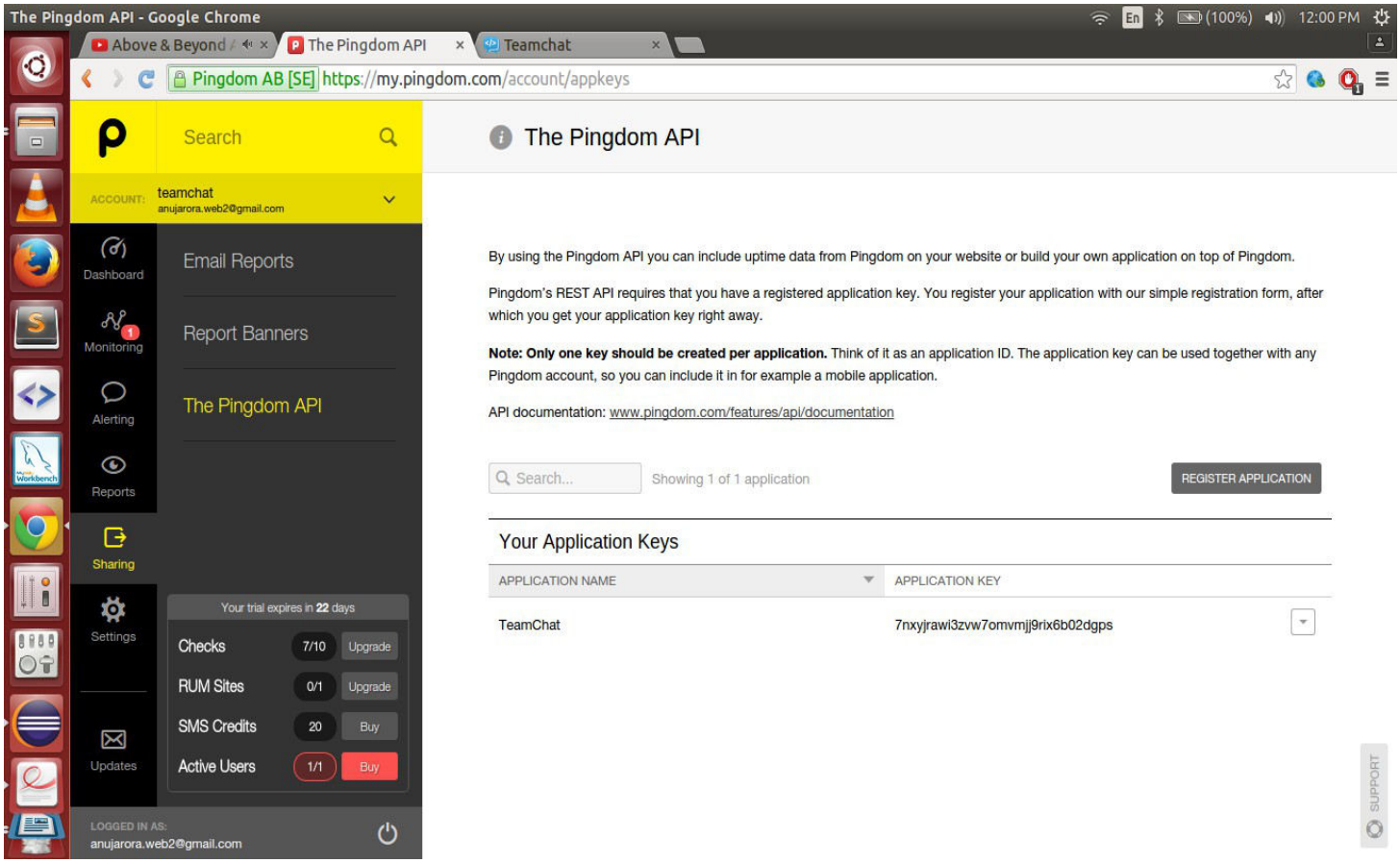
Integrating Pingdom with Teamchat

Step-1: CREATING A USER ACCOUNT

Register for a Pingdom account at http://www.pingdom.com. This will be your account for creating ,deleting and checking the status of your checks. You don't have to create any developer account for doing the integration with teamchat.There will be two options either you can make trial account for 30 days or you can pay them according to the plans they are offering and can get other benefits not available in your trial account.

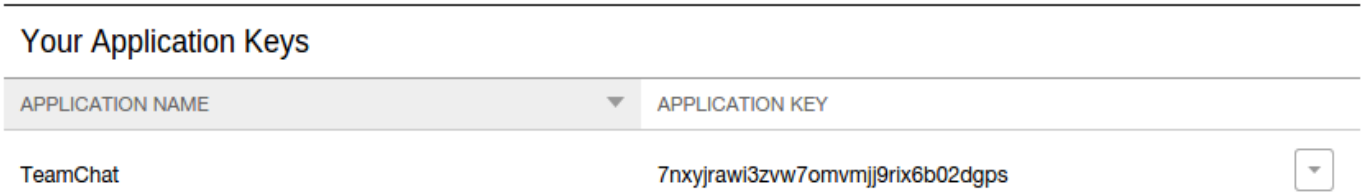
Step-2: CREATING A NEW APP

When you log in using your User account, at the home page you can see the option “Sharing”. After clicking on it, you will see an option “The pingdom API” , just choose that option then the follwing screen will appear:



There you will find an option “Register Application”, just click on that option and fill in the details of your application that you want to integrate. Once the app is created, your app is now assigned a Application-key. It is

supposed to be unique on an application basis, not user basis. This means that if you produce an application and then distribute it to the public, all users of this application should use the same application key.



Step-3: Authentication

The credentials used for accessing the API are the same used to access the Pingdom control panel - in other words your email address and your password. You will also need an application key. The authentication method for user credentials is HTTP Basic Access Authentication (encrypted over HTTPS). This means you will provide your credentials every time you make a request. No sessions are used. The application key is provided as a normal HTTP header in your request. Here is an example: App-Key: zoent8w9cbt810rsdkweir23vcxb87zrt5541. The base server address is: https://api.pingdom.com .

The above step is just to give an idea for the output of the given request to the pingdom API. The output is a JSON object. This object is parsed and the value of the required key is extracted.

Step-4: USING APP-KEY, USERNAME AND PASSWORD TO CALL FUNCTIONS

Providing Parameters: GET requests should provide their parameters as a query string, part of the URL.

POST, PUT and DELETE requests should provide their parameters as a query string. This should be part of the body, URL or a combination.

The encoding of the query string should be standard URL-encoding, as provided by various programming libraries.

HTTP/1.1 Status Code Definitions: The HTTP status code returned by a successful API request is 200 OK.

If something goes wrong, other codes may be returned. The API uses standard HTTP/1.1 status codes defined by RFC 2616.

JSON Responses: All responses are sent JSON-encoded.

However, if something goes wrong, their standard JSON error message (together with an appropriate status code) follows this format:

```
{
       "error":{
          "statuscode":403,
          "statusdesc":"Forbidden",
          "errormessage":"Something went wrong! This string describes what happened."
       }
    }
```

METHOD: GET CHECK LIST:

Returns a list overview of all checks.

URL : https://api.pingdom.com/api/2.0/checks/
Method : GET
Headers : header=“Auhorization”, value=”encoded_username_password”
header=“App-Key”, value=”application_key”

Output:


```
    "checks": [
        {
            "hostname": "example.com",
            "id": 85975,
            "lasterrortime": 1297446423,
            "lastresponsetime": 355,
            "lasttesttime": 1300977363,
            "name": "My check 1",
            "resolution": 1,
            "status": "up",
            "type": "http",
            "tags": [
                {
                    "name": "apache",
                    "type": "a",
                    "count": 2
                }
            ]
        },
        {
            "hostname": "mydomain.com",
            "id": 161748,
            "lasterrortime": 1299194968,
            "lastresponsetime": 1141,
            "lasttesttime": 1300977268,
            "name": "My check 2",
            "resolution": 5,
            "status": "up",
            "type": "ping",
            "tags": [
                {
                    "name": "nginx",
                    "type": "u",
                    "count": 1
                }
            ]
        },
        {
            "hostname": "example.net",
            "id": 208655,
            "lasterrortime": 1300527997,
            "lastresponsetime": 800,
            "lasttesttime": 1300977337,
            "name": "My check 3",
            "resolution": 1,
            "status": "down",
            "type": "http",
            "tags": [
                {
                    "name": "apache",
                    "type": "a",
                    "count": 2
                }
            ]
        }
    ]
}
```

Code:

The following code gives a JSON Output which contains information about all chcks present in the user account. You can parse this JSON Output and extract necessary information and display it to the user.

```
    }
//authenticating and recieving checks from your account.
  public String getChecks(String username,String password,String App_key,String url) throws IOException{

      String err = "Error";

    OkHttpClient client = new OkHttpClient();
    String auth = username + ":" + password;
    String encodedAuth = DatatypeConverter.printBase64Binary(auth.getBytes(Charset.forName("US-ASCII")));
    String authHeader = "Basic " + new String(encodedAuth);
    System.out.println("auth key: " + authHeader);
    Request request = new Request.Builder()
      .url(url)
      .get()
      .addHeader("authorization", authHeader )
      .addHeader("app-key", App_key)
      .build();

    Response response = client.newCall(request).execute();

     if (!response.isSuccessful())
     {
         return err;

     }

     else
    return response.body().string();
```

METHOD: CREATE NEW CHECK:

Creates a new check with settings specified by provided parameters.

URL : https://api.pingdom.com/api/2.0/checks/
Method : POST
Headers : header="Auhorization", value="encoded_username_password"
header="App-Key", value="application_key"

Parameters : paramName="name", value="Check_name"
paramName="host", value="URL"
paramName="type", value="HTTP/HTTPS"

Output:

```
{
        "check":{
           "id":138631,
           "name":"My new HTTP check"
        }
    }
```

Code:

The following code gives a JSON Output which contains information about recently added check.

```
public String addChecks(String username,String password,String App_key,String host, String name,String Protocol) throws IOException{

  String err = "Error";
  String success ="Check successfully added!!";
OkHttpClient client = new OkHttpClient();
String auth = username + ":" + password;
String encodedAuth = DatatypeConverter.printBase64Binary(auth.getBytes(Charset.forName("US-ASCII")));
String authHeader = "Basic " + new String(encodedAuth);
System.out.println("auth key: " + authHeader);
RequestBody formBody = new FormEncodingBuilder()
.add("name", name)
.add("host", host)
.add("type", Protocol)
.build();

Request request = new Request.Builder()
  .url("https://api.pingdom.com/api/2.0/checks")
  .post(formBody)
  .addHeader("authorization", authHeader)
  .addHeader("app-key", App_key)
  .build();

Response response = client.newCall(request).execute();
     if (!response.isSuccessful())
     {
         return err;
     }

     else
    return success;
}
```

METHOD: DELETE CHECK:

Deletes a check. THIS METHOD IS IRREVERSIBLE! You will lose all collected data. Be careful!

URL : https://api.pingdom.com/api/2.0/checks/{checkid}
Method : DELETE
Headers : header=“Auhorization”, value=”encoded_username_password”
header=“App-Key”, value=”application_key”

Output:


```
{
        "message":"Deletion of check was successful!"
    }
```

Code:

```
public String delChecks(String username,String password,String App_key,int[] del) throws IOException{
       String err = "Error";
      String success ="Deletion successfull!!";
    OkHttpClient client = new OkHttpClient();
    String auth = username + ":" + password;
    String encodedAuth = DatatypeConverter.printBase64Binary(auth.getBytes(Charset.forName("US-ASCII")));
    String authHeader = "Basic " + new String(encodedAuth);
    System.out.println("auth key: " + authHeader);
    int length,i,len;
    length= del.length;
    String delids="";
    for(i=0;i<length;i++){
        if(del[i]!=1){
            delids=delids+del[i];
            delids=delids+",";
        }
    }
    len=delids.length();
    delids=delids.substring(0, (len-1));
    System.err.println(delids);
    Request request = new Request.Builder()
      .url("https://api.pingdom.com/api/2.0/checks/?delcheckids="+delids)
      .delete(null)
      .addHeader("authorization", authHeader)
      .addHeader("app-key", App_key)
      .build();
    Response response = client.newCall(request).execute();
         if (!response.isSuccessful()){
             return err;
         }

         else
        return success;
}
```

Step-5: Exporting the WAR file

Step-6: Deploying the WAR file

Step-7: Ready to use

Step-8: Follow the commands given below to use Pingdom:

Commands	Work
help	It will tell user about the commands he/she should know before using Pingdom
ping	Type this command to Check the status for your website added in your pingdom account.

addcheck	Type this command to create any check i.e to add any website in your pingdom account.
deletecheck	Type this command to delete any check i.e to delete any website in your pingdom account.
Login	Type this command if you are logging for the first time from this room or you want to log in from other account.
logout	Type this command to log out from your existing account.

Note:-

1. **Getters and Setters classes are used for extracting the data from json object. These classes are in the package "com.teamchat.integration.pingdom.classes".**

- **Check**
- **Checks (main class for calling functions of other classes)**
- **Counts**

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 “pingdom-config.properties” file , i.e your botemail, botpass, dbname, dbpass and tablename should be changed according to your requirement**
- **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**
- **Table should be made in the same server in which you will deploy your project's WAR file**

SQL query:-

```
use Bot;
CREATE TABLE pingdom_auth (username VARCHAR(100) NOT NULL,
pass VARCHAR(32) NOT NULL,
appkey VARCHAR(100) NOT NULL,
roomId VARCHAR(256) NOT NULL
);
```

4. Bot will ask the user about the appkey and login credentials for the first time