Fexa - API Filter Breakdown

1. Construct each filter. In an example case we need two filters for vendor and workorders, which would be JSON objects { "property": "workorders.id", "value": 1 }, and { "property": "vendors.id", "value": 25 }
2. Our documentation is not completely updated from a filter “property” standpoint. The property field is generally <object>.<field_name>. For example, to filter invoices by the workorder id, the "property" field is "workorders.id". Apologies that this isn't super intuitive in our documentation, we are planning to update our documentation in the future. If what you are trying doesn't seem to work, please put in a help desk ticket and we can provide you with the appropriate field name.
3. We offer an additional "operator" field in each JSON object and here is the breakdown for the operators that we support :
    - No "operator" key (default action)
        - { "property": "some_field", "value": 1 }
        - Creates a filter structure of "WHERE some_field = 1"
    - "in"
        - { "operator": "in", "property": "some_field", "value": [1, 2] }
        - Creates a filter structure of "WHERE some_field IN (1, 2)"
    - "not in"
        - { "operator": "not in", "property": "some_field", "value": [1, 2] }
        - Creates a filter structure of "WHERE some_field NOT IN (1, 2)"
    - "between"
        - { "operator": "between", "property": "some_date_field", "value": ["date1", "date2] }
        - Creates a filter structure of "WHERE some_date_field BETWEEN date1 AND date2"
4. Package these up into an array. For our example it would be [{"property":"workorders.id","value":1},{"property": "vendors.id","value":25}] (I've

removed the spaces for compactness, I don't think it's actually strictly required though)

5. Encode that using standard URL encoding: %5B%7B%22property%22%3A%22workorders.id%22%2C%22value%22%3A1%7D%2C%7B%22property%22%3A%20%22vendors.id%22%2C%22value%22%3A25%7D%5D
6. The value from step 3 is the value that should be used for the filter parameter in the request.
   (Note: I'm calling the filters JSON objects since that's how they're formatted, but they're really just strings. Building these out with a JSON library is usually easier, but you can also just use string manipulation)
7. Here are two examples of API calls to the client invoices endpoint :
    - Client invoices connected to one specific workorder
        - https:///api/ev1/client_invoices?start=0&limit=5&filters=%5B%7B%22property%22%3A%22workorders.id%22%2C%22value%22%3A116%7D%5D
        - https:///api/ev1/client_invoices?start=0&limit=5&filters=[{"property":"workorders.id","value":116}
    - Client invoices connected to a group of workorders
        - https:///api/ev1/client_invoices?start=0&limit=5&filters=%5B%7B%22property%22%3A%22workorders.id%22%2C%22operator%22%3A%22in%22%2C%22value%22%3A%5B116%2C117%5D%7D%5D
        - https:///api/ev1/client_invoices?start=0&limit=5&filters=[{"property":"workorders.id","operator":"in","value":[116,117]}]