Concurrency governance cheat sheet

2 governance types are in place simultaneously:

**User-level limit**
Defines limit per user and applies for specific authentication method and specific API. User-level limit defines maximum but does not guarantee minimum available due to account limit (10 requests for concurrent WS user (cWSu)* are not guaranteed).

**Account-level limit**
This limit applies to the combined total of SOAP Web Services (WS) and RESTlet requests per given account. Maximum request count is derived from the service tier, the number of SuiteCloud Plus (SC+) licenses and account type (developer accounts have base limit = 5).

### API and Authentication Method

**Authentication Method**

<table>
<thead>
<tr>
<th>API</th>
<th>Request-level Credentials (RLC)</th>
<th>Login/Logout (L/L)</th>
<th>SSOLogin</th>
<th>Token-based (TBA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAP WS</td>
<td>1/10**</td>
<td>1/10</td>
<td>1/10</td>
<td>No limit per user</td>
</tr>
<tr>
<td>RESTlet</td>
<td>No limit per user</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>No limit per user</td>
</tr>
</tbody>
</table>

* One concurrent WS user (cWSu) can be defined on employee record for each SC+ license you get

** 1/10 = 1 request/user OR max 10 requests/cWSu. You can have 1 concurrent request per user at given time OR maximum 10 concurrent requests if it is cWSu

### Service Tier and Account Limit

<table>
<thead>
<tr>
<th>Service Tier</th>
<th>Account Base Limit*</th>
<th>1 SC+ Licence</th>
<th>2 SC+ Licence</th>
<th>10 SC+ Licence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared, 3</td>
<td>5 concurrent requests for the entire account</td>
<td>5+1×10=15</td>
<td>5+2×10=25</td>
<td>5+10×10=105**</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10+10</td>
<td>10+20</td>
<td>10+100**</td>
</tr>
<tr>
<td>1, 1+, 0</td>
<td>15</td>
<td>15+10</td>
<td>15+20</td>
<td>15+100</td>
</tr>
</tbody>
</table>

* The base limit is increased by 10 for each SC+ license. The number of SC+ licences may vary from 1 to many.

** Not a standard license count for this service tier

### Sample Scenarios – how many concurrent requests can I have?

#### End users/Client application(s) are querying my company account with following amount of requests

<table>
<thead>
<tr>
<th>Scenario</th>
<th>SOAP WS Requests Total</th>
<th>RESTlet Requests Total</th>
<th>Total Account Limit</th>
<th>Success</th>
<th>Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>27</td>
<td>25</td>
<td>12</td>
<td>2</td>
</tr>
</tbody>
</table>

**User-level limits apply here**

**User A** is concurrent WS User (company can enable it for 2 users because it has 2 SC+ licenses). **User A** can send max 10 concurrent requests. Without this privilege, 9 requests of user A would be rejected and all others processed.

**User B** is using TBA and does not have to be a concurrent WS user to be able to send more requests. TBA is recommended authentication method with even more benefits.

**Account-level limit applies here**

External application(s) send concurrently 27 requests in total. Any 2 requests can be rejected due to account limit 25.

Concurrent vs sequential requests: If one of above requests is being processed for 1 minute (e.g. a long search), during that 1 minute can be sent and processed MORE than 24 other requests if those requests are not reaching the server in one moment, but sequentially after response to the other request.

* Please note: Governance framework can temporarily shortly allow more concurrent requests

---

See Help Center for other scenario
Concurrent governance cheat sheet

## Recommended Actions

1. Analyse the frequency and level of concurrency peaks and consider rescheduling requests to be outside of regular peak times.
2. Consider if more SC+ licenses are needed and learn on SC+ settings.
3. Handle the error codes in client application.
4. Implement retry logic.
   a. Retry gradually increasing the delay if more attempts needed.
5. For non-concurrent users serialize your requests in client applications to not overlap.
6. Use TBA to take advantage of a more flexible concurrency.
7. Monitor trends in concurrency usage to prevent broken integrations (see Navigation table below).

### Code example demonstrates basic handling of WS error codes

```java
int i = 0;
int maxAttempts = 5; // try it 5 times, then fail for good

while (i < maxAttempts) {
    response = doWSCall();
    isSuccess = response.getIsSuccess();
    errorMsg = response.getErrorMsg();

    if (isSuccess == false && (errorMsg == WS_CONCUR_SESSION_DISALLOWED || errorMsg == WS_REQUEST_BLOCKED)) {
        wait();
        i++;// try again
    } else {
        break; // end the cycle
    }
}
```

### Method | Error codes | Error codes
--- | --- | ---
Web Services + L/L or RLC | ExceededRequestLimitFault | WS_CONCUR_SESSION_DISALLOWED
Web Services + TBA | ExceededConcurrentRequestLimitFault | WS_REQUEST_BLOCKED
RESTlet | HTTP error code: 400 Bad Request | SuiteScript error code: SSS_REQUEST_LIMIT_EXCEEDED

Error can occur for any of the requests that exceed the limit at that moment

### NetSuite navigation

**What**

- Account concurrency limit
- If account concurrency limit is enabled
- Total requests (number, ratio)
- Rejected requests
- Reports about rejected SOAP WS requests
- Reports about rejected RESTlet requests
- Details about SOAP WS requests that were rejected due to concurrency violation
- Web Services performance dashboard
- Concurrency Monitor dashboard, monthly/hourly overview (heatmap), charts showing concurrency usage with drill down possibility to seconds
- Scheduling of integrations

**Where**

- Setup > Integration > Integration Management > Integration Governance
- Reports > New Search -> Web Services Operations
- RESTlet script record > Log
- Setup > Integration > Web Services Usage Log
- The Application Performance Management SuiteApp (link)
- Help Center article