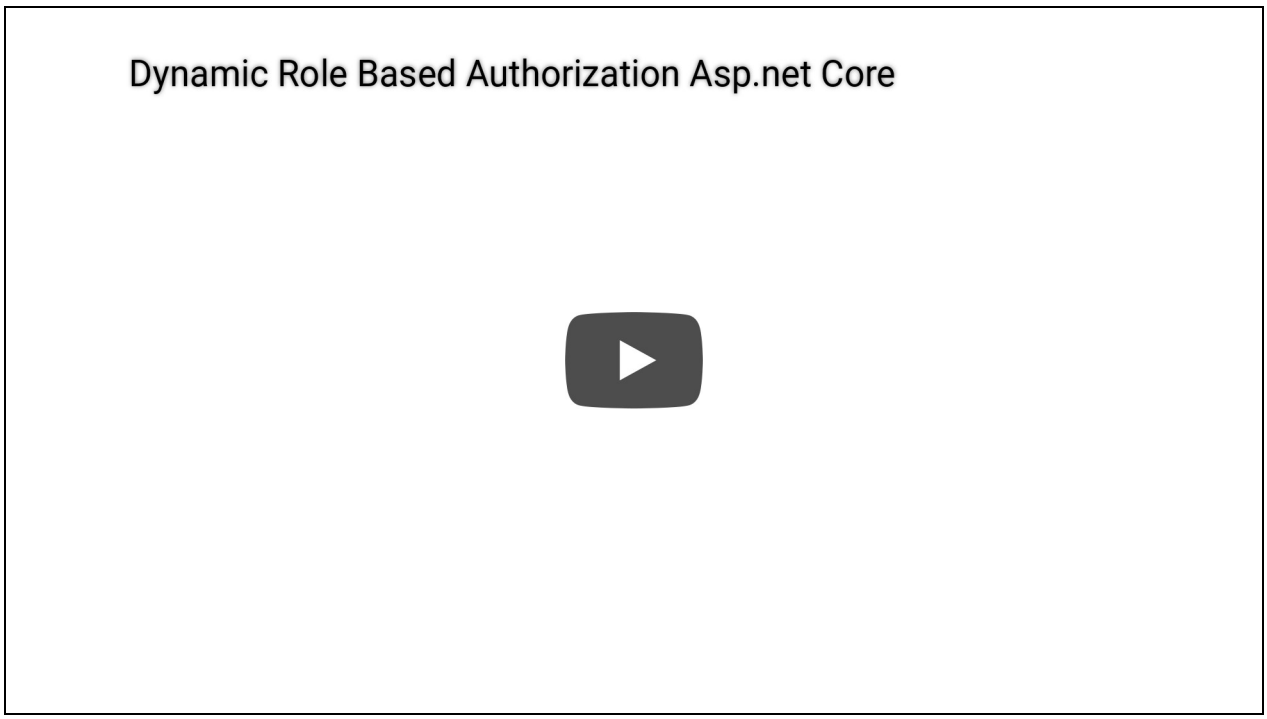
DYNAMIC ROLE BASED AUTHORIZATION ASP.NET CORE | ASSIGN ROLE FROM DB

NOVEMBER 24, 2018

Often times, after you've authenticated your user, now you want to authorize what he actually has control over based on his role. A user should only have access for what he's authorized to control.



In Applications like Admin Panel where your Application is being managed by multiple users, you must manage your users according to their roles.

Typically in an **ASP.NET MVC** project, this can be achieved by using an authorize attribute, Something like this:

```
1.  [Authorize(Roles = "Manager, SecondUser")]
2.  public ActionResult HelloWorld()
```

Now Only "Manager" & "SecondUser" can access the HelloWorld Action.

But what's the problem here?

Actually, the above code ties the **"Manager" & "SecondUser"** with your Controller & Action. So If you want to change User Role in Future, you have to update your code

and publish a new version of your Application.

So In this Tutorial, we'll see the better way to implement Role based Authorization & dynamically Create & Assign Roles to Users.

If you missed Admin Panel Tutorial, Find here => Creating Admin Panel in Asp.net Core MVC – Step by Step Tutorial

DIFFERENCE BETWEEN AUTHENTICATION & AUTHORIZATION

Before Start Implementing **Dynamic Role-based Authorization** we must know the difference between Authorization & Authentication.

Authentication is actually the process of validating Users Identity by verifying Credentials e.g. Username & Password. Whereas Authorization is the process to validate If a user has rights to access a specific action.

Authorization always comes after the Authentication process.

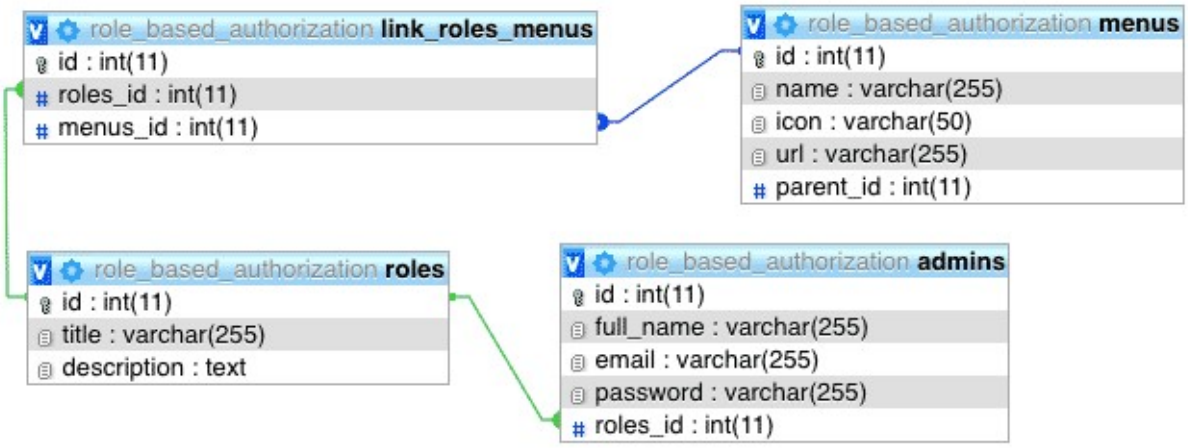
Now Let's see how you can Implement Dynamic Role-based Authorization using Asp.net.

DYNAMIC ROLE-BASED AUTHORIZATION

DATABASE STRUCTURE IS IMPORTANT

The database structure is really Important for this. First of all, we should have an **Admins** Table where we'll add your users & a **Roles** Table for adding Roles for Users.

Menus Table is for defining all actions or menus & **link_roles_menus** is for assigning access to roles.



Here's the Code Database Script. I have also shared the database script inside the GitHub repository.

```
-- phpMyAdmin SQL Dump
-- version 4.7.7
-- https://www.phpmyadmin.net/
--
-- Host: localhost
-- Generation Time: Nov 28, 2018 at 09:46 PM
-- Server version: 10.1.30-MariaDB
-- PHP Version: 7.2.1

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `role_based_authorization`
```

```
--

-- --------------------------------------------------------

--
-- Table structure for table `admins`
--

CREATE TABLE `admins` (
  `id` int(11) NOT NULL,
  `full_name` varchar(255) DEFAULT NULL,
  `email` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `roles_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `admins`
--

INSERT INTO `admins` (`id`, `full_name`, `email`, `password`, `roles_id`) VALUES
(1, 'Shehryar Khan', 'shehryarkn@gmail.com', '12345', 1),
(2, 'Ahsan Saeed', 'ahsansaeed067@gmail.com', '12345', 2),
(3, 'Shayan tahir', 'shayan@codinginfinite.com', '12345', 6);

-- --------------------------------------------------------

--
-- Table structure for table `link_roles_menus`
--

CREATE TABLE `link_roles_menus` (
  `id` int(11) NOT NULL,
  `roles_id` int(11) NOT NULL,
  `menus_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `link_roles_menus`
--

INSERT INTO `link_roles_menus` (`id`, `roles_id`, `menus_id`) VALUES
(47, 2, 1),
(48, 2, 2),
(49, 2, 4),
(50, 2, 5),
(51, 2, 6),
(52, 2, 7),
(65, 1, 1),
(66, 1, 2),
(67, 1, 3),
```

```
(68, 1, 4),
(69, 1, 5),
(70, 1, 6),
(71, 1, 7),
(76, 6, 1),
(77, 6, 2),
(78, 6, 4);

-- --------------------------------------------------------

--
-- Table structure for table `menus`
--

CREATE TABLE `menus` (
  `id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `icon` varchar(50) NOT NULL,
  `url` varchar(255) DEFAULT NULL,
  `parent_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `menus`
--

INSERT INTO `menus` (`id`, `name`, `icon`, `url`, `parent_id`) VALUES
(1, 'Dashboard', 'fa fa-dashboard', '/', 0),
(2, 'Admins', 'fa fa-users', '#', 0),
(3, 'Create Admin', 'fa fa-plus', '/Admins/Create', 2),
(4, 'Manage Admins', 'fa fa-users', '/Admins/Index', 2),
(5, 'Roles', 'fa fa-lock', '#', 0),
(6, 'Create Role', 'fa fa-lock', '/Roles/Create', 5),
(7, 'Manage Roles', 'fa fa-lock', '/Roles/Index', 5);

-- --------------------------------------------------------

--
-- Table structure for table `roles`
--

CREATE TABLE `roles` (
  `id` int(11) NOT NULL,
  `title` varchar(255) NOT NULL,
  `description` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `roles`
--

```

```
INSERT INTO `roles` (`id`, `title`, `description`) VALUES
(1, 'Manager', 'Super Admin with all rights...'),
(2, 'Supervisor', 'Can View Dashboard, Admins & Roles'),
(6, 'Developer', 'Can View Dashboard &  Admins List');

--
-- Indexes for dumped tables
--

--
-- Indexes for table `admins`
--
ALTER TABLE `admins`
  ADD PRIMARY KEY (`id`),
  ADD KEY `admins_ibfk_1` (`roles_id`);

--
-- Indexes for table `link_roles_menus`
--
ALTER TABLE `link_roles_menus`
  ADD PRIMARY KEY (`id`),
  ADD KEY `menus_id` (`menus_id`),
  ADD KEY `roles_id` (`roles_id`);

--
-- Indexes for table `menus`
--
ALTER TABLE `menus`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `roles`
--
ALTER TABLE `roles`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `admins`
--
ALTER TABLE `admins`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT for table `link_roles_menus`
--
ALTER TABLE `link_roles_menus`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=79;
```

```
176.
177.  --
178.  -- AUTO_INCREMENT for table `menus`
179.  --
180.  ALTER TABLE `menus`
181.    MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;
182.
183.  --
184.  -- AUTO_INCREMENT for table `roles`
185.  --
186.  ALTER TABLE `roles`
187.    MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
188.
189.  --
190.  -- Constraints for dumped tables
191.  --
192.
193.  --
194.  -- Constraints for table `admins`
195.  --
196.  ALTER TABLE `admins`
197.    ADD CONSTRAINT `admins_ibfk_1` FOREIGN KEY (`roles_id`) REFERENCES `roles`
      (`id`);
198.
199.  --
200.  -- Constraints for table `link_roles_menus`
201.  --
202.  ALTER TABLE `link_roles_menus`
203.    ADD CONSTRAINT `link_roles_menus_ibfk_1` FOREIGN KEY (`menus_id`) REFERENCES
      `menus` (`id`),
204.    ADD CONSTRAINT `link_roles_menus_ibfk_2` FOREIGN KEY (`roles_id`) REFERENCES
      `roles` (`id`);
205.  COMMIT;
206.
207.  /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
208.  /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
209.  /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

HOW TO IMPLEMENT

As I have already said, the most important thing here is your Database Tables Linking. Above database is perfect for the easy to implement role-based authorization system.

After Creating a database, simply Add pages for “admins”, “roles”, “menus”

Tables with CRUD Operations.

Here's the Simple CRUD Operations Tutorial with Admin Template => CRUD Operations in Asp.net Core MVC

HOW TO CREATE MENU FROM DATABASE

As you can see in "menus" Table that It has a column with the name "parent_id" so It means that menus will be in a Tree format with the Parent-Child relations in the same Table. So I'll recommend you to Follow This Tutorial to load Data from Database to TreeView for Assigning Menus Access to a Role.

Here's the Code for Creating Menu from Database.

```
1.  public class AccountController : Controller
2.  {
3.      MyDbContext db = new MyDbContext();
4.
5.      public IActionResult Login()
6.      {
7.          return View();
8.      }
9.
10.     public ActionResult Validate(Admins admin)
11.     {
12.         var _admin = db.Admins.Where(s => s.Email == admin.Email).FirstOrDefault();
13.         if (_admin != null)
14.         {
15.             if (_admin.Password == admin.Password)
16.             {
17.                 HttpContext.Session.SetString("email", _admin.Email);
18.                 HttpContext.Session.SetInt32("id", _admin.Id);
19.                 HttpContext.Session.SetInt32("role_id", (int)_admin.RolesId);
20.                 HttpContext.Session.SetString("name", _admin.FullName);
21.
22.                 int roleId = (int)HttpContext.Session.GetInt32("role_id");
23.                 List<Menus> menus = db.LinkRolesMenus.Where(s => s.RolesId ==
    roleId).Select(s => s.Menus).ToList();
24.
25.                 DataSet ds = new DataSet();
26.                 ds = ToDataSet(menus);
27.                 DataTable table = ds.Tables[0];
28.                 DataRow[] parentMenus = table.Select("ParentId = 0");
```

```
29.
30.                 var sb = new StringBuilder();
31.                 string menuString = GenerateUL(parentMenus, table, sb);
32.                 HttpContext.Session.SetString("menuString", menuString);
33.                 HttpContext.Session.SetString("menus",
    JsonConvert.SerializeObject(menus));
34.
35.                 return Json(new { status = true, message = "Login Successfull!" });
36.             }
37.             else
38.             {
39.                 return Json(new { status = true, message = "Invalid Password!" });
40.             }
41.         }
42.         else
43.         {
44.             return Json(new { status = false, message = "Invalid Email!" });
45.         }
46.     }
47.
48.     private string GenerateUL(DataRow[] menu, DataTable table, StringBuilder sb)
49.     {
50.         if (menu.Length > 0)
51.         {
52.             foreach (DataRow dr in menu)
53.             {
54.                 string url = dr["Url"].ToString();
55.                 string menuText = dr["Name"].ToString();
56.                 string icon = dr["Icon"].ToString();
57.
58.                 if (url != "#")
59.                 {
60.                     string line = String.Format(@"<li><a href=""{0}""><i class=""
    {2}""></i> <span>{1}</span></a></li>", url, menuText, icon);
61.                     sb.Append(line);
62.                 }
63.
64.                 string pid = dr["Id"].ToString();
65.                 string parentId = dr["ParentId"].ToString();
66.
67.                 DataRow[] subMenu = table.Select(String.Format("ParentId = '{0}'",
    pid));
68.                 if (subMenu.Length > 0 && !pid.Equals(parentId))
69.                 {
70.                     string line = String.Format(@"<li class=""treeview""><a
    href=""#""><i class=""{0}""></i> <span>{1}</span><span class=""pull-right-
    container""><i class=""fa fa-angle-left pull-right""></i></span></a><ul
    class=""treeview-menu"">", icon, menuText);
71.                     var subMenuBuilder = new StringBuilder();
72.                     sb.AppendLine(line);
73.                     sb.Append(GenerateUL(subMenu, table, subMenuBuilder));
74.                     sb.Append("</ul></li>");
```

```
                }
            }
        }
        return sb.ToString();
    }

    public DataSet ToDataSet<T>(List<T> items)
    {
        DataTable dataTable = new DataTable(typeof(T).Name);
        //Get all the properties
        PropertyInfo[] Props = typeof(T).GetProperties(BindingFlags.Public |
BindingFlags.Instance);
        foreach (PropertyInfo prop in Props)
        {
            //Setting column names as Property names
            dataTable.Columns.Add(prop.Name);
        }
        foreach (T item in items)
        {
            var values = new object[Props.Length];
            for (int i = 0; i < Props.Length; i++)
            {
                values[i] = Props[i].GetValue(item, null);
            }
            dataTable.Rows.Add(values);
        }
        DataSet ds = new DataSet();
        ds.Tables.Add(dataTable);
        return ds;
    }

    public ActionResult Logout()
    {
        HttpContext.Session.Clear();
        return RedirectToAction("Login", "Account");
    }
}
```

Validate Method is called when we click on the Login button on the Login Page. It Generates the Menu HTML based string keep in the Session.

HOW IT WILL WORK

On Successful Login, First of all, get the Role ID of the Admin, from Role ID get all **menus** list from Database which is allowed to the User & display in your Application Menu. Also, store in your Sessions memory.

Use **ActionFilterAttribute** to check before every Action that either this Action is Valid for the Logged in User or Not.

Here's the Code for **ActionFilterAttribute**

```
1.  public class AuthorizedAction: ActionFilterAttribute
2.  {
3.      public override void OnResultExecuting(ResultExecutingContext filterContext)
4.      {
5.
6.      }
7.
8.      public override void OnActionExecuting(ActionExecutingContext filterContext)
9.      {
10.         base.OnActionExecuting(filterContext);
11.
12.         if (filterContext.HttpContext.Session.GetString("email") == null)
13.         {
14.             filterContext.Result = new RedirectToRouteResult(
15.                 new RouteValueDictionary { { "controller", "Account" }, { "action",
    "Login" } });
16.             return;
17.         }
18.
19.         var menus = JsonConvert.DeserializeObject<List<Menus>>
    (filterContext.HttpContext.Session.GetString("menus"));
20.         var controllerName = filterContext.RouteData.Values["controller"];
21.         var actionName = filterContext.RouteData.Values["action"];
22.         string url = "/" + controllerName + "/" + actionName;
23.
24.         if (!menus.Where(s => s.Url == url).Any())
25.
26.         {
27.             filterContext.Result = new RedirectToRouteResult(
28.                 new RouteValueDictionary { { "controller", "Account" }, { "action",
    "Login" } });
29.             return;
30.         }
31.     }
32. }
```

I would recommend using routing to not reload your whole page on every option-click.

Here's the Angular UI-Router Tutorial with Asp.net Core.

I'll recommend to Download Complete Code from GitHub & run it. In my opinion, If you are able to understand the Database, you can easily Implement the solution in your own way.

Download Complete Code

If you want any support from me please comment below. I'm always available to help you.

FAQ.

Q: HOW TO AUTHORIZE LINKS OTHER THAN MENU?

Ans: Add another column in the **"Menus"** Table as **"is_menu"** & when getting menus list on Successful login, filter the records only have **"is_menu" = 1.** Rest of the work will be as explained above.

Now you can Add all links(other than menu items), you want to authorize in **"Menus"** Table.

You might be interested in:

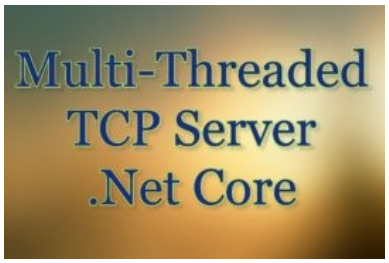
User's Activity Logging

AUTHOR

SHEHRYAR KHAN

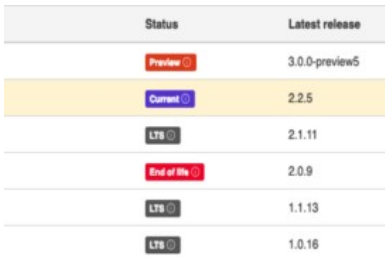
I'm passionate about learning new technologies as well as mentoring and helping others get started with their programming career. This blog is my way of giving back to the Community.

RELATED POSTS



MULTI-THREADED TCP SERVER USING DOTNET CORE EXAMPLE | C#

JUNE 3, 2019



SELECTING DOTNET CORE SDK VERSION WHILE CREATING A NEW PROJECT

MAY 23, 2019

THE ULTIMATE GUIDE TO THE BEST .NET CORE E-COMMERCE OPEN SOURCE PLATFORMS

MAY 12, 2019

10 COMMENTS

HARIOM REPLY

4 WEEKS AGO

i have done all thing that is mentioned in your code but i m struggling with one problem is that :
In my code have a dropdown in a form and it bind from database so to open that form then i have to save form url as well as dropdown url while assigning authority to Role.
Please suggest me the solution for this

SHEHRYAR KHAN POST AUTHOR REPLY

2 MONTHS AGO

QSL Script is available above.

JOHN REPLY

2 MONTHS AGO

I am having a problem creating the database in SQL server, do you have a script for that?

STEVE REPLY

8 MONTHS AGO

Hi Shehryar, great article thanks mate. I have been trying to build a MVC app just like this for a few days and your article really helped – Thankyou. I also wouldn't mind seeing the code, just to see how a pro does it 🙂
Thanks again, great article. Steve

SHEHRYAR KHAN POST AUTHOR REPLY

8 MONTHS AGO

Thanks a lot Steve. I'm glad that my Article helped you. I'll share complete solution on GitHub soon.

DIMITAR MIHOV REPLY

8 MONTHS AGO

So essentially you are making a predefined database structure that has a filter table like. Now rather than editing source code to add authorizations we submit new roles via Entity framework to the database as entries and our "filter-system" already Enforces the Authorizations, correct?

SHEHRYAR KHAN POST AUTHOR

REPLY

8 MONTHS AGO

Yeah, Right.

SHEHRYAR KHAN POST AUTHOR

REPLY

8 MONTHS AGO

Thanks for your interest. I'll share Complete Code within the next couple of days. keep connected.

MOSES

REPLY

8 MONTHS AGO

Hi, please provide the code for better understanding and implemenation

Thank you very much

JONATHAN LUSTGARTEN

REPLY

8 MONTHS AGO

I would appreciate sample code from your great example. Thanks!

WRITE A COMMENT

Name | Email | Website

Enter your comment here..

Save my name, email, and website in this browser for the next time I comment.

Notify me when new comments are added.

POST COMMENT

Type and hit enter...

1.6K Facebook

Twitter

YouTube

Instagram

SUBSCRIBE

Subscribe to our newsletters to get an Email on every new Article

Email Address *

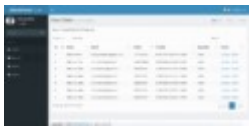
Choose Technology *

SUBSCRIBE

TOP ARTICLES

SIGNUP LOGIN PAGE IN PHP WITH DATABASE MYSQL SOURCE CODE

MAY 27, 2018



CRUD OPERATIONS IN ASP.NET CORE MVC

NOVEMBER 10, 2018

HERE ARE THE TEN BEST PROGRAMMING LANGUAGES TO LEARN IN 2019

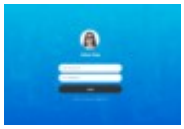
DECEMBER 22, 2018

OPEN SOURCE BULK SMS SENDER ANDROID APP

APRIL 16, 2018

CAR LOCATION TRACKING ANDROID APP WITH FIREBASE TUTORIAL

AUGUST 28, 2018



LOGIN PAGE IN ASP.NET CORE MVC WITH DATABASE

OCTOBER 31, 2018

CAR LOCATION TRACKING ANDROID APP WITH FIREBASE TUTORIAL | PART 2

AUGUST 28, 2018

^ TOP