Web API coding guidelines

# Web API Action Return Types:

|  |  |
| --- | --- |
| IActionResult 🡪 ActionResult 🡪 ObjectResult | HTTP Code Status |
| Microsoft.AspNetCore.Mvc.AcceptedAtActionResult | 202 |
| Microsoft.AspNetCore.Mvc.AcceptedAtRouteResult | 202 |
| Microsoft.AspNetCore.Mvc.AcceptedResult | 202 |
| Microsoft.AspNetCore.Mvc.BadRequestObjectResult | 400 |
| Microsoft.AspNetCore.Mvc.ConflictObjectResult | 409 |
| Microsoft.AspNetCore.Mvc.CreatedAtActionResult | 201 |
| Microsoft.AspNetCore.Mvc.CreatedAtRouteResult | 201 |
| Microsoft.AspNetCore.Mvc.CreatedResult | 201 |
| Microsoft.AspNetCore.Mvc.NotFoundObjectResult | 404 |
| Microsoft.AspNetCore.Mvc.OkObjectResult | 200 |
| Microsoft.AspNetCore.Mvc.UnauthorizedObjectResult | 401 |
| Microsoft.AspNetCore.Mvc.UnprocessableEntityObjectResult | 422 |
| System.Web.Http.BadRequestErrorMessageResult | 400 |
| System.Web.Http.ExceptionResult | 500 |
| System.Web.Http.InvalidModelStateResult | 400 |
| System.Web.Http.NegotiatedContentResult<T> | Content negotiation means client (request) sends the response type negotiation through header to get the proper response type back. "The process of selecting the best representation for a given response when there are multiple representations available." |
| System.Web.Http.ResponseMessageResult | General Response, User define status |

# RESTful API HTTP verbs

* Safe: do not modify resource
* Idempotent: called many times without different outcome

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| HTTP Method | CRUD operation | Entire Collection (e.g. /users) | Specific Item (e.g. /users/{id}) | Safe | Idempotent |
| GET | Read | 200 (OK), list of entities. Use pagination, sorting and filtering to navigate big lists. | 200 (OK), single entity.  404 (Not Found), if ID not found or invalid. | Yes | Yes |
| POST | Create | 201 (Created), Response contains response similar to GET /user/{id} containing new ID. | not applicable | No | No |
| PATCH | Update | [Batch API](https://doc.oroinc.com/api/batch-api/#web-services-api-batch-api) | 200 (OK) or 204 (No Content).  404 (Not Found), if ID not found or invalid. | No | Yes |
| DELETE | Delete | 204 (No Content).  400(Bad Request) if no filter is specified. | 204 (No Content).  404 (Not Found), if ID not found or invalid. | Yes | Yes |
| PUT | Create/Replace | not implemented, same as POST | not implemented, same as POST | Yes | No |

* Additional HTTP Methods

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| HTTP Method | CRUD operation | Entire Collection (e.g. /users) | Specific Item (e.g. /users/{id}) | Safe | Idempotent |
| HEAD |  | 200 (OK), returns metadata about the resource | 200 (OK), Open API | Yes | Yes |
| TRACE |  | 200 (OK), echo back for diagnosis | 200 (OK), Echo | Yes | Yes |
| OPTIONS |  | 200(OK), return list of HTTP methods supported and allowed | 200 (OK), e.g.  Allow: GET, POST, DELETE, HEAD | Yes | Yes |
| CONNECT |  | 200(OK), connect to Proxy | 200 (OK), rarely used | No | No |

# ASP.NET Core Filters

|  |  |  |
| --- | --- | --- |
| Filter Pipeline | Filter Types | Filter types interact in Filter Pipeline |
| The request is processed through Other Middleware, Routing Middleware, Action Selection, and the Action Invocation Pipeline. The request processing continues back through Action Selection, Routing Middleware, and various Other Middleware before becoming a response sent to the client. | Authorization filters:   * Run first. * Determine whether the user is authorized for the request. * Short-circuit the pipeline if the request is not authorized.   Resource filters:   * Run after authorization. * OnResourceExecuting runs code before the rest of the filter pipeline. For example, OnResourceExecuting runs code before model binding. * OnResourceExecuted runs code after the rest of the pipeline has completed.   Action filters:   * Run immediately before and after an action method is called. * Can change the arguments passed into an action. * Can change the result returned from the action. * Are not supported in Razor Pages.   Endpoint filters:   * Run immediately before and after an action method is called. * Can change the arguments passed into an action. * Can change the result returned from the action. * Are not supported in Razor Pages. * Can be invoked on both actions and route handler-based endpoints. * Exception filters apply global policies to unhandled exceptions that occur before the response body has been written to.   Result filters:   * Run immediately before and after the execution of action results. * Run only when the action method executes successfully. * Are useful for logic that must surround view or formatter execution. | The request is processed through Authorization Filters, Resource Filters, Model Binding, Action Filters, Action Execution and Action Result Conversion, Exception Filters, Result Filters, and Result Execution. On the way out, the request is only processed by Result Filters and Resource Filters before becoming a response sent to the client. |