

In this method, there is no criteria check before the file is written:

```
[AllowAnonymous]
[HttpPost]
[Route("Games/UploadGame")]
0 references
public async Task<IActionResult> UploadGame([FromForm] IFormFile gameFile)
{
    Console.WriteLine("DEBUG: UploadGame endpoint hit.");

    if (gameFile == null || gameFile.Length == 0)
    {
        Console.WriteLine("DEBUG: No file received.");
        Response.Headers.Add("Access-Control-Allow-Origin", "*");
        return BadRequest("No file uploaded.");
    }

    var uploadsPath = "/mnt/ebs-storage";
    Directory.CreateDirectory(uploadsPath);
    var filePath = Path.Combine(uploadsPath, Path.GetFileName(gameFile.FileName));

    try
    {
        using (var stream = new FileStream(filePath, FileMode.Create))
        {
            await gameFile.CopyToAsync(stream);
        }

        Console.WriteLine($"DEBUG: File saved to {filePath}");
        Response.Headers.Add("Access-Control-Allow-Origin", "*");
        return Json(new { message = "File uploaded successfully!" });
    }
}
```

It's saved immediately, even if it's invalid. We need to have the criteria check happen prior to saving. We could write the validation logic as a separate method or even a service class so that we can simply call something like "GameCriteriaService.Validate(gamefile)"