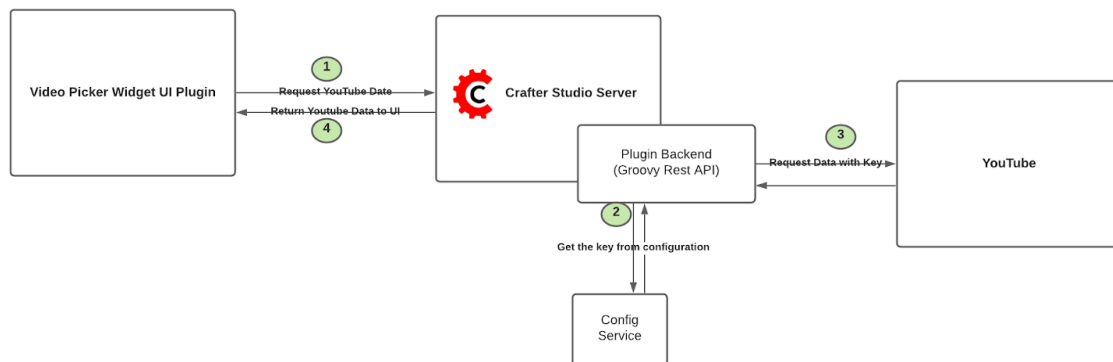
## Add a Google API Key

A Google API Key is required to search YouTube videos. For this plugin, we are using **YouTube Data API v3**. To request an API key, refer to this [document from Google](#).

While we can use *YouTube Data API* key directly from browsers and send HTTP requests to *youtube.googleapis.com*, passing the key from the front end exposes the key to the browser.

Instead, we'll keep the key on the server and encrypt it by using the [Crafter CMS Encryption Tool](#). This ensures that it's easy to manage, unique per environment and secured.

This diagram shows how we will send requests securely using the *YouTube Data API*.



In order to achieve this, we need to manually update the following:

- Create an encrypted version of YouTube Data API Key
- Update encrypted key to `site-config.xml`
- Update blacklist to allow HTTP call from Groovy Script to external
- Update `studio-config.yaml` to access to beans

While authors use the YouTube Widget search function, the plugin sends the search keyword to our Groovy script. The Groovy script gets the encrypted api key from `site-config.xml` and then decrypts at run time, sending actual requests to *youtube.googleapis.com*.

To learn more about the Groovy API, check out [this documentation](#).

## Encrypt API Key

Go to Global Menu > Encryption Tool. Input your key to the box and click “Encrypt Text”

The screenshot shows the 'Encryption Tool' interface. On the left is a sidebar menu with items: Sites, Users, Groups, Cluster, Audit, Logging Levels, Log Console, Global Config, Encryption Tool (highlighted), Token Management, About, Account, and Documentation. The main content area is titled 'Encryption Tool'. It features a 'Raw Text' input field with the text 'sample\_api\_key'. Below this is a text area displaying the encrypted output: `${enc:CCE-V1#sBHDxkSqWi7cFI+DT1Wb/UviDWI9JIGfV/6SJEsD78=}`. At the bottom of the main area are two buttons: 'Clear' and 'Encrypt Text'.

Copy the text output to use in the next step.

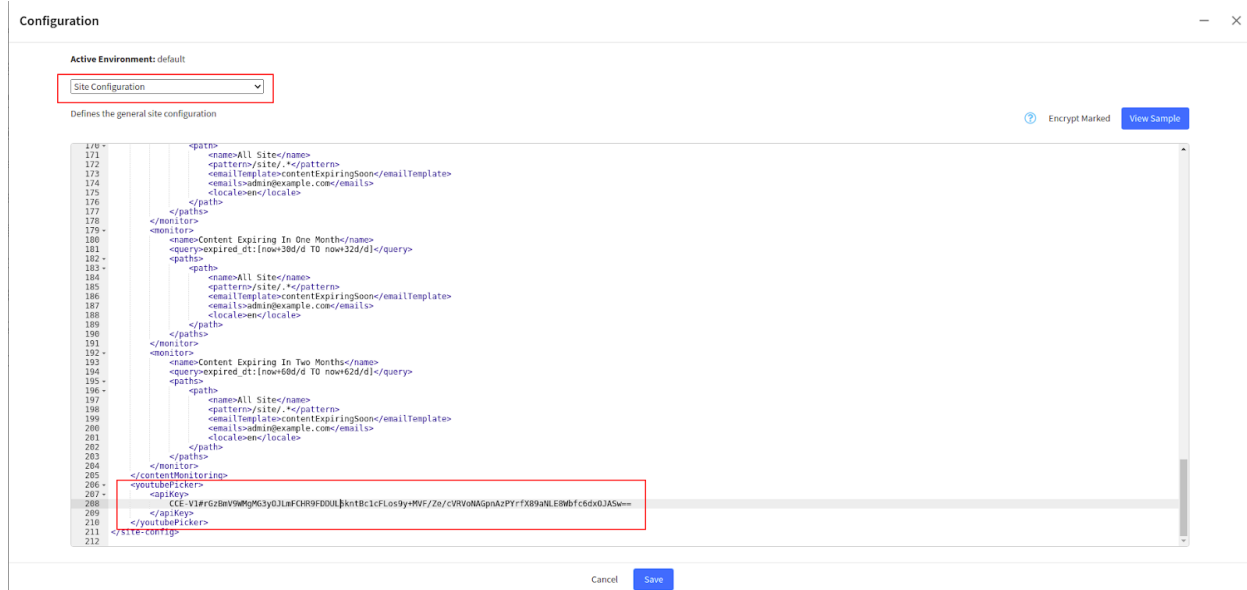
*Note: To know more about Encryption Tool, check out [this documentation](#).*

## Update site-config.xml

Go to your site Configuration, select Site Configuration then add the following line to the end of this file, just above the ending tag `</site-config>`. Remember to add the correct ``apiKey`` with the value from the above-encrypted output.

*Note: The ``apiKey`` begins with ``CCE-...==``. We remove the variable holder ``${enc:}``.*

```
<youtubePicker>  
  <apiKey>CCE-V1#rGzBmV9WM.....==</apiKey>  
</youtubePicker>
```



## Update Groovy blacklist

Detail instructions can be found here: [Using a custom blacklist](#)

We need to comment out the following lines, then restart Studio.

```

# method java.net.URL.openConnection
...
# staticMethod org.codehaus.groovy.runtime.DefaultGroovyMethods getAt
java.lang.Object java.lang.String

```

## Update studio-config.yaml

We need to enable 2 beans:

- cstudioSiteServiceSimple: in order to read site-configuration.xml
- crafter.textEncryptor: in order to decrypt api key

```

# Indicates if access to beans should be restricted
studio.scripting.restrictBeans: false
# List of patterns for bean names that should be accessible for the scripts
(regexes separated by commas)
studio.scripting.allowedBeans: 'crafter.textEncryptor'

```

## Groovy Script

Here is the final source code of the Groovy script for this internal API:

```
import org.apache.commons.lang3.StringUtils

def result = [:]

def site = request.getParameter("siteId")
if (StringUtils.isEmpty(site)) {
    result.code = 400;
    result.message = "Invalid siteId"
    return result
}

def keyword = request.getParameter("keyword")
if (StringUtils.isEmpty(keyword)) {
    result.code = 400;
    result.message = "Invalid keyword"
    return result
}

def configurationService = applicationContext["configurationService"]
def textEncryptor = applicationContext["crafter.textEncryptor"]
def config = configurationService.legacyGetConfiguration(site, "site-config.xml");

if (config.youtubePicker != null && config.youtubePicker.apiKey != null) {
    def apiKeyEncrypt = config.youtubePicker.apiKey.trim()
    def apiKey = textEncryptor.decrypt(apiKeyEncrypt)

    def searchUrl = 'https://youtube.googleapis.com/youtube/v3/search?part=snippet&type=video&q=' +
keyword + '&key=' + apiKey
    def get = new URL(searchUrl).openConnection()
    def getRC = get.getResponseCode()

    if (getRC.equals(200)) {
        def data = get.getInputStream().getText()
        result.code = 200
        result.data = data
        return result
    }
} else {
    result.code = 400
    result.message = "Invalid YouTube API Key"
}

return null
```

