## **Cruise Class Filtering in Tourplan API**

For codes, check: /docs/tourplan/Tourplan Codes-service-button.csv

Cruises are returned via the API with Country > Destination > Class filtering capability. Here's how the filtering system works:

### **How the Filtering System Works:**

1. **GetProductSearchData Request**: This is the key request that provides all necessary filtering data for cruise searches. It returns three main elements:
   * CodeTables - Contains various code tables including class codes
   * CountryList - Lists countries and their destinations
   * ServiceButtons - Service button details with associated filtering codes
2. **Class Codes Structure**:
   * **CodeTableType**: CLS represents the class codes table
   * Class codes are returned per country/destination combination
   * Each destination can have different class codes available
3. **Service Button Structure for Cruises**:
   * **ButtonName**: "Cruises"
   * **Button No**: 91 (from the service codes documentation)
   * **Product Type**: Non Accommodation
   * **ClassLabel**: Each service button has a ClassLabel field

### **API Structure for Class Filtering:**

| <**GetProductSearchDataReply**>  <**CodeTables**>  <**CodeTable**>  <**CodeTableType**>CLS</**CodeTableType**> <!-- Class codes -->  <**CodeTableEntries**>  <**CodeTableEntry**>  <**Code**>LUXURY</**Code**>  <**Description**>Luxury Class</**Description**>  </**CodeTableEntry**>  <!-- More class codes... -->  </**CodeTableEntries**>  </**CodeTable**>  </**CodeTables**>  <**ServiceButtons**>  <**ServiceButton**>  <**ButtonName**>Cruises</**ButtonName**>  <**ClassLabel**>Class</**ClassLabel**>  <**ButtonCountries**>  <**ButtonCountry**>  <**CountryCode**>ZA</**CountryCode**>  <**DestinationList**>  <**DestinationInfo**>  <**DestinationCode**>WCP</**DestinationCode**>  <**CodeLists**>  <**CodeList**>  <**CodeTableType**>CLS</**CodeTableType**>  <**Code**>BASIC</**Code**>  <**Code**>STANDARD</**Code**>  <**Code**>LUXURY</**Code**>  </**CodeList**>  </**CodeLists**>  </**DestinationInfo**>  </**DestinationList**>  </**ButtonCountry**>  </**ButtonCountries**>  </**ServiceButton**>  </**ServiceButtons**> </**GetProductSearchDataReply**> |
| --- |

For Cruise Search Requests:

| <**OptionInfoRequest**>  <**AgentID**>YOUR\_AGENT\_ID</**AgentID**>  <**Password**>YOUR\_PASSWORD</**Password**>  <**ButtonName**>Cruises</**ButtonName**>  <**CountryName**>South Africa</**CountryName**> <!-- Country filtering -->  <**DestinationName**>Western Cape</**DestinationName**> <!-- Destination filtering -->  <**Class**>LUXURY</**Class**> <!-- Class filtering -->  <**DateFrom**>2025-07-15</**DateFrom**>  <**RoomConfigs**>  <**RoomConfig**>  <**Adults**>2</**Adults**>  <**RoomType**>CABIN\_TYPE</**RoomType**>  </**RoomConfig**>  </**RoomConfigs**> </**OptionInfoRequest**> |
| --- |

### **Troubleshooting Your Class Filtering Issues:**

1. **First, get the available class codes**:
   * Make a GetProductSearchDataRequest to get all available class codes for cruises
   * Check what class codes are actually available for your specific country/destination combinations
2. **Verify the class codes match exactly**:
   * The class codes in your search requests must match exactly with what's returned from GetProductSearchData
   * Class codes are case-sensitive
3. **Check your current implementation**:
   * Ensure you're using the correct XML element <Class> in your OptionInfoRequest
   * Verify the class codes you're using exist in the Tourplan system for cruise products

The filtering system definitely supports Country > Destination > Class filtering for cruises, but the specific class codes available depend on what's configured in your Tourplan system for each destination.

## **Key Issues with Cruise Products in Tourplan API:**

### **1. Product Classification Issues**

* **Service Type**: Cruises have SType = "N" (Non Accommodation) but are configured as "Non Accommodation" product type
* **Button Number**: Cruise button number is 91 with sequence 100, which places it differently in the product hierarchy
* **This mismatch** between being called "Cruise" but classified as "Non Accommodation" can cause confusion in filtering

### **2. Rate Structure Complications**

**Cruises can be configured in two different ways**, causing different rate structures:

#### **Pax-Based Pricing (SType=N, MPFCU=1):**

* Rates returned as <PersonRates> with <AdultRate> and <ChildRate>
* Per-person pricing model
* Example from docs shows cruise products using this structure

#### **Group-Based Pricing (SType=N, MPFCU>1):**

* Rates returned as <OptionRates> with <OptionRate> elements
* Group pricing up to MPFCU passengers
* More complex to handle in your booking engine

### **3. Room Configuration vs Cabin Configuration**

**Major Issue**: The API documentation shows cruise requests using <RoomConfigs> and <RoomType> but cruises should logically use cabin configurations:

| <!-- This is confusing for cruises --> <**RoomConfigs**>  <**RoomConfig**>  <**Adults**>2</**Adults**>  <**RoomType**>CABIN\_TYPE</**RoomType**> <!-- Should be CabinType? -->  </**RoomConfig**> </**RoomConfigs**> |
| --- |

### **4. Fixed-Length Package Considerations**

* Document mentions cruises have "Fixed-length packages" as a special consideration
* Unlike accommodation, you don't specify number of nights - the cruise duration is pre-determined
* **No SCU (Second Charge Units)** should be specified for cruise bookings

### **5. Class Filtering Issues**

Based on your original question about class filtering problems:

#### **Potential Root Causes:**

1. **Class codes may not be properly returned** for cruise products in GetProductSearchData
2. **Different class code structure** - cruises might use different class categories than other products
3. **Missing or incomplete CodeList** for cruise destinations
4. **Case sensitivity** in class code matching

### **6. API Response Structure Differences**

**Non-Accommodation products have different response shapes:**

* Quote from docs: *"Non-Accommodation product can be a different shape, the key difference is that the breakdown of the rate, reflecting that the product does not have room types associated with it."*
* This means cruise responses won't have <RoomRates> but will have <PersonRates> or <OptionRates>

## **Recommendations to Fix Your Issues:**

### **For Class Filtering Problems:**

1. **Test GetProductSearchData specifically for cruises:**

| <**GetProductSearchDataRequest**>  <**AgentID**>YOUR\_AGENT</**AgentID**>  <**Password**>YOUR\_PASSWORD</**Password**> </**GetProductSearchDataRequest**> |
| --- |

1. **Check the ServiceButton response for cruises** - verify that class codes are actually returned for cruise destinations
2. **Verify class code format** - ensure your filtering uses exact class codes returned by the API

### **For Rate Structure Issues:**

1. **Check the SType and MPFCU values** in your cruise product responses
2. **Handle both rate structures** in your booking engine:
   * PersonRates for pax-based cruises
   * OptionRates for group-based cruises
3. **Don't send SCUqty** in AddService requests for cruises

The main issue is likely that **cruises are treated as "Non Accommodation" products with special handling requirements** that differ from standard accommodation products, and your current filtering logic may not account for these differences.

Current Behavior for New Cruise Products

❌ What WON'T work automatically:

1. Product Discovery: New products won't appear in search results because:

- We use a hardcoded list in ALL\_CRUISES array (services.ts:25-32)

- We use static region mapping in getCruiseProductsForRegion()

- The TourPlan API search returns empty for cruises, so we rely on catalog approach

2. Class Filtering: New products won't be classified correctly because:

- Our class filtering uses hardcoded product code patterns

- Unknown product codes will fail class matching and be filtered out

✅ What WOULD work:

1. Individual Product Pages: Direct product links would work (e.g., /products/NEW\_PRODUCT\_CODE)

2. Booking Flow: New products could be booked if accessed directly

3. API Integration: All the XML/API infrastructure supports new products

How to Make It Future-Proof

Option 1: Semi-Automated (Recommended)

Update the hardcoded lists when new products are added:

// lib/tourplan/services.ts

const ALL\_CRUISES = [

'BBKCRCHO018TIACP2', // Existing

'BBKCRCHO018TIACP3', // Existing

'BBKCRTVT001ZAM2NM', // Existing

'BBKCRTVT001ZAM2NS', // Existing

'BBKCRTVT001ZAM3NM', // Existing

'BBKCRTVT001ZAM3NS', // Existing

'NEW\_CRUISE\_PRODUCT\_CODE', // ADD NEW ONES HERE

];

And update class filtering patterns based on new product naming conventions.

Option 2: Fully Automated

Create a GetServiceButtonDetails request to discover cruise products dynamically, but this would require:

- Implementing the button details API call

- Parsing the response structure

- Creating dynamic class mapping logic

- More complex error handling

Option 3: Hybrid Approach

- Keep hardcoded list for known products (fast, reliable)

- Add fallback API discovery for unknown products

- Admin interface to manage product catalog

Recommendation

For now, Option 1 (Semi-Automated) is best because:

- ✅ Reliable: No dependency on API quirks

- ✅ Fast: No additional API calls

- ✅ Predictable: You control exactly what appears

- ✅ Simple: Just update arrays when new products added

When TourPlan adds new cruise products, you'll need to:

1. Add product codes to ALL\_CRUISES array

2. Add class filtering patterns based on new product naming

3. Test the new products appear in correct classes