|  |
| --- |
| <http://intelligent-book.ir/Account/EditProfile> |
| ورودی |
| IMEI, Email, Username, Name, Family, Mobile, Field, Degree, City, Job, pass |
| خروجی |
| {  "Error": 0,  "image": null  } |

ثبت نام

|  |
| --- |
| <http://intelligent-book.ir/Account/Register1> |
| ورودی |
| IMEI, Email, Username, Name, Family, Mobile, Field, Degree, City, Job, pass |
| خروجی |
| {  "Error": 0,  "image": null  } |

ورود

|  |
| --- |
| <http://intelligent-book.ir/Account/CheckUsername1> |
| ورودی |
| IMEI, Username |
| خروجی |
| {  "Error": 100  } |

لیست کامنتهای یک تاپیک

|  |
| --- |
| <http://intelligent-book.ir/Typic/Detail> |
| ورودی |
| IMEI, Username, pageidx,id |
| خروجی |
| {  "Comments": [  {  "id": 71249,  "typicid": 28,  "senderid": 1,  "senderName": "",  "senderUserName": null,  "senderImage": "",  "sender": null,  "tname": null,  "body": "سیبچه تمامی کتابها را تایید نموده. است دوستانی که از دستگاه های iphone وipad و ipod استفاده می کنند می توانند سایر کتابهایی که تهیه ننموده اند را نیز تهیه نمایند",  "like": 4,  "user": null,  "unlike": 0,  "dell": false,  "status": true,  "del": false,  "bad": false,  "datetime": "09:24:54 1394/02/28",  "type": 1  },  {  "id": 71245,  "typicid": 28,  "senderid": 1,  "senderName": "",  "senderUserName": null,  "senderImage": "",  "sender": null,  "tname": null,  "body": " کتاب ترمینولوژی بصورت غیر هوشمند به فروشگاه اضافه شد لیکن افرادی که از دستگاه اپل (سیستم عامل ios) استفاده می کنند تا زمان تایید سیبچه که حتی ممکنه تا ۷۲ ساعت به طول بیانجامد نمی توانند خرید نمایند",  "like": 2,  "user": null,  "unlike": 1,  "dell": false,  "status": true,  "del": false,  "bad": false,  "datetime": "20:28:49 1394/02/27",  "type": 1  }] |

|  |
| --- |
| کلاس زیر جهت ارسال اطلاعات کاربر به همراه عکس کاربر با استفاده از وب سرویسهای ویرایش پروفایل و ثبت نام استفاده میشود.  هنگام ساخت شی از این کلاس باید یک یوارال به عنوان ورودی به آن بدهیم که این یوارال با کمک دومین و کلیه پارامتر های مربوط به ثبت نام به جز عکس ساخته میشود.بعد از ساخت شی از این کلاس متدsend\_now  صدا زده میشود که یک استریم ورودی از فایل عکس را به عنوان ورودی دریافت میکند  در این کلاس یک استریم ورودی از فایل عکس داریم و یک استریم خروجی هم روی کانکشن ایجاد شده به یوارال ورودی فراهم میکنیم و فایل عکس با استفاده از استریم ورودی به صورت 1کیلوبایت 1کیلوبایت خوانده و روی استریم خروجی که کانکشن ما به سرور است نوشته میشود |
| **public** **class** HttpFileUpload **extends** Thread  {  URL connectURL;  String responseString;  String Title;  String Description;  **byte**[ ] dataToServer;  FileInputStream fileInputStream = **null**;  **long** size;  Handler h;    HttpFileUpload(String urlString,String vTitle,String vDesc,**long** size,Handler h)  {  **this**.size=size;  **this**.h=h;  **try**  {  connectURL = **new** URL(urlString);  Title= vTitle;  Description = vDesc;  }  **catch**(Exception ex)  {  Log.*i*("HttpFileUpload","URL Malformatted");  }  }  **void** Send\_Now(FileInputStream fStream)  {  fileInputStream = fStream;  start();  }  **void** Sending()  {  String iFileName = Title;  String lineEnd = "\r\n";  String twoHyphens = "--";  String boundary = "\*\*\*\*\*";  String Tag="fSnd";  **try**  {  Log.*e*(Tag,"Starting Http File Sending to URL");  HttpURLConnection conn = (HttpURLConnection)connectURL.openConnection();  conn.setDoInput(**true**);  conn.setDoOutput(**true**);  conn.setUseCaches(**false**);  conn.setRequestMethod("POST");  conn.setRequestProperty("Connection", "Keep-Alive");  conn.setRequestProperty("Content-Type", "multipart/form-data;boundary="+boundary);  DataOutputStream dos = **new** DataOutputStream(conn.getOutputStream());  dos.writeBytes(twoHyphens + boundary + lineEnd);  dos.writeBytes("Content-Disposition: form-data; name=\"title\""+ lineEnd);  dos.writeBytes(lineEnd);  dos.writeBytes(Title);  dos.writeBytes(lineEnd);  dos.writeBytes(twoHyphens + boundary + lineEnd);  dos.writeBytes("Content-Disposition: form-data; name=\"description\""+ lineEnd);  dos.writeBytes(lineEnd);  dos.writeBytes(Description);  dos.writeBytes(lineEnd);  dos.writeBytes(twoHyphens + boundary + lineEnd);  dos.writeBytes("Content-Disposition: form-data; name=\"uploadedfile\";filename=\"" + iFileName +"\"" + lineEnd);  dos.writeBytes(lineEnd);  Log.*e*(Tag,"Headers are written");  **if**(fileInputStream!=**null**)  {  **int** bytesAvailable = fileInputStream.available();  **int** maxBufferSize = 1024;  **int** bufferSize = Math.*min*(bytesAvailable,maxBufferSize);  **byte**[] buffer =**new** **byte**[bufferSize];  **int** bytesRead =fileInputStream.read(buffer,0,bufferSize);  **long** step=bufferSize\*100/size;  **int** sum=bufferSize;  **while** (bytesRead>0)  {  dos.write(buffer,0,bufferSize);  bytesAvailable = fileInputStream.available();  bufferSize = Math.*min*(bytesAvailable,maxBufferSize);  sum=sum+bufferSize;  bytesRead = fileInputStream.read(buffer,0,bufferSize);  step=sum\*100/size;  }  dos.writeBytes(lineEnd);  dos.writeBytes(twoHyphens+boundary+twoHyphens+lineEnd);  fileInputStream.close();  dos.flush();  }  Log.*e*(Tag,"File Sent,Response:"+String.*valueOf*(conn.getResponseCode()));  **try**  {  InputStream is = conn.getInputStream();  **int** ch;  StringBuffer b =**new** StringBuffer();  **while**((ch = is.read()) != -1)  {  b.append((**char**)ch);  }  String s=b.toString();  Message msg=**new** Message();  msg.obj=s;  h.sendMessage(msg);  }  **catch** (Exception e)  {  Message msg=**new** Message();  msg.obj="Error";  h.sendMessage(msg);  }  dos.close();  }  **catch** (MalformedURLException ex)  {  Log.*e*(Tag, "URL error: " + ex.getMessage(), ex);  }  **catch** (IOException ioe)  {  Log.*e*(Tag, "IO error: " + ioe.getMessage(), ioe);  }  }  @Override  **public** **void** run()  {  Sending();  }  } |