## **Overwatch API Integration & Threat Modeling**

### **1. API Endpoint Overview**

**URL:**

GET <https://overfast-api.tekrop.fr/players/{player_id}/stats/summary>

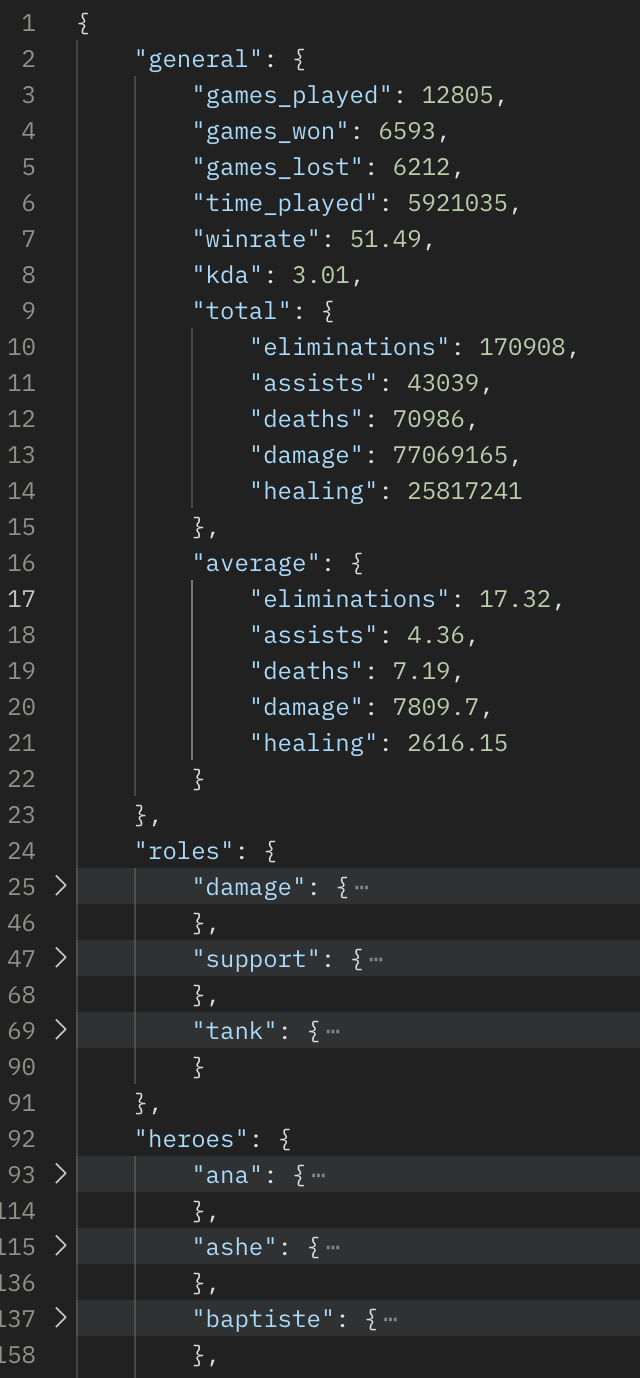
**Notes:**

* Public endpoint (no API key required).
* Replace # with - in BattleTag (e.g., PlayerOne-1234).
* Rate limit: 30 requests per second per IP.

### **2. Request Parameters**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Location** | **Description** |
| player\_id | path | User’s BattleTag (case-sensitive). |

### **3. Example JSON Response**



Roles and Heroes follow the same structure as general.

(

Similar pattern as our idea [top-down]:

General -> Global profile stats

Roles -> Category (FPS, MOBA, ...) stats

Heroes -> Individual games stats

)

### **4. Error Responses**

|  |  |  |
| --- | --- | --- |
| **Status Code** | **Response** | **Reason** |
| 404 | { "detail": "Player not found" } | Invalid/private profile |
| 429 | { "detail": "Rate limit exceeded." } | Exceeded request rate limit |
| 400/422 | Validation error JSON | Incorrect player\_id format |

### **5. STRIDE Threat Model**

|  |  |  |
| --- | --- | --- |
| **Threat Type** | **Risk Description** | **Recommended Mitigations** |
| **Spoofing** | Fake or unauthorized requests | Use HTTPS & certificate pinning, user-bound JWT tokens |
| **Tampering** | Malformed or injected player\_id input | Strict client/server-side input validation |
| **Repudiation** | User denies linking username | Log all username additions with timestamps and user IDs |
| **Information Leak** | Sensitive data exposure locally or in transit | Encrypt data storage, app-private storage, secure logging |
| **Denial of Service** | API abuse by excessive requests | Client-side request throttling, exponential backoff handling |
| **Privilege Elevation** | Exploit logic to access other users’ data | Android least-privilege permissions, strict user binding |

Check Postman VGS workspace for more info