GET /v2/travel_requests

Travel Requests List

این اندپوینت لیست درخواست‌های سفر (مانند پرواز، هتل، قطار و ...) را با قابلیت‌های پیشرفته فیلترینگ و **غنی‌سازی (Data Enrichment)** بازمی‌گرداند.
نکته متمایز این کنترلر، منطق **Scoping** (تعیین سطح دسترسی بر اساس گروه کاربری) و همچنین اتصال به چندین جدول (Operators, Customers, Colleagues, Cities, Factors) و دیتابیس **Redis** برای ساخت یک آبجکت پاسخ کامل است.

List Requests (Index)

URL: `/v2/travel_requests`
Method: GET
Controller: TravelRequestsController@index

Query Parameters

Parameter	Type	Description
branch	integer	**(الزامی)** شناسه شعبه.
group	string	تعیین کننده سطح دسترسی (`requester_id`): • `base` / `b2e`: دسترسی ادمین به درخواست‌های یک کاربر خاص (نیاز به ارسال `requester_id`). • `colleague` / `b2b`: لیست درخواست‌های خود اپراتور B2B. • `agent` / `b2c`: لیست درخواست‌های خود اپراتور B2C.
requester_id	integer	در صورتی که `group` برابر با `base` باشد، برای فیلتر کردن درخواست‌های یک کاربر خاص استفاده می‌شود.

Parameter	Type	Description
operator_id	integer	فیلتر بر اساس اپراتور هندلکننده درخواست.
method	string	فیلتر نوع اصلی خدمت (مثلاً `flight`, `hotel`).
submethod	string	فیلتر زیرمجموعه خدمت.
status	string	فیلتر وضعیت درخواست.
payment_status	string	فیلتر وضعیت پرداخت.
paginate[length]	integer	تعداد آیتم در صفحه (پیشفرض: ۳۰).
paginate[start]	integer	آفست شروع (پیشفرض: ۰). شماره صفحه به صورت دینامیک محاسبه میشود.

Logic Details

1. **منطق صفحه‌بندی (Pagination Calculation):**
   سیستم از متد استاندارد `page` لاراول استفاده نمیکند، بلکه شماره صفحه را بر اساس `start` و `length` محاسبه میکند:
   `Current Page = (start + length) / length`
   اگر `start=0` باشد، صفحه ۱ در نظر گرفته میشود.
2. **منطق Scoping (تعیین Requester):**
   - اگر `group` برابر **base/b2e** باشد: کوئری روی `requester_id` ورودی فیلتر میشود.
   - اگر **colleague/b2b** یا **agent/b2c** باشد: کوئری روی `operator->id` (کاربر لاگین شده) قفل میشود.
   - در غیر این صورت: همه درخواستها نمایش داده میشوند.
3. **غنی‌سازی داده‌ها (Data Enrichment Loop):** پس از دریافت داده‌های خام از جدول `travel_requests`، روی هر آیتم عملیات زیر انجام میشود:
   - **Polymorphic Requester:** بسته به `operator_group`، اطلاعات درخواست‌دهنده از جدول `customers` (برای B2C) یا `colleagues` (برای B2B از طریق جدول واسط `colleague_auth`) واکشی میشود.
   - **Cities:** نام فارسی و انگلیسی `origin` و `destination` از جدول `cities` جوین میشود.
   - **Operator:** جزئیات اپراتور هندلکننده (شامل آواتار) اضافه میشود.
   - **Reference (Factor):**
     - سریال فاکتور از جدول `factors` خوانده شده و با **10000** جمع میشود.
     - عنوان فاکتور از **Redis** با کلید `reference:{id}:information:title:fa` استعلام میشود.
   - **Accommodation:** اگر هتل باشد، اطلاعات آن از طریق متد استاتیک `StaticController::getAccommodation` دریافت میشود.
   - **Dynamic Title Construction:** یک عنوان توصیفی شامل "مبدا + مقصد + هتل + تاریخها" ساخته میشود. (تاریخها برای نمایش صحیح در متون فارسی با کاراکتر کنترل یونیکد `\u{200E}` محصور میشوند).

- **Details:** فیلد `details` که JSON است، دیکد می‌شود.

Flowchart (Request Processing)

Start Request
↓
Calc Pagination
Page = (Start + Length) / Length
↓
Check 'group' param

base/b2e
Filter by Input Requester_ID
b2b/b2c
Filter by Auth Operator ID
Others
No Filter (Admin View)

↓
DB Query & Filters
Apply branch, method, status...
↓
Data Mapping (Loop)
Fetch Cities, Operator, Factor (Redis), Hotel info
↓
Generate Title
Origin + Dest + Hotel + Dates
↓
Return JSON (Items + Meta)

Response Structure

```
{
  "items": [
    {
      "id": 123,
      "branch": 1,
      "requester_id": { "id": 50, "first_name": "...", "last_name": "..." }, // Object
(Customer or Colleague)
      "origin": { "id": 1, "fa_name": "تهران", "en_name": "Tehran" },
      "destination": { "id": 2, "fa_name": "مشهد", "en_name": "Mashhad" },
      "reference": {
         "id": 99,
```

```
        "serial": 10055, // Original + 10000
        "title": "تور هشده" // From Redis
      },
      "accommodation": {
        "title": { "fa": "هتل درویشی" },
        "category": { "title": "hotel", "subtitle": false }
      },
      "title": "تهران به هشده هتل درویشی از 1403/01/01 تا 1403/01/04", // Generated String
      "details": { ... } // Decoded JSON
      // ... other fields
    }
  ],
  "meta": {
    "timestamp": 1715000000,
    "table": {
      "total": 100,
      "per_page": 30,
      "current_page": 1,
      "last_page": 4,
      "from": 1,
      "to": 30,
      "next_page": 2
    }
  }
}
```


Revision #1
Created 2025-12-09 06:56:51 UTC by Alireza
Updated 2025-12-09 06:58:54 UTC by Alireza