Integrating Godaddy with Teamchat

Step-1: Access and API key, API secret

The Godaddy API is not open to the public. API key and the API secret used in the project:

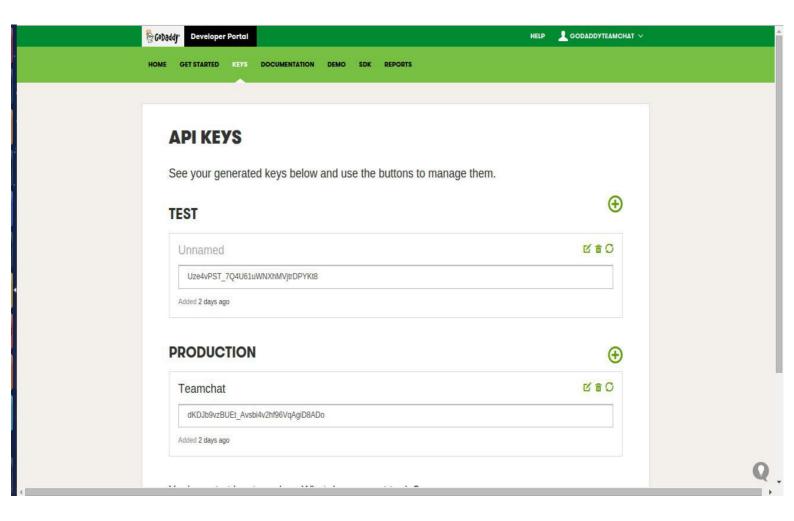| Key	
	dKDJb9vzBUEt_Avsbi4v2hf96VqAgiD8ADo
Secret	
	DuTSUzdf3NAFLqK8XHske7

This API key and API secret is required for all endpoints. This given API key and the API secret has been hardcoded into the project.

Note:- In this project API key and this API secret of my account is used and is in working condition so you dont have to change it in the project unless you want to do the integration with a API key of your account.

If incase this API key is not working or you want to use your new account's API key , then you have to perform certain steps:

- Go to this link : https://developer.godaddy.com/ and create an account

- After opening your account in the developers click on keys tab:

- There you will find a title known as "Production" , under this title you can add your product and can gain access to API key and API secret

- Then you just have to change the API key and API secret inside the the project.

- You just have to make changes in the "godaddyintegrator.java" file given in the package "com.teamchat.integration.godaddy.bot"

- Open the "godaddyintegrator.java" file then

```
J godaddyintegrator.java ⊠
 1  package com.teamchat.integration.godaddy.bot;
 4⊕  * @author:Anuj Arora
 6⊕ import java.io.IOException;

12  public class godaddyintegrator {
13      //getting the availability from the godaddy server
14⊝ public String getavailability(String domain) throws IOException
15  {
16          String err = "Error";
17          OkHttpClient client = new OkHttpClient();
18      //"sso-key {API_KEY}{API_SECRET}"
19
20      String API_KEY = dKDJb9vzBUEt_Avsbi4v2hf96VqAgiD8ADo;
21
22      String API_SECRET = "DuTSUzdf3NAFLqK8XHske7"
23
24      Request request = new Request.Builder()
25        .url("https://api.godaddy.com/v1/domains/available?domain="+domain.trim())
26        .get()
27        .addHeader("authorization", "sso-key "+API_KEY+":"+API_SECRET)
28        .build();
29
30      Response response = client.newCall(request).execute();
31
32       if (!response.isSuccessful())
33       {
34           return err;
35       }
36       else
37       return response.body().string();
38
39       }
40
41  }
```

API key

API secret

replace the API key and API secret present inside the red boxes with your new API key and API secret and you are ready to go.

Step-2: USING API-KEY AND API-SECRET TO CALL FUNCTIONS

Providing Parameters: GET requests should provide their parameters as a query string, part of the URL. API key and API secret should be provided in the header of the request.

JSON Responses: All responses are sent JSON-encoded.

METHOD: CHECK DOMAIN AVAILABILITY:

You can check if a specific domain is available.

URL: https://api.godaddy.com/v1/domains/available

Method : GET

Parameters : paramName="domain", value="Domain_name"

Headers : header="authorization", value="sso-key {API_KEY}{API_SECRET}"

Output:

```
{
  "price": 39990000,
  "currency": "USD",
  "period": 1,
  "available": false,
  "domain": "example.guru"
}
```

Code:

The following code gives a JSON Output which contains information about Searched domain.

```java
godaddyintegrator.java ⊠
 1  package com.teamchat.integration.godaddy.bot;
 4⊕ * @author:Anuj Arora☐
 6⊕ import java.io.IOException;☐
11
12  public class godaddyintegrator {
13      //getting the availability from the godaddy server
14⊖ public String getavailability(String domain) throws IOException
15  {
16          String err = "Error";
17          OkHttpClient client = new OkHttpClient();
18      //"sso-key {API_KEY}{API_SECRET}"
19
20      String API_KEY = "dKDJb9vzBUEt_Avsbi4v2hf96VqAgiD8ADo";
21
22      String API_SECRET = "DuTSUzdf3NAFLqK8XHske7";
23
24      Request request = new Request.Builder()
25          .url("https://api.godaddy.com/v1/domains/available?domain="+domain.trim())
26          .get()
27          .addHeader("authorization", "sso-key "+API_KEY+":"+API_SECRET)
28          .build();
29
30      Response response = client.newCall(request).execute();
31
32       if (!response.isSuccessful())
33       {
34           return err;
35       }
36       else
37       return response.body().string();
38
39      }
40
41  }
```

Step-3: Exporting the WAR file

Step-4: Deploying the WAR file

Step-5: Ready to use

Step-6: Follow the commands given below to use Godaddy:

Commands	Work
search	Type this command to search any domain's availability
help	It will tell user about the commands he/she should know before using godaddy

Note:-

1. **Getters and Setters class is used for extracting the data from json object. This classes is in the package "com.teamchat.integration.godaddy.classes".**

- **Godaddymain (main class for calling fields from the json object)**

2. **Libraries used:-**

- **gson 2.3.1 (for parsing json object)**

- **okhttp 2.4.0 (for authentication)**

- **okio 1.4.0 (prerequisite for okhttp)**

- **Teamchat Client SDK 1.5**

3. What changes should be done before running this project on any server?

- **Changes should be made in "web.xml" file , i.e your bot-email, bot-password, db-host, db-name, db-user and db-password as per your requirement**

- **No database or table used during the integration**

- **You only need to deploy the project's WAR file on the server**