**API Helper for testing endpoints**

**1. CSV Upload Test Data**

Save this as detections.csv and upload it to POST /api/kpi/upload.

**Csv File**

id,class,x,y,timestamp,speed,heading,vest,zone,eventType

H001,human,12.5,8.3,2023-10-01T08:15:00,1.2,90,true,ZoneA,close\_call

V001,vehicle,15.0,9.0,2023-10-01T08:16:00,3.5,270,true,ZoneA,overspeed

H002,human,13.0,8.5,2023-10-01T08:17:00,0.8,85,false,ZoneA,vest\_violation

AGV01,agv,20.0,5.0,2023-10-01T08:18:00,2.0,180,true,ZoneB,close\_call

PT01,pallet\_truck,18.0,6.5,2023-10-01T08:19:00,2.8,270,true,ZoneB,overspeed

**📊 2. KPI Computation Test Payload**

Use this JSON for POST /api/kpi/compute:

Input Json 1

{

"metric": "count",

"filters": {

"time\_range": {

"start": "2023-10-01T08:00:00",

"end": "2023-10-01T09:00:00"

},

"class": ["human", "vehicle"],

"zone": ["ZoneA", "ZoneB"],

"speed": {

"min": 1.0,

"max": 5.0

},

"vest": 1,

"heading": {

"min": 80,

"max": 300

}

},

"group\_by": ["timestamp\_bucket", "class"],

"bucket\_interval\_minutes": 5

}

**💾 3. Save Preset Test Payload**

Use this JSON for POST /api/kpi/presets?name=TestPrest1

json

{

"metric": "count",

"filters": {

"time\_range": {

"start": "2023-10-01T08:00:00",

"end": "2023-10-01T09:00:00"

},

"class": ["human", "vehicle"],

"zone": ["ZoneA", "ZoneB"],

"speed": {

"min": 1.0,

"max": 5.0

},

"vest": 1,

"heading": {

"min": 80,

"max": 300

}

},

"group\_by": ["timestamp\_bucket", "class"],

"bucket\_interval\_minutes": 5

}

Expected response:

json

{ message = "Preset saved successfully." }

**📂 4. Retrieve Presets Test Response**

Expected output from GET /api/kpi/presets:

**Response body**

Download

[

{

"id": 1,

"name": "TestPreset",

"jsonPayload": "{\"Name\":\"\",\"metric\":\"count\",\"filters\":{\"time\_range\":{\"start\":\"2023-10-01T08:00:00\",\"end\":\"2023-10-01T09:00:00\"},\"class\":[\"human\"],\"zone\":[\"ZoneA\"],\"speed\":{\"min\":1,\"max\":5},\"vest\":1,\"heading\":{\"min\":0,\"max\":360}},\"group\_by\":[\"class\",\"zone\"],\"bucket\_interval\_minutes\":5,\"chart\_type\":\"bar\"}",

"createdAt": "2025-10-25T00:33:50.7275987"

},

{

"id": 2,

"name": "Testing",

"jsonPayload": "{\"Name\":\"\",\"metric\":\"count\",\"filters\":{\"time\_range\":{\"start\":\"2023-10-01T08:00:00\",\"end\":\"2023-10-01T09:00:00\"},\"class\":[\"human\"],\"zone\":[\"ZoneA\"],\"speed\":{\"min\":1,\"max\":5},\"vest\":1,\"heading\":{\"min\":0,\"max\":360}},\"group\_by\":[\"class\",\"zone\"],\"bucket\_interval\_minutes\":5,\"chart\_type\":\"bar\"}",

"createdAt": "2025-10-25T00:39:54.6960925"

},

{

"id": 3,

"name": "TestKPI",

"jsonPayload": "{\"Name\":\"Custom KPI\",\"metric\":\"count\",\"filters\":{\"time\_range\":{\"start\":\"2023-10-01T08:00:00\",\"end\":\"2023-10-01T09:00:00\"},\"class\":[],\"zone\":[],\"speed\":{\"min\":0,\"max\":5.6},\"vest\":null,\"heading\":{\"min\":0,\"max\":205}},\"group\_by\":[],\"bucket\_interval\_minutes\":5,\"chart\_type\":\"bar\"}",

"createdAt": "2025-10-25T00:53:43.8823198"

},

{

"id": 4,

"name": "RamTest1",

"jsonPayload": "{\"Name\":\"Custom KPI\",\"metric\":\"count\",\"filters\":{\"time\_range\":{\"start\":\"2023-10-01T08:00:00\",\"end\":\"2023-10-01T09:00:00\"},\"class\":[],\"zone\":[],\"speed\":{\"min\":1.1,\"max\":8.3},\"vest\":null,\"heading\":{\"min\":65,\"max\":217}},\"group\_by\":[\"class\"],\"bucket\_interval\_minutes\":5,\"chart\_type\":\"bar\"}",

"createdAt": "2025-10-25T00:55:24.7590276"

}

]