**[Authorize]**

Instead of applying on whole Regions controller

Apply it on Regions conrtoller methods for role base authorization

Like **[Authorize(Roles =”Reader”)]**

//Get All Regions

//Get: https://localhost:7050/api/regions

[HttpGet]

[Authorize(Roles ="Reader")]

public async Task<IActionResult> GetAll()

//Get Region By Id

//Get: https://localhost:7050/api/regions/{id}

[HttpGet]

[Route("{id:Guid}")]

[Authorize(Roles = "Reader")]

public async Task<IActionResult> GetById(Guid id)

//Post to create New Region

//POST: https://locathost:7050/api/regions

[HttpPost]

[ValidateModel]

[Authorize(Roles = "Writer")]

public async Task<IActionResult> Create([FromBody] AddRegionRequestDTO addregiondto)

//update region

//PUT: https://localhost:7050/api/regions/{id}

[HttpPut("{id:Guid}")]

[ValidateModel]

[Authorize(Roles ="Writer")]

public async Task<IActionResult> Update([FromRoute] Guid id, [FromBody] UpdateRegionRequestDTO updateRegiondto)

//Delete region

[HttpDelete]

[Route("{id:Guid}")]

[Authorize(Roles = "Writer")]

public async Task<IActionResult> Delete([FromRoute]Guid id)

**Assing multiple roles**

//Delete region

[HttpDelete]

[Route("{id:Guid}")]

[Authorize(Roles = "Writer ,Reader")]

public async Task<IActionResult> Delete([FromRoute]Guid id)

or

//Delete region

[HttpDelete]

[Route("{id:Guid}")]

[Authorize]

public async Task<IActionResult> Delete([FromRoute]Guid id)

**Note create a new user have writer role to check the functionality**