

## "Dual asynchronous HTTP request in Xamarin Android".

This project is designed to search for information on YouTube. If in the request to YouTube the option "&alt" set to "json" the response from YouTube will come in the form of a JSON object, which is easy then to collate items.

```
_url =String.Format( "http://gdata.youtube.com/feeds/api/videos?vq={0}" +  
"&alt=json&orderby=rating&start-index=11&max-results=10",channel);
```

This project was implemented to search and display images corresponding to the query string. The main highlight of this project is to implement a dual asynchronous request. The first asynchronous request is the request of the JSON object.

```
WebClientweb = newWebClient();  
  
web.DownloadStringCompleted += (s, e) =>    //First asynchronous request  
  
{  
  
...
```

Decomposition JSON object elegantly implemented using LINQ.

```
...  
  
JObject jsnObj = JObject.Parse(json);  
  
varjCollection = jsnObj["feed"]["entry"].Children();  
  
vardata = fromp injCollectionselectp;  
  
foreach(JObject p indata)  
  
{
```

```

YTImageData dataN = newYTImageData();

dataN.imgUrl = (string)p["media$group"]["media$thumbnail"][1]["url"];

//In this particular case we are interested in the address of the required images, which
// we use as an argument in the program of loading bmp file.

dataN.imgTitle=(string)p["media$group"]["media$title"]["$t"];

lstYTImageData.Add(dataN);
}

...

```

After receiving and decomposition into elements the second asynchronous procedure call downloads the pictures.

```

DownloadYTImage (stringurl);           //Second asynchronous request

```

To build the form of display used the GridView template. As elements of the templates used ImageView and TextView. These elements are involved in the use with *public class ImageYTAdapter*, which is built on the basis of the *BaseAdapter<askImageYT.YTImageData>* class. The templates of the GridView are populated with data using the method

```

public override View GetView (intposition, View customView, ViewGroup parent).

```

The finding of the author in this project was the use of all of the above actions within a block

```

RunOnUiThread(() =>
{

```

Attempt to implement this project without this instruction did not succeed. Perhaps there are other solutions, but this gives the result in the form of practical implementation, which is represented by the screenshots.