Partially Updating a Resource

* Full updates with PUT aren't always advised, it adds the overhead.
* In fact, if you would look at OData, which is a standard that's essentially a set of best practices for creating RESTful APIs, we'd notice that that standard states PATCH should be preferred over PUT.
* PATCH = PARTIAL UPDATES
* We need a way to pass a change set to a resource using the HTTP PATCH method.
* Luckily there's a standard for this, the JSON Patch standard. This defines a JSON document structure for expressing a sequence of operations to apply to a JSON document. You can look at that structure as a change set, a set of operations that'll be applied to the resource a PATCH request with that JsonPatchDocument in the body is sent to.
* The application/json-patch+json media type is used to identify such PATCH documents.

Timeline

Description automatically generated

* Imagine the below is a PATCH request to a specific course for an author. It starts with straight brackets, signifying an array. The array is a list of operations that have to be applied to the resource to our course. We see two operations in the example onscreen. The first one is a replace operation, signified by op. Path signifies the path to the property, title, in this case. These are property names of the resource, the DTO, and not of whatever lies beneath that layer. Value signifies a new value that should be given to the title property, new title.
* The second operation removes the description of a course. The operation is set to remove and the path is thus description. There's no value. The properties value will be removed.
* In some dynamic systems, the property itself will be removed from the resource, but when working with DTO classes, it should be set to its default value.
* Once this request is received, the API will apply it to the resource. Each operation is applied after the other one, and a request is only successful when all operations can be applied.

Graphical user interface, text, application

Description automatically generated

* There's six different operations possible.
* The Add operation will add a property at a path location with a specific value, passed through via value. If you choose an unexisting path, the property should be added to the resource. But something like that is only possible when working with dynamic resources, often in CRM-like systems. It's not applicable in our case, as we're currently working with statically-typed classes.
* The remove operation will remove a property, or, in non-dynamic cases, set it to its default value. It only has one property that has to be set, path.
* Replace replaces the value at the specified path with the provided value. It's functionally the same as a remove operation, followed by an add operation.

Chart

Description automatically generated with low confidence

* Copy will take the value of the from property and copy it over to the path property. It's thus an add operation at the path location with the value specified in the from member.
* Move will copy over the value at the from property to the path property and remove the value at the from property. This operation is functionally identical to a remove operation on the from location, followed by an add at the path location with a removed value.
* And lastly, test tests that a value at the target location is equal to a specified value.

Chart

Description automatically generated

* These cases aren't limited to simple properties on a resource; we can manipulate array properties, we can access nested properties, we can even add a list of items to an array. So, path doesn't have to be a simple property and value doesn't have to be one string value; it might as well be an array.
* From these, it follows that PATCH is neither safe nor idempotent. It changes resource representations, and, as it can add to an array, sending it multiple times will have different outcomes. So, it's really powerful. The most important thing to remember is that a JSON PATCH document is essentially a list of operations that have to be applied to the resource, allowing for partial updates.