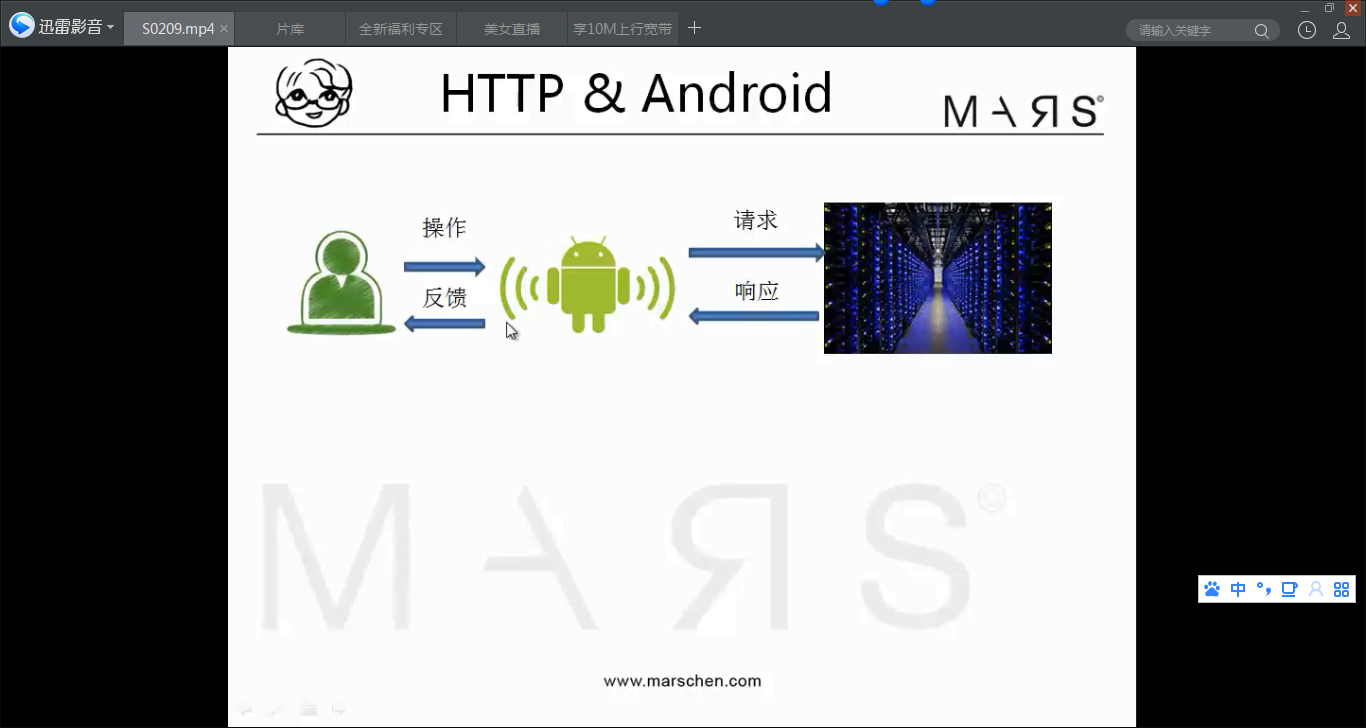
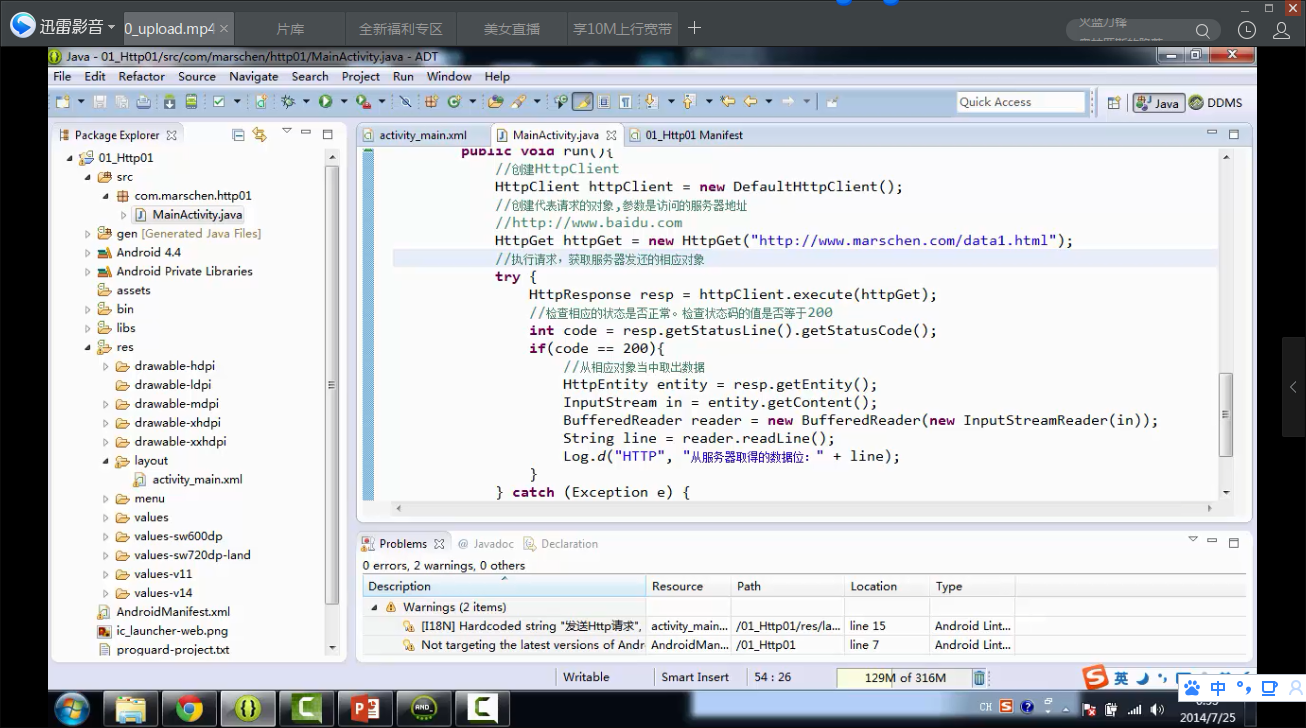
**网络**



**不允许在主线程中访问网络**

新线程中：

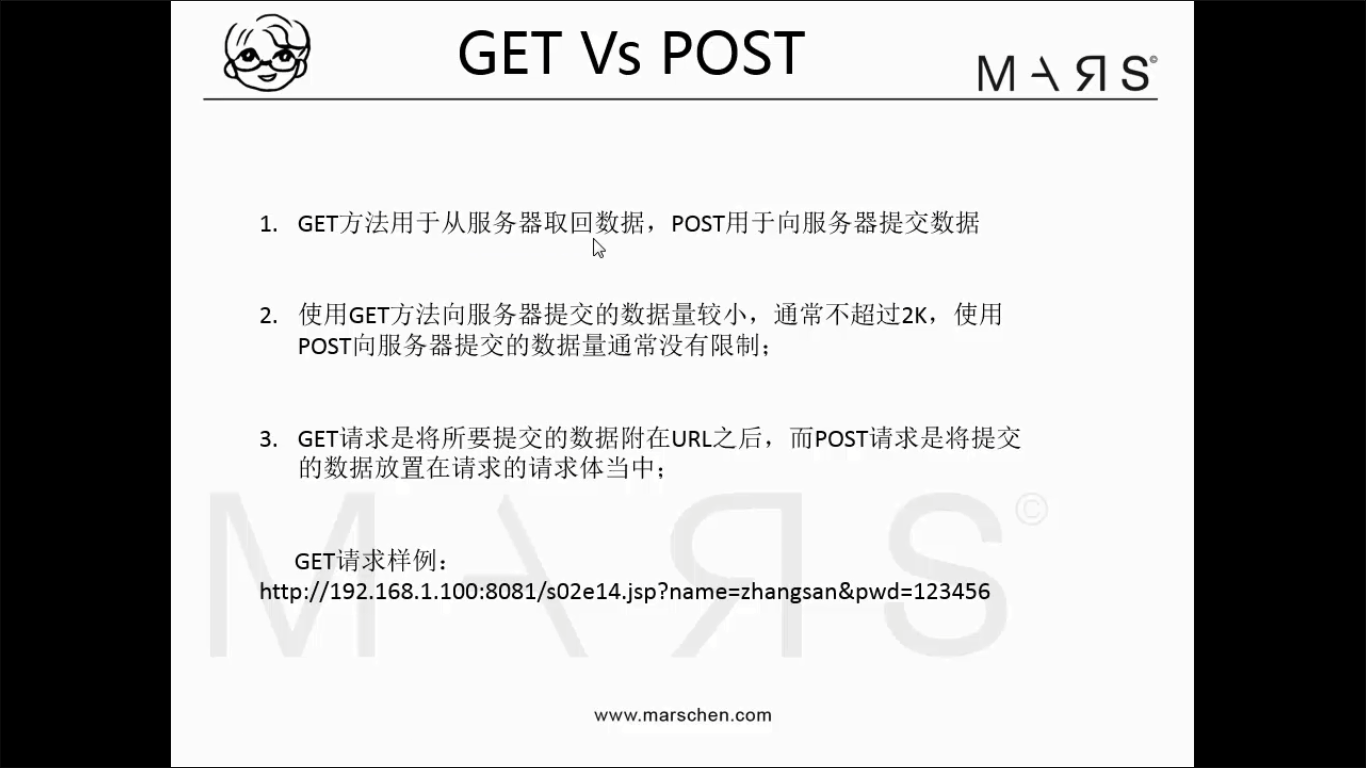


**请求头**



**响应头**





**Get 只有请求头没有请求体**

**1. Http Get**

**ReadUrl( String url)**{

new AsyncTask<初始化传入数据的数据类型String，Void处理过程中返回值的类型，String返回结果的数据类型>{

**doInBackground**(初始化传入数据的数据类型String… arg0){ //执行后台的耗时操作，… 代表数组

URL url=new URL(arg0[0]); //arg0数组中第一个元素就是url‘

URLConnection connection= url.openConnection(); //获取当前的网络连接

InputStream in=connection.getInputStream();

InputStreamReader reader=new InputStreamReader(in, “utf-8”);

BufferedReader bufferReader= new BufferedReader(reader);

String line;

StringBuilder stbuilder=new StringBuilder();

while((bufferReder.readLine())!=null){ //先赋值给line，在判断当前行是否有数据

stbuilder.append(line);

}

bufferReader.close();

reader.close();

in.close|();

return stbuilder.toString();

}.execute(url);

}

**回调方法**

**e.g.**

**onPostExecute**(String result){ //result就是doInBackground的返回值

}

}

**2. HttpClient Get**

**使用10.0.2.2代替8080可以使安卓访问本地的localhost**

//用异步任务实现

readNet(String url){

HttpClient client= new DefaultHttpClient();

AsyncTask<String, Void, String>(){

**doInBackground**(String… arg0){

String urlStr=arg0[0];

HttpGet get =new HttpGet(urlStr);

HttpResponse response= client.execute(get);

String value = EntityUtils.toString(Response.getEntity()); //getEntity无法直接调用toString()

}.execute(url);

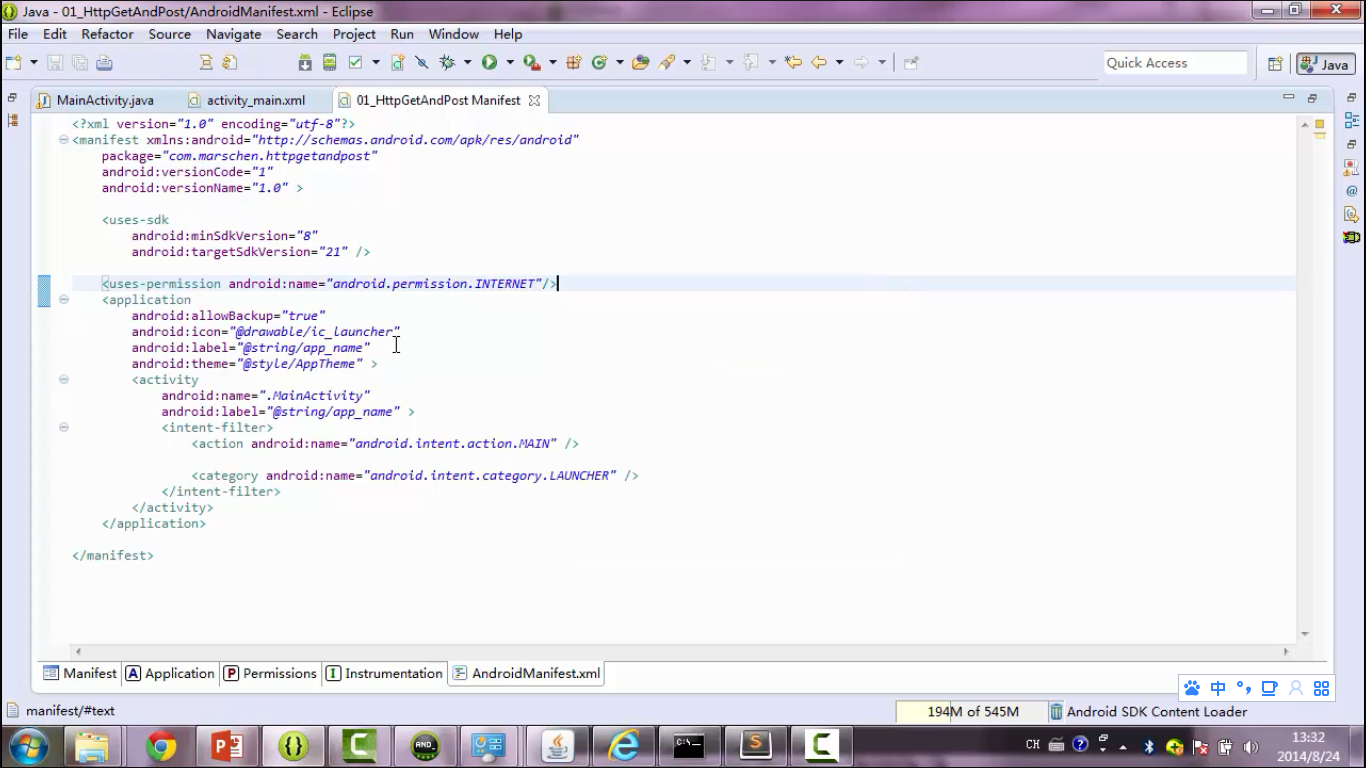
**onPostExecute**(String result){ //result就是doInBackground的返回值

对value进行操作。

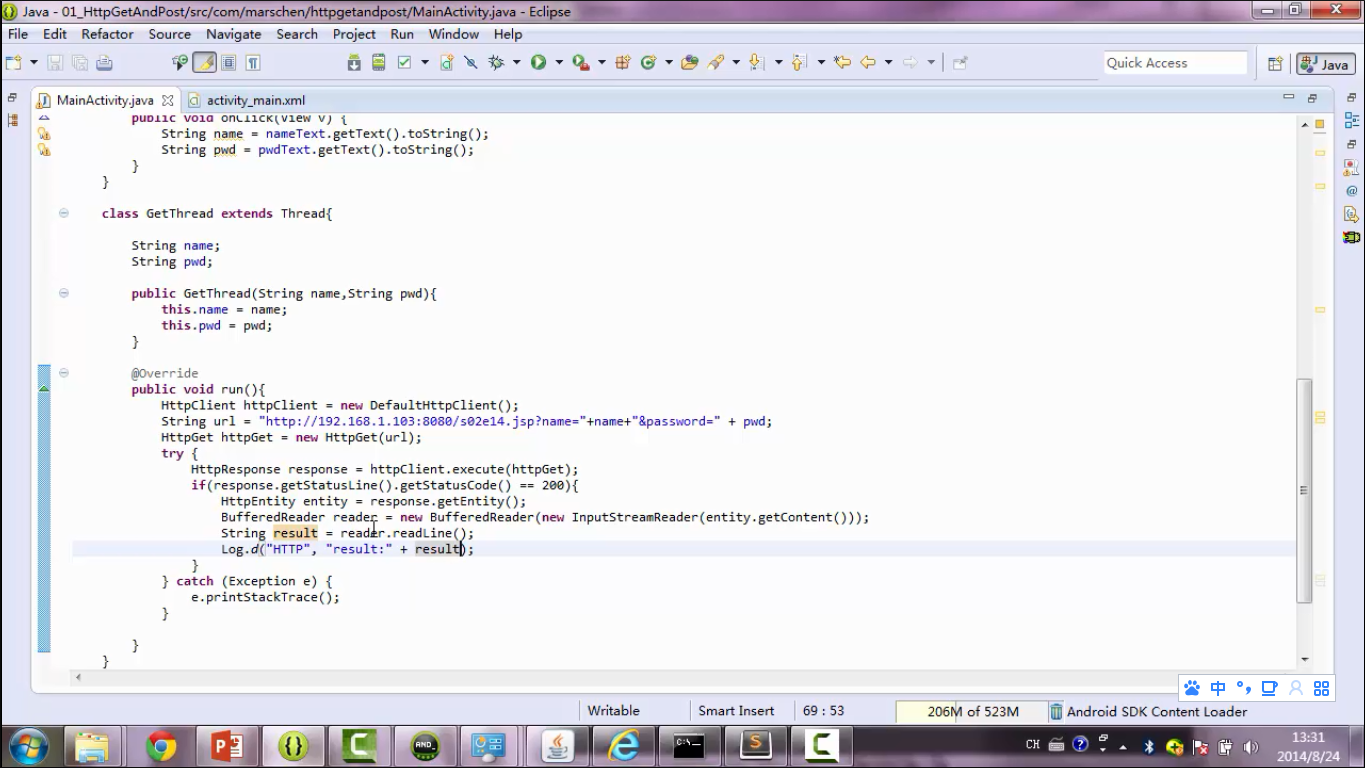
}

}

要获取网络连接许可



**//用新的线程实现**



**Post**

**1. Http Post**

**ReadUrl( String url)**{

new AsyncTask<初始化传入数据的数据类型String，Void处理过程中返回值的类型，String返回结果的数据类型>{

**doInBackground**(初始化传入数据的数据类型String… arg0){ //执行后台的耗时操作，… 代表数组

URL url=new URL(arg0[0]); //arg0数组中第一个元素就是url‘

URLConnection connection= url.openConnection(); //获取当前的网络连接

// connection.setDoInput(true); 是默认的可以不写

connection.setDoOutput(true);’ //配置connection

connection.setRequestMethod(“Post“);

//将json数据传递到服务器

OutputStreamWriter wt=new OutputWriter(Connection.getOutputStream(),”utf-8”);

BufferedWriter bw=new BufferedWriter(wt)

bw.write(….); //json数据

bw.flush();

InputStream in=connection.getInputStream();

InputStreamReader reader=new InputStreamReader(in, “utf-8”);

BufferedReader bufferReader= new BufferedReader(reader);

String line;

StringBuilder stbuilder=new StringBuilder();

while((bufferReder.readLine())!=null){ //先赋值给line，在判断当前行是否有数据

stbuilder.append(line);

}

bufferReader.close();

reader.close();

in.close|();

return stbuilder.toString();

}.execute(url);

}

**回调方法**

**e.g.**

**onPostExecute**(String result){ //result就是doInBackground的返回值

}

}

**2. Http Client Post**

//用异步任务实现

readNet(**String url,String in**){ //两个字符串,因为get方法可以在url中？paraname=xxx来携带参数，而post方法只能分开传递

HttpClient client= new DefaultHttpClient();

AsyncTask<String, Void, String>(){

**doInBackground**(String… arg0){

String urlStr=arg0[0];

HttpPost post =new HttpPost(urlStr);

List<BasicNameValuePair> list=new ArrayList< BasicNameValuePair >();

List.add(new BasicNameValuePair(“xxx”, arg0[1]));

post.setEntity(new UrlEncodedFormEntity(list)); //让list中的每一个键值对被格式化为表单传递出去

HttpResponse response= client.execute(post);

String value = EntityUtils.toString(Response.getEntity()); //getEntity无法直接调用toString()

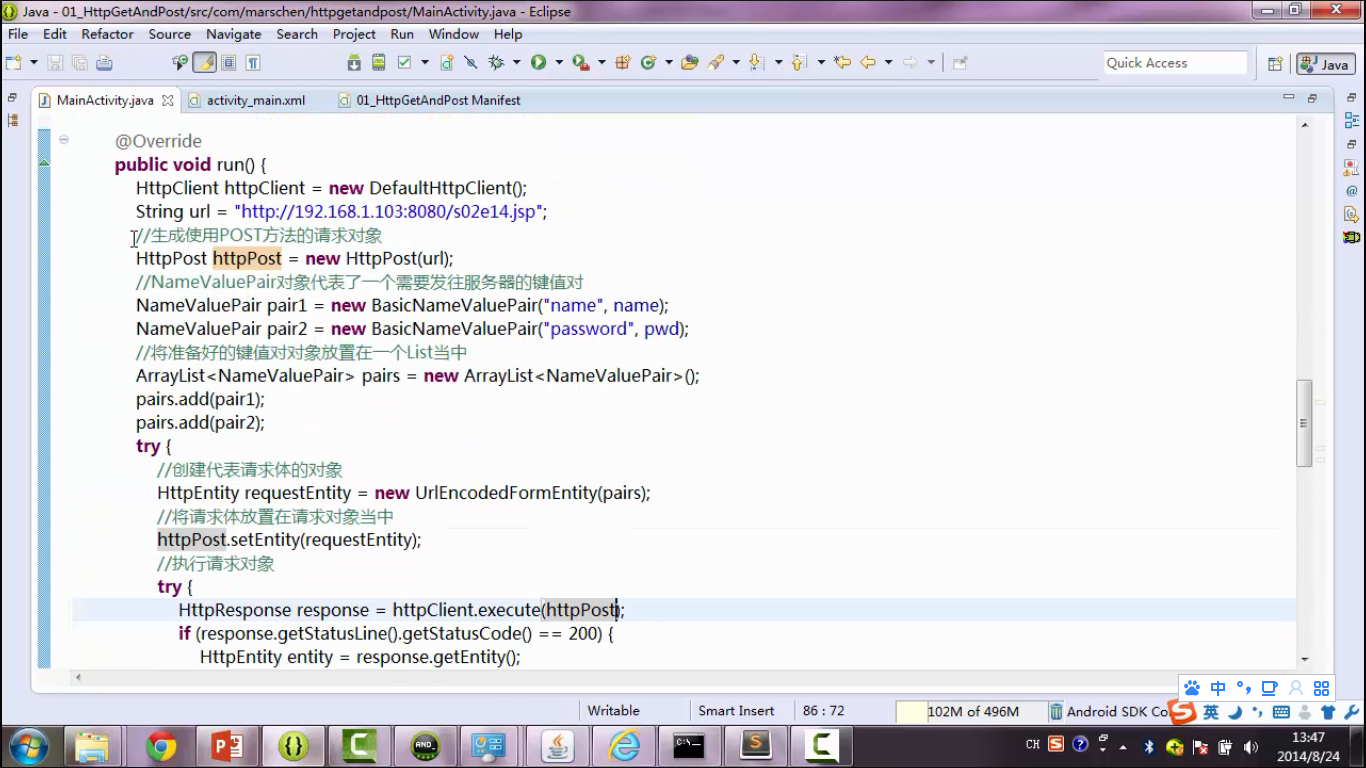
}.execute(url,in);

**onPostExecute**(String result){ //result就是doInBackground的返回值

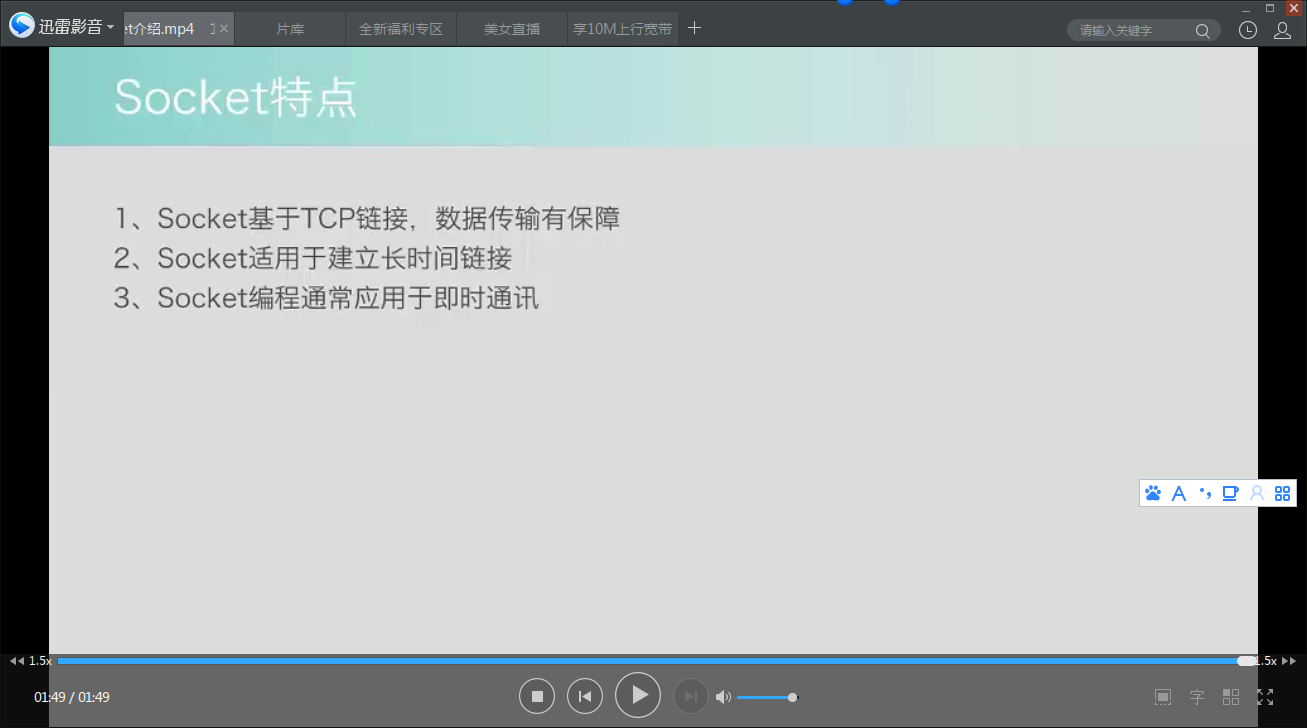
对value进行操作。

}

}



**SOCKET**



**扫码登录**