



## TABLE OF CONTENTS

<b>1</b>	<b>PROJECT CONFIGURATION .....</b>	<b>1</b>
1.1	Creating a New Project .....	1
1.2	launchSettings.json File Configuration .....	2
1.3	Program.cs Class Explanations.....	4
1.4	Extension Methods and CORS Configuration.....	7
1.5	IIS Configuration .....	9
1.6	Additional Code in the Program Class.....	11
1.7	Environment-Based Settings .....	12
1.8	ASP.NET Core Middleware .....	13
1.8.1	Creating a First Middleware Component.....	16
1.8.2	Working with the Use Method.....	18
1.8.3	Using the Map and MapWhen Methods .....	20
1.8.4	Using MapWhen Method .....	21
<b>2</b>	<b>CONFIGURING A LOGGING SERVICE .....</b>	<b>23</b>
2.1	Creating the Required Projects .....	23
2.2	Creating the ILoggerManager Interface and Installing NLog .....	24
2.3	Implementing the Interface and Nlog.Config File.....	26
2.4	Configuring Logger Service for Logging Messages .....	27
2.5	DI, IoC, and Logger Service Testing .....	29
<b>3</b>	<b>ONION ARCHITECTURE IMPLEMENTATION .....</b>	<b>31</b>
3.1	About Onion Architecture .....	32

---



3.1.1	Advantages of the Onion Architecture .....	33
3.1.2	Flow of Dependencies.....	33
<b>3.2</b>	<b>Creating Models .....</b>	<b>34</b>
<b>3.3</b>	<b>Context Class and the Database Connection.....</b>	<b>36</b>
<b>3.4</b>	<b>Migration and Initial Data Seed.....</b>	<b>39</b>
<b>3.5</b>	<b>Repository Pattern Logic .....</b>	<b>42</b>
<b>3.6</b>	<b>Repository User Interfaces and Classes .....</b>	<b>44</b>
<b>3.7</b>	<b>Creating a Repository Manager .....</b>	<b>45</b>
<b>3.8</b>	<b>Adding a Service Layer.....</b>	<b>47</b>
<b>3.9</b>	<b>Registering RepositoryContext at a Runtime.....</b>	<b>50</b>
<b>4</b>	<b>HANDLING GET REQUESTS .....</b>	<b>52</b>
<b>4.1</b>	<b>Controllers and Routing in WEB API.....</b>	<b>52</b>
<b>4.2</b>	<b>Naming Our Resources.....</b>	<b>57</b>
<b>4.3</b>	<b>Getting All Companies From the Database .....</b>	<b>57</b>
<b>4.4</b>	<b>Testing the Result with Postman.....</b>	<b>61</b>
<b>4.5</b>	<b>DTO Classes vs. Entity Model Classes .....</b>	<b>62</b>
<b>4.6</b>	<b>Using AutoMapper in ASP.NET Core.....</b>	<b>65</b>
<b>5</b>	<b>GLOBAL ERROR HANDLING .....</b>	<b>70</b>
<b>5.1</b>	<b>Handling Errors Globally with the Built-In Middleware.....</b>	<b>70</b>
<b>5.2</b>	<b>Program Class Modification .....</b>	<b>72</b>
<b>5.3</b>	<b>Testing the Result .....</b>	<b>73</b>

---



<b>6</b>	<b>GETTING ADDITIONAL RESOURCES .....</b>	<b>75</b>
6.1	Getting a Single Resource From the Database.....	75
6.1.1	Handling Invalid Requests in a Service Layer .....	77
6.2	Parent/Child Relationships in Web API .....	80
6.3	Getting a Single Employee for Company.....	83
<b>7</b>	<b>CONTENT NEGOTIATION .....</b>	<b>87</b>
7.1	What Do We Get Out of the Box?.....	87
7.2	Changing the Default Configuration of Our Project .....	88
7.3	Testing Content Negotiation.....	89
7.4	Restricting Media Types .....	91
7.5	More About Formatters .....	92
7.6	Implementing a Custom Formatter .....	93
<b>8</b>	<b>METHOD SAFETY AND METHOD IDEMPOTENCY .....</b>	<b>96</b>
<b>9</b>	<b>CREATING RESOURCES .....</b>	<b>98</b>
9.1	Handling POST Requests .....	98
9.2	Code Explanation .....	101
9.2.1	Validation from the ApiController Attribute.....	102
9.3	Creating a Child Resource .....	105
9.4	Creating Children Resources Together with a Parent .....	108
9.5	Creating a Collection of Resources .....	109
9.6	Model Binding in API.....	115

---



<b>10</b>	<b>WORKING WITH DELETE REQUESTS</b> .....	<b>119</b>
10.1	Deleting a Parent Resource with its Children .....	121
<b>11</b>	<b>WORKING WITH PUT REQUESTS</b> .....	<b>123</b>
11.1	Updating Employee .....	123
11.1.1	About the Update Method from the RepositoryBase Class .....	127
11.2	Inserting Resources while Updating One .....	127
<b>12</b>	<b>WORKING WITH PATCH REQUESTS</b> .....	<b>130</b>
12.1	Applying PATCH to the Employee Entity .....	131
<b>13</b>	<b>VALIDATION</b> .....	<b>138</b>
13.1	ModelState, Rerun Validation, and Built-in Attributes .....	138
13.1.1	Rerun Validation .....	139
13.1.2	Built-in Attributes .....	140
13.2	Custom Attributes and IValidatableObject .....	141
13.3	Validation while Creating Resource .....	143
13.3.1	Validating Int Type .....	146
13.4	Validation for PUT Requests .....	147
13.5	Validation for PATCH Requests.....	149
<b>14</b>	<b>ASYNCHRONOUS CODE</b> .....	<b>154</b>
14.1	What is Asynchronous Programming? .....	154
14.2	Async, Await Keywords and Return Types.....	156
14.2.1	Return Types of the Asynchronous Methods .....	158
14.2.2	The IRepositoryBase Interface and the RepositoryBase Class Explanation	159

---



<b>14.3</b>	<b>Modifying the ICompanyRepository Interface and the CompanyRepository Class .....</b>	<b>159</b>
<b>14.4</b>	<b>IRepositoryManager and RepositoryManager Changes .....</b>	<b>160</b>
<b>14.5</b>	<b>Updating the Service layer .....</b>	<b>161</b>
<b>14.6</b>	<b>Controller Modification .....</b>	<b>163</b>
<b>14.7</b>	<b>Continuation in Asynchronous Programming .....</b>	<b>166</b>
<b>14.8</b>	<b>Common Pitfalls .....</b>	<b>167</b>
<b>15</b>	<b>ACTION FILTERS .....</b>	<b>169</b>
<b>15.1</b>	<b>Action Filters Implementation .....</b>	<b>169</b>
<b>15.2</b>	<b>The Scope of Action Filters .....</b>	<b>170</b>
<b>15.3</b>	<b>Order of Invocation .....</b>	<b>171</b>
<b>15.4</b>	<b>Improving the Code with Action Filters .....</b>	<b>173</b>
<b>15.5</b>	<b>Validation with Action Filters .....</b>	<b>173</b>
<b>15.6</b>	<b>Refactoring the Service Layer .....</b>	<b>176</b>
<b>16</b>	<b>PAGING .....</b>	<b>181</b>
<b>16.1</b>	<b>What is Paging? .....</b>	<b>181</b>
<b>16.2</b>	<b>Paging Implementation .....</b>	<b>182</b>
<b>16.3</b>	<b>Concrete Query .....</b>	<b>185</b>
<b>16.4</b>	<b>Improving the Solution .....</b>	<b>187</b>
16.4.1	Additional Advice .....	190
<b>17</b>	<b>FILTERING .....</b>	<b>192</b>
<b>17.1</b>	<b>What is Filtering? .....</b>	<b>192</b>

---



<b>17.2</b>	<b>How is Filtering Different from Searching?</b> .....	<b>193</b>
<b>17.3</b>	<b>How to Implement Filtering in ASP.NET Core Web API</b> .....	<b>194</b>
<b>17.4</b>	<b>Sending and Testing a Query</b> .....	<b>196</b>
<b>18</b>	<b>SEARCHING</b> .....	<b>199</b>
<b>18.1</b>	<b>What is Searching?</b> .....	<b>199</b>
<b>18.2</b>	<b>Implementing Searching in Our Application</b> .....	<b>199</b>
<b>18.3</b>	<b>Testing Our Implementation</b> .....	<b>201</b>
<b>19</b>	<b>SORTING</b> .....	<b>204</b>
<b>19.1</b>	<b>What is Sorting?</b> .....	<b>204</b>
<b>19.2</b>	<b>How to Implement Sorting in ASP.NET Core Web API</b> .....	<b>206</b>
<b>19.3</b>	<b>Implementation – Step by Step</b> .....	<b>208</b>
<b>19.4</b>	<b>Testing Our Implementation</b> .....	<b>210</b>
<b>19.5</b>	<b>Improving the Sorting Functionality</b> .....	<b>211</b>
<b>20</b>	<b>DATA SHAPING</b> .....	<b>213</b>
<b>20.1</b>	<b>What is Data Shaping?</b> .....	<b>213</b>
<b>20.2</b>	<b>How to Implement Data Shaping</b> .....	<b>214</b>
<b>20.3</b>	<b>Step-by-Step Implementation</b> .....	<b>216</b>
<b>20.4</b>	<b>Resolving XML Serialization Problems</b> .....	<b>221</b>
<b>21</b>	<b>SUPPORTING HATEOAS</b> .....	<b>224</b>
<b>21.1</b>	<b>What is HATEOAS and Why is it so Important?</b> .....	<b>224</b>
21.1.1	<b>Typical Response with HATEOAS Implemented</b> .....	<b>225</b>

---



21.1.2	What is a Link?.....	225
21.1.3	Pros/Cons of Implementing HATEOAS .....	226
<b>21.2</b>	<b>Adding Links in the Project .....</b>	<b>227</b>
<b>21.3</b>	<b>Additional Project Changes .....</b>	<b>228</b>
<b>21.4</b>	<b>Adding Custom Media Types.....</b>	<b>230</b>
21.4.1	Registering Custom Media Types .....	230
21.4.2	Implementing a Media Type Validation Filter .....	231
<b>21.5</b>	<b>Implementing HATEOAS.....</b>	<b>233</b>
<b>22</b>	<b>WORKING WITH OPTIONS AND HEAD REQUESTS.....</b>	<b>241</b>
<b>22.1</b>	<b>OPTIONS HTTP Request .....</b>	<b>241</b>
<b>22.2</b>	<b>OPTIONS Implementation .....</b>	<b>241</b>
<b>22.3</b>	<b>Head HTTP Request.....</b>	<b>243</b>
<b>22.4</b>	<b>HEAD Implementation.....</b>	<b>243</b>
<b>23</b>	<b>ROOT DOCUMENT .....</b>	<b>245</b>
<b>23.1</b>	<b>Root Document Implementation .....</b>	<b>245</b>
<b>24</b>	<b>VERSIONING APIS .....</b>	<b>250</b>
<b>24.1</b>	<b>Required Package Installation and Configuration .....</b>	<b>250</b>
<b>24.2</b>	<b>Versioning Examples .....</b>	<b>251</b>
24.2.1	Using Query String .....	253
24.2.2	Using URL Versioning .....	254
24.2.3	HTTP Header Versioning .....	255
24.2.4	Deprecating Versions .....	256
24.2.5	Using Conventions.....	257
<b>25</b>	<b>CACHING.....</b>	<b>258</b>

---



<b>25.1</b>	<b>About Caching</b> .....	<b>258</b>
25.1.1	Cache Types .....	258
25.1.2	Response Cache Attribute .....	259
<b>25.2</b>	<b>Adding Cache Headers</b> .....	<b>259</b>
<b>25.3</b>	<b>Adding Cache-Store</b> .....	<b>261</b>
<b>25.4</b>	<b>Expiration Model</b> .....	<b>264</b>
<b>25.5</b>	<b>Validation Model</b> .....	<b>266</b>
<b>25.6</b>	<b>Supporting Validation</b> .....	<b>267</b>
25.6.1	Configuration .....	268
<b>25.7</b>	<b>Using ETag and Validation</b> .....	<b>270</b>
<b>26</b>	<b>RATE LIMITING AND THROTTLING</b> .....	<b>273</b>
<b>26.1</b>	<b>Implementing Rate Limiting</b> .....	<b>273</b>
<b>27</b>	<b>JWT, IDENTITY, AND REFRESH TOKEN</b> .....	<b>277</b>
<b>27.1</b>	<b>Implementing Identity in ASP.NET Core Project</b> .....	<b>277</b>
<b>27.2</b>	<b>Creating Tables and Inserting Roles</b> .....	<b>280</b>
<b>27.3</b>	<b>User Creation</b> .....	<b>281</b>
<b>27.4</b>	<b>Big Picture</b> .....	<b>287</b>
<b>27.5</b>	<b>About JWT</b> .....	<b>287</b>
<b>27.6</b>	<b>JWT Configuration</b> .....	<b>289</b>
<b>27.7</b>	<b>Protecting Endpoints</b> .....	<b>291</b>
<b>27.8</b>	<b>Implementing Authentication</b> .....	<b>292</b>
<b>27.9</b>	<b>Role-Based Authorization</b> .....	<b>297</b>

---





<b>28</b>	<b>REFRESH TOKEN.....</b>	<b>300</b>
28.1	Why Do We Need a Refresh Token .....	302
28.2	Refresh Token Implementation .....	303
28.3	Token Controller Implementation .....	307
<b>29</b>	<b>BINDING CONFIGURATION AND OPTIONS PATTERN.....</b>	<b>311</b>
29.1	Binding Configuration .....	312
29.2	Options Pattern .....	314
29.2.1	Using IOptions .....	315
29.2.2	IOptionsSnapshot and IOptionsMonitor.....	317
<b>30</b>	<b>DOCUMENTING API WITH SWAGGER .....</b>	<b>320</b>
30.1	About Swagger.....	320
30.2	Swagger Integration Into Our Project.....	321
30.3	Adding Authorization Support .....	325
30.4	Extending Swagger Configuration .....	328
<b>31</b>	<b>DEPLOYMENT TO IIS .....</b>	<b>332</b>
31.1	Creating Publish Files.....	332
31.2	Windows Server Hosting Bundle .....	334
31.3	Installing IIS.....	335
31.4	Configuring Environment File .....	338
31.5	Testing Deployed Application .....	339
<b>32</b>	<b>BONUS 1 - RESPONSE PERFORMANCE IMPROVEMENTS.....</b>	<b>343</b>

---



<b>32.1</b>	<b>Adding Response Classes to the Project</b>	<b>343</b>
<b>32.2</b>	<b>Service Layer Modification</b>	<b>345</b>
<b>32.3</b>	<b>Controller Modification</b>	<b>347</b>
<b>32.4</b>	<b>Testing the API Response Flow</b>	<b>349</b>
<b>33</b>	<b>BONUS 2 - INTRODUCTION TO CQRS AND MEDIATR WITH ASP.NET CORE WEB API</b>	<b>352</b>
<b>33.1</b>	<b>About CQRS and Mediator Pattern</b>	<b>352</b>
33.1.1	CQRS	352
33.1.2	Advantages and Disadvantages of CQRS	354
33.1.3	Mediator Pattern	355
<b>33.2</b>	<b>How MediatR facilitates CQRS and Mediator Patterns</b>	<b>356</b>
<b>33.3</b>	<b>Adding Application Project and Initial Configuration</b>	<b>356</b>
<b>33.4</b>	<b>Requests with MediatR</b>	<b>359</b>
<b>33.5</b>	<b>Commands with MediatR</b>	<b>365</b>
33.5.1	Update Command	367
33.5.2	Delete Command	369
<b>33.6</b>	<b>MediatR Notifications</b>	<b>370</b>
<b>33.7</b>	<b>MediatR Behaviors</b>	<b>373</b>
33.7.1	Adding Fluent Validation	374
33.7.2	Creating Decorators with MediatR PipelineBehavior	375
33.7.3	Validating null Object	379

---