***PASANDO SÓLO $idBusinessCentral DE PADRE COMO PARÁMETRO***

***REPOSITORY***

public function getSubproductWithElaborados($idBusinessCentral)

{

try {

$sql = "

SELECT

sbc.idBusinessCentral,

(

SELECT s\_main.idTpv

FROM subproducts s\_main

JOIN subproducts\_business\_central sbc\_main ON sbc\_main.idNumier = s\_main.id

WHERE sbc\_main.idBusinessCentral = :idBusinessCentral

LIMIT 1

) AS idTpv,

GROUP\_CONCAT(sbc.idNumier) AS elaborados

FROM subproducts\_business\_central sbc

WHERE sbc.idBusinessCentral = :idBusinessCentral

GROUP BY sbc.idBusinessCentral

";

$stmt = $this->db->prepare($sql);

$stmt->execute(['idBusinessCentral' => $idBusinessCentral]);

$row = $stmt->fetch(\PDO::FETCH\_ASSOC);

if (!$row) {

return null;

}

return [

"idBusinessCentral" => $row["idBusinessCentral"],

"idTpv" => $row["idTpv"],

"elaborados" => array\_map('intval', explode(',', $row["elaborados"]))

];

} catch (\Exception $ex) {

$this->logger->error("❌ Error al obtener subproducto con elaborados: " . $ex->getMessage());

return null;

}

}

***CONTROLLER***

public function getSubproductWithElaborados(Request $request, Response $response)

{

$params = $request->getParsedBody();

$idBusinessCentral = $params["idBusinessCentral"] ?? null;

if (!$idBusinessCentral) {

return $response->withJson([

"response" => false,

"message" => "Falta el parámetro idBusinessCentral"

], 400);

}

$resultado = $this->subproductRepository->getSubproductWithElaborados($idBusinessCentral);

if (!$resultado) {

return $response->withJson([

"response" => false,

"message" => "No se encontraron datos para el idBusinessCentral proporcionado"

], 404);

}

return $response->withJson([

"response" => true,

"result" => $resultado

]);

}

***PASANDO $idBusinessCentral y elaborados (de idsNumier) COMO PARÁMETRO AL CONTROLADOR***

***REPOSITORY***

public function getSubproductBaseData($idBusinessCentral)

{

try {

$sql = "

SELECT s\_main.idTpv

FROM subproducts s\_main

JOIN subproducts\_business\_central sbc\_main ON sbc\_main.idNumier = s\_main.id

WHERE sbc\_main.idBusinessCentral = :idBusinessCentral

LIMIT 1

";

$stmt = $this->db->prepare($sql);

$stmt->execute(['idBusinessCentral' => $idBusinessCentral]);

$row = $stmt->fetch(\PDO::FETCH\_ASSOC);

if (!$row) {

return null;

}

return [

"idTpv" => $row["idTpv"]

];

} catch (\Exception $ex) {

$this->logger->error("❌ Error al obtener datos base del subproducto: " . $ex->getMessage());

return null;

}

}

***CONTROLLER***

public function getSubproductWithCustomElaborados(Request $request, Response $response)

{

$params = $request->getParsedBody();

$idBusinessCentral = $params["idBusinessCentral"] ?? null;

$elaborados = $params["elaborados"] ?? [];

if (!$idBusinessCentral || !is\_array($elaborados)) {

return $response->withJson([

"response" => false,

"message" => "Parámetros inválidos"

], 400);

}

$baseData = $this->subproductRepository->getSubproductBaseData($idBusinessCentral);

if (!$baseData) {

return $response->withJson([

"response" => false,

"message" => "No se encontró el subproducto principal"

], 404);

}

$result = [

"idBusinessCentral" => $idBusinessCentral,

"idTpv" => $baseData["idTpv"],

"elaborados" => array\_map('intval', $elaborados)

];

return $response->withJson([

"response" => true,

"result" => $result

]);

}