//Клас, за допомогою якого взаємодіють клієнтська та серверна частини.

public class ScraperController : Controller

{

private IScrapeTaskRepository \_tasksRepository;

private IDataRepository \_dataRepository;

public UserManager<ApplicationUser> UserManager { get; private set; }

private ApplicationUser \_user;

public ScraperController(IScrapeTaskRepository tasksRepository, IDataRepository dataRepository, IUsersRepository usersRepository)

{

\_tasksRepository = tasksRepository;

\_dataRepository = dataRepository;

var store = new UserStore<ApplicationUser>(usersRepository.GetDefaultCollection());

UserManager = new UserManager<ApplicationUser>(store);

}

// Повертає головну сторінку програми

public ActionResult Index()

{

if (User.Identity.IsAuthenticated)

{

return View();

}

else

{

return RedirectToAction("Login", "Account");

}

}

// Повертає сторінку перегляду завдань

public ActionResult Tasks()

{

if (User.Identity.IsAuthenticated)

{

\_user = UserManager.FindByName(User.Identity.Name);

var tasks = \_tasksRepository.GetAll().Where(x => x.UserId == \_user.Id);

return View(tasks.ToList().OrderBy(x=>x.DateCreated));

}

else

{

return RedirectToAction("Login", "Account");

}

}

// Виконує перезапуск завершеного завдання

public ActionResult Restart(Guid id)

{

var task = \_tasksRepository.GetById(id);

task.Status = Status.Pending;

\_tasksRepository.Save(task);

\_dataRepository.RemoveDataForTask(id);

return RedirectToAction("Tasks", "Scraper");

}

//Видаляє існуюче задання

public ActionResult Remove(Guid? id)

{

if (!id.HasValue)

return RedirectToAction("List");

var task = \_tasksRepository.GetById(id.Value);

\_tasksRepository.Delete(task);

\_dataRepository.RemoveDataForTask(task.Id);

return RedirectToAction("Tasks", "Scraper");

}

//Здійснює обробку сторінки при завантаженні для преперегляду

public async Task<ActionResult> GetUrl(string encoding, string url)

{

try

{

var uri = new Uri(url);

string domain = string.Format("{0}://{1}", uri.Scheme, uri.Host);

HttpWebRequest webRequest = (HttpWebRequest)WebRequest.Create(url);

webRequest.Method = "GET";

webRequest.UserAgent = "Mozilla/5.0 (Windows NT 6.1; rv:14.0) Gecko/20100101 Firefox/14.0.1";

HtmlDocument htmlDoc = new HtmlDocument();

HttpWebResponse webResponse = (HttpWebResponse)webRequest.GetResponse();

String responseString;

using (Stream stream = webResponse.GetResponseStream())

{

Encoding enc;

switch (encoding)

{

case "ASCII":

enc = Encoding.ASCII;

break;

case "BEU":

enc = Encoding.BigEndianUnicode;

break;

case "Unicode":

enc = Encoding.Unicode;

break;

case "UTF32":

enc = Encoding.UTF32;

break;

case "UTF7":

enc = Encoding.UTF7;

break;

case "UTF8":

enc = Encoding.UTF8;

break;

default:

enc = Encoding.Default;

break;

}

StreamReader reader = new StreamReader(stream, enc);

responseString = reader.ReadToEnd();

}

htmlDoc.LoadHtml(responseString);

var el = htmlDoc.DocumentNode.SelectNodes("//script").Count;

foreach (HtmlNode script in htmlDoc.DocumentNode.SelectNodes("//script"))

{

HtmlAttributeCollection atrib = script.Attributes;

HtmlAttribute src = atrib.FirstOrDefault(x => x.Name == "src");

if (src != null)

{

string oldAttr = src.Value;

if (!(oldAttr.StartsWith("//") || oldAttr.StartsWith("http://") || oldAttr.StartsWith("https://")))

{

if (oldAttr.StartsWith("."))

{

oldAttr = oldAttr.TrimStart('.');

}

if (!oldAttr.StartsWith("/"))

{

oldAttr = "/" + oldAttr;

}

var newAttr = string.Format("{0}{1}", domain, oldAttr);

src.Value = newAttr;

}

}

}

int el2;

if (htmlDoc.DocumentNode.SelectNodes("//link") != null)

{

el2 = htmlDoc.DocumentNode.SelectNodes("//link").Count;

foreach (HtmlNode link in htmlDoc.DocumentNode.SelectNodes("//link"))

{

HtmlAttributeCollection atrib = link.Attributes;

HtmlAttribute href = atrib.FirstOrDefault(x => x.Name == "href");

if (href != null)

{

string oldAttr = href.Value;

if (!(oldAttr.StartsWith("//") || oldAttr.StartsWith("http://") || oldAttr.StartsWith("https://")))

{

if (oldAttr.StartsWith("."))

{

oldAttr = oldAttr.TrimStart('.');

}

if (!oldAttr.StartsWith("/"))

{

oldAttr = "/" + oldAttr;

}

var newAttr = string.Format("{0}{1}", domain, oldAttr);

href.Value = newAttr;

}

}

}

};

int el3;

if (htmlDoc.DocumentNode.SelectNodes("//img") != null)

{

el3 = htmlDoc.DocumentNode.SelectNodes("//img").Count;

foreach (HtmlNode img in htmlDoc.DocumentNode.SelectNodes("//img"))

{

HtmlAttributeCollection atrib = img.Attributes;

HtmlAttribute src = atrib.FirstOrDefault(x => x.Name == "src");

if (src != null)

{

string oldAttr = src.Value;

if (!(oldAttr.StartsWith("//") || oldAttr.StartsWith("http://") || oldAttr.StartsWith("https://")))

{

if (oldAttr.StartsWith("."))

{

oldAttr = oldAttr.TrimStart('.');

}

if (!oldAttr.StartsWith("/"))

{

oldAttr = "/" + oldAttr;

}

var newAttr = string.Format("{0}{1}", domain, oldAttr);

src.Value = newAttr;

}

}

}

}

int el4;

if (htmlDoc.DocumentNode.SelectNodes("//a") != null)

{

el4 = htmlDoc.DocumentNode.SelectNodes("//a").Count;

foreach (HtmlNode a in htmlDoc.DocumentNode.SelectNodes("//a"))

{

HtmlAttributeCollection atrib = a.Attributes;

HtmlAttribute href = atrib.FirstOrDefault(x => x.Name == "href");

if (href != null)

{

string oldAttr = href.Value;

if (oldAttr.StartsWith("http://") || oldAttr.StartsWith("https://"))

{

var newAttr = string.Format("/Scraper/GetUrl?encoding=" + encoding + "&url={0}", oldAttr);

href.Value = newAttr;

}

else

{

if (oldAttr.StartsWith("."))

{

oldAttr = oldAttr.TrimStart('.');

}

if (!oldAttr.StartsWith("/"))

{

oldAttr = "/" + oldAttr;

}

var newAttr = string.Format("/Scraper/GetUrl?encoding=" + encoding + "&url={0}{1}", domain, oldAttr);

href.Value = newAttr;

}

}

}

}

var res = htmlDoc.DocumentNode.OuterHtml;

return View("GetUrl", (object)res);

}

catch (Exception ex)

{

return View("GetUrlExeption", (object)ex.Message);

}

}

//Зберігає завдання

[HttpPost]

public JsonResult SaveTask(ScrapeTaskVM task)

{

\_user = UserManager.FindByName(User.Identity.Name);

var scrapeTask = new ScrapeTask();

scrapeTask = Mapper.Map<ScrapeTask>(task);

scrapeTask.Status = Status.Pending;

scrapeTask.UserId = \_user.Id;

scrapeTask.LinkSelectors = task.Links;

scrapeTask.Query = task.Query;

\_tasksRepository.Save(scrapeTask);

return Json(new { success = true }, JsonRequestBehavior.AllowGet);

}

protected override void OnActionExecuting(ActionExecutingContext filterContext)

{

\_user = UserManager.FindByName(User.Identity.Name);

if (\_user != null)

{

ViewBag.FirstName = \_user.FirstName;

ViewBag.LastName = \_user.LastName;

}

}

}}