

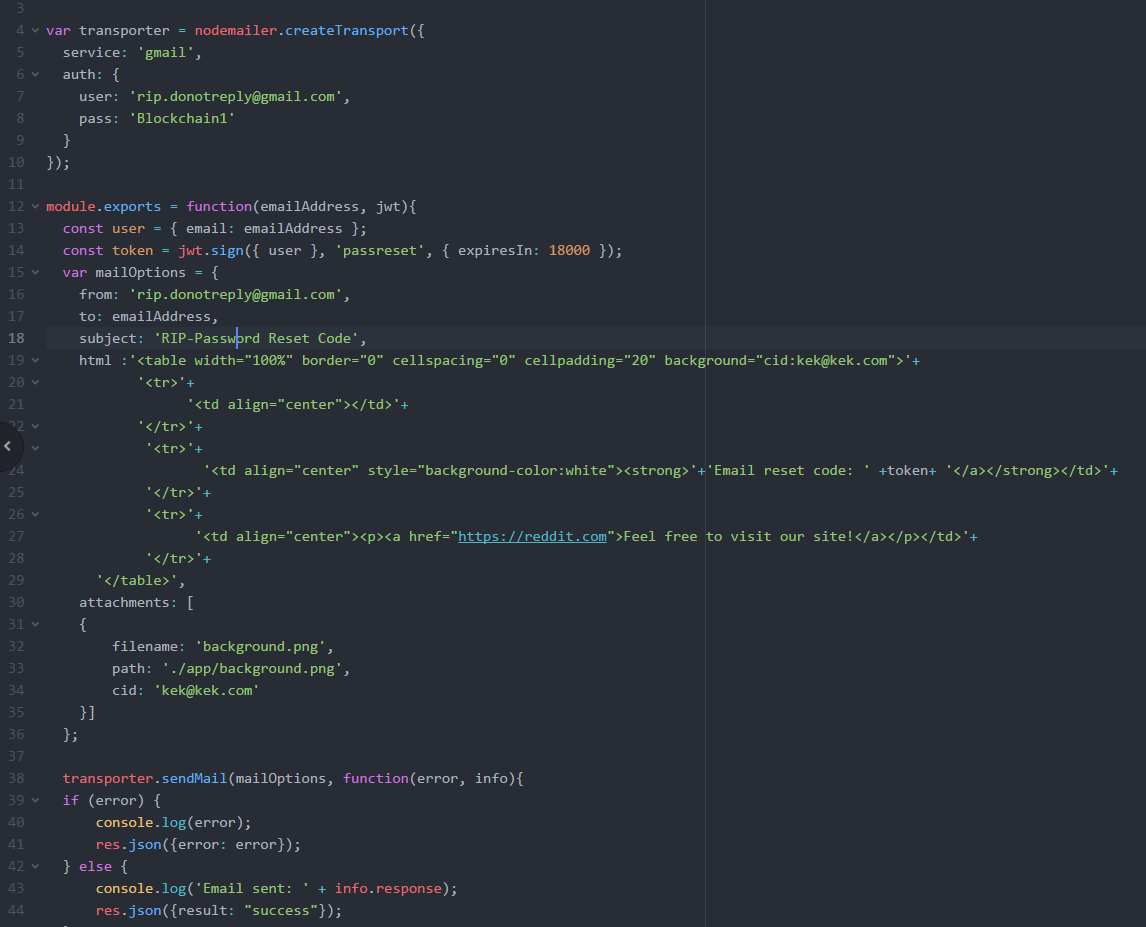
Packages used in the Node.js webservice.js file, this is the main file that runs the web service, one uses many available packages to create a powerful platform for web services and API’s. express is a popular package for managing the get and set requests and lets you use other packages such as cors (which allows headers from a certain web address/IP) and express-fileupload which we used to upload music files to the server.



The above snip shows the connection to the MySQL database hosted alongside this one on a AWS instance, as shown the it creates a connection pool, and every request then gets assigned a connection, the limit is 30 concurrent connections, this can be increased as demand increases and more powerful computing solutions become available.



The above image is the login module, this module (login.js) can be called from webservice.js and checks if the user is in the database, it receives a hashed password from the clients, and then compares it to a password and username combination in the database. If there is such a user, it then checks if that user’s account has been verified, if not the it shows a corresponding message.



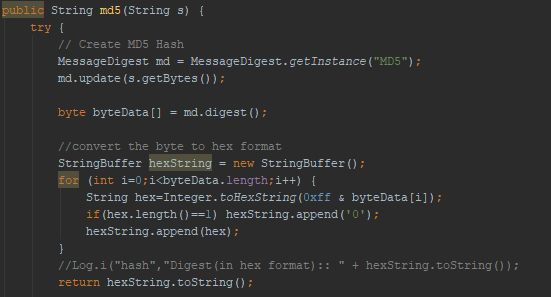
The above is an example of an email module that can be called to send an email corresponding to the given email address, by our own gmail account ([rip.donotreply@gmail.com](mailto:rip.donotreply@gmail.com)) and is formatted in html to look more professional.



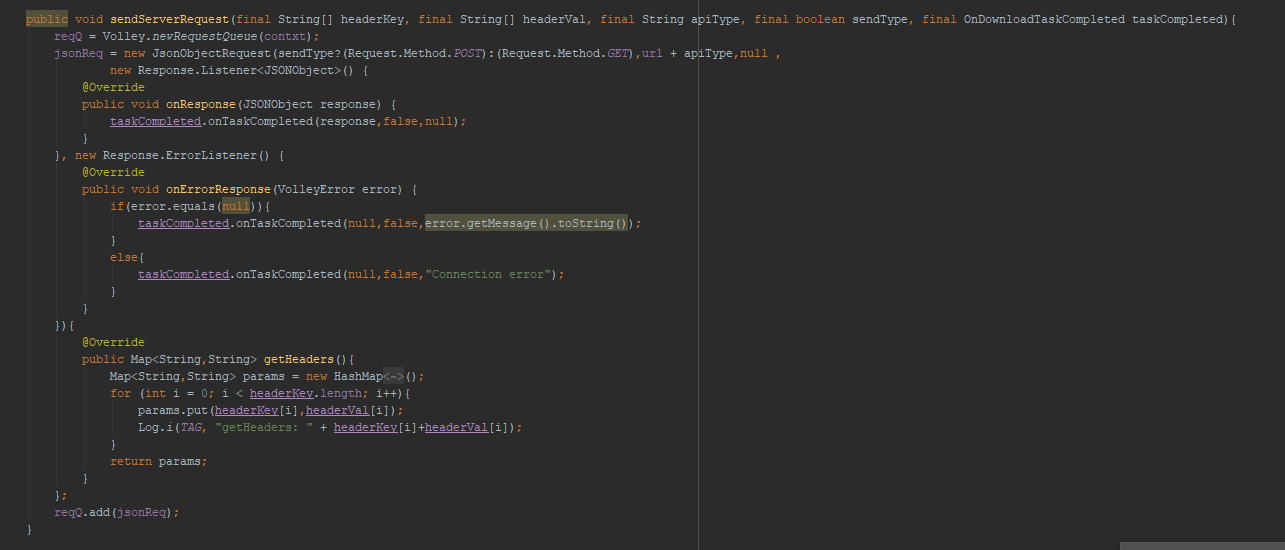
Used for uploading a complete album, this is used by the user/client to upload a complete album, it first checks that there is indeed files in the request, then creates a path from given data (artist/user name, album name) then moves the files into that folder. The for loop moves each song inside the songs array into that folder and when the album art is uploaded afterwards a success JSON is returned



Streams music with the media server package, it searches for the requested song, then plays that song if the given token is valid.

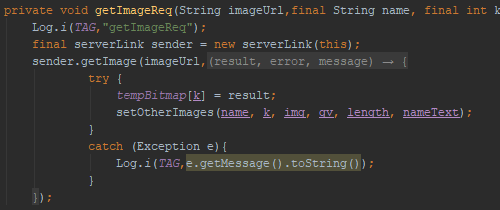


Creating the MD5 algorithm to create the password hash, it creates a byte array, then converts it to a hex string and returns that string to the calling method.

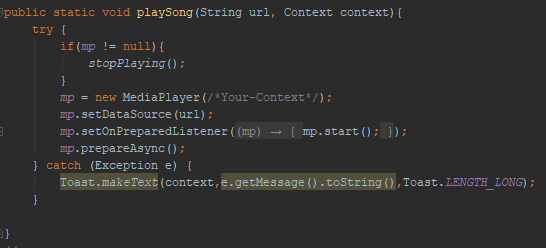


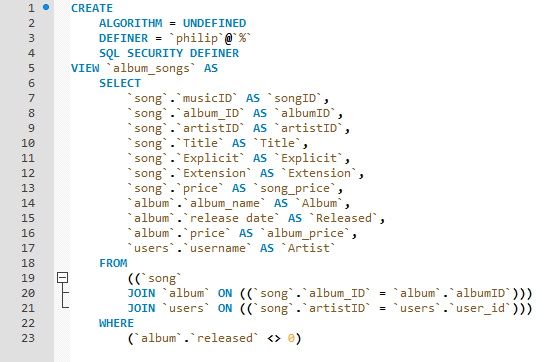
The above image comes the serverlink class, it creates the link to the webservice, and uses google’s volley library, this example returns a JSON object. The calling functions parameters include 2 string arrays which represent the headers that could be used. The 3rd value is a true or false representing get or post, which is the only 2 types of requests our web service uses at this point in time. When the volley request returns the value (They create their own new thread so thread handling isn’t our problem) the onDownloadTaskCompleted is called, that is an interface, and that allows us to use the following in other classes to connect with the web service:  
 

This creates an object of serverlink that waits for the response by implementing the interface ondownloadcomplete and lets the request run in the background without disturbing the user form (prevents crashing and freezing).

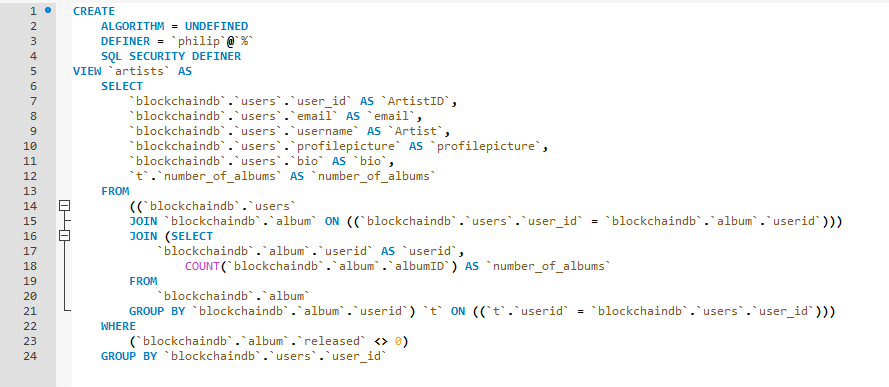


Above snip loads an image request also using volley and server link bvut instead of JSON object it awaits an image reply.

playing image using mp (object of mediaplayer) to play music from supplied url (sends request to webservice with url to stream music)



The above is an example of a view in the MySQL DB, named album\_songs, it shows all important info connected to an album, showing each song in with their album and artist.



The above is the view artists, and shows each artists amount of albums, and other important info such as profile picture and bio.