# Dokumentation: LoggedIn-User-Info für Coacher ([BNRDR-1672](https://jira.cic-software.de:8443/browse/BNRDR-1672))

## Allgemein

Dieser Service bietet Überblick über die eingeloggten User:

1. Wieviele User sind eingeloggt?
2. Welcher User war wann und wielange eingeloggt?
3. ServiceCalls sollen nachvollziehbar sein.

Das Logging selbst wird automatisch vom WebService durchgeführt. Durch die Eingabe der Info für die Felder sessionTimeoutMin, appId, hostID und userSource mittels **UpdateLoggedInLogProperties** im Frontend kann die Art und Weise des Loggings bestimmt werden oder ein (zusätzlicher) Log-Eintrag ausgelöst oder eine Liste der aktuell geloggten User ermittelt werden.

Die Eigenschaften im Einzelnen:

1. SessionTimeoutMin
2. Application-ID
3. Host-ID
4. User-Source

## Schnittstellen Lösungen

Neu In CIC.One::AppManagementservice: Statics im Modul **LogUtil**:

* **UpdateLoggedInLogProperties** (iLoggedInLogPropertiesDto dto);
* List<CicLogDto> **ListLoggedIn** (CicLogDto searchPattern)
* **RegisterLoggedIn** (long sysperole, int sessionTimeoutMin, long appId, string hostID, string userSource)

## Features

* **RegisterLoggedIn:** Mitloggen aller User die sich einloggen:
  + Bei LOGIN
  + Bei LOGOUT
  + Bei allen Aktionen (ausser Aktionen innerhalb einer Maske)
  + UNKNOWN USER – Logging (User mit falscher (nicht findbarer) SysPeRole)
* **ListLoggedIn:** Liste alle User zu einem bestimmten Zeitpunkt:
  + Suchen aller aktuellen Users, wenn:
    - searchPattern.cicbenutzer NULL ist
    - LoggedInSearchPattern NULL ist
* **LoggedIn-Dictionary**, damit maximal EINMAL pro Minute gelogged wird:  
  static Dictionary<long, DateTime> dictLoggedIn //assert logging each minute only
* **Schalter** der das LogIn-Logging EIN/AUS schalten kann:  
  LogLoggedInUser in Properties\Config.settings true oder false

## Interface

### logLoggedIn

/// <summary>

/// Log loggedIn users

/// rh 20170203

/// </summary>

/// <returns></returns>

public ologLoggedInDto logLoggedIn (ilogLoggedInDto dto)

{

ServiceHandler<ilogLoggedInDto, ologLoggedInDto> ew = new ServiceHandler<ilogLoggedInDto, ologLoggedInDto> (dto);

return ew.process (delegate (ilogLoggedInDto input, ologLoggedInDto rval)

{

LogUtil.RegisterLoggedIn (input.sysperole);

});

}

### listLoggedIn

/// <summary>

/// List loggedIn users

/// rh 20170203

/// </summary>

/// <returns></returns>

public oLoggedInListDto listLoggedIn (iLoggedInListDto dto)

{

// rh 20170206 Feature: support empty LoggedInSearchPattern-List

if (dto == null)

dto = new iLoggedInListDto();

ServiceHandler <iLoggedInListDto, oLoggedInListDto> ew = new ServiceHandler <iLoggedInListDto, oLoggedInListDto> (dto);

return ew.process (delegate (iLoggedInListDto input, oLoggedInListDto rval)

{

rval.loggedInUsers = LogUtil.ListLoggedIn (input.loggedInSearchPattern);

});

}

### **CicLogDto - Format**

nach CICLOG-Entity-Format:

|  |  |  |
| --- | --- | --- |
| **Name** | **Entity-Typ** | **Mapping** |
| SYSCICLOG | NUMBER(12) | long sysciclog |
| LOGINDATE | DATE | DateTime? logindate |
| LOGOUTDATE | DATE | DateTime? logoutdate |
| ORABENUTZER | VARCHAR2(20) | string orabenutzer |
| CICBENUTZER | VARCHAR2(20) | string cicbenutzer |
| MASCHINE | VARCHAR2(20) | string maschine |
| ID | NUMBER(12) | long? id |
| SOURCE | VARCHAR2(128) | string source |

### **iLoggedInLogPropertiesDto**

|  |  |
| --- | --- |
| **Name** | **Mapping** |
| sessionTimeoutMin | int sessionTimeoutMin |
| appId | long? appId |
| hostID | string hostID |
| userSource | string userSource |

### **ON-Status-Matrix**

|  |  |  |
| --- | --- | --- |
| **LogInDate** | **LogOutDate** | **Status** |
| NULL | NOT NULL | UNKNOWN 🡪 OFF |
| VALID | NULL | UNKNOWN 🡪 OFF |
| VALID | >= Now: | ON |
| VALID | < Now | OFF |

SET:

* KEIN LogInDate 🡪 User noch nicht registriert
* LogInDate ABER KEIN LogOutDate 🡪 kein von LoggedInUser-Log geloggter Datensatz 🡪 User wird NEU als ON registriert
* LogInDate UND LogOutDate in der Zukunft 🡪 User als ON registriert
* = wenn User schon registriert ist und eine neue Aktivität zeigt, wird das LogoutDate um das sessionInactivityTimeout verlängert (= wahrscheinlicher Logout-Zeitpunkt):

### **ON-Action-Matrix**

|  |  |  |
| --- | --- | --- |
| **LogInDate** | **LogOutDate** | **SWITSCH** |
| VALID | < Now | INSERT new CICLOG-entry Now + TimeOut 🡪 Status = ON |
| VALID | >= Now | ONLY UPDATE LogOutDate:  Now + TimeOut 🡪 Status = ON |
| NULL | NOT NULL | INSERT new CICLOG-entry Now + TimeOut 🡪 Status = ON |
| VALID | NULL | INSERT new CICLOG-entry Now + TimeOut 🡪 Status = ON |

## SAMPLE Calls

### Sample SET logLoggedIn-property

//—————————————————————————————————————————————————

// rh 20170208: Sample SET logLoggedIn-property

//—————————————————————————————————————————————————

ILoggedInLogPropertiesDto loggedInLogProperties **=** **new** ILoggedInLogPropertiesDto **();**

loggedInLogProperties**.**setSessionTimeoutMin**((**int**)** sessionTimeout**/**1000**/**60**);**

loggedInLogProperties**.**setAppId**((**long**)** 717171**);**

loggedInLogProperties**.**setHostID**(**"XY-HostID"**);**

loggedInLogProperties**.**setUserSource**(**"SIRC:TheOne"**);**

getService**().**getappManagementService**().**updateLoggedInLogProperties**(**loggedInLogProperties**);**

### Sample CALL to logLoggedIn

//—————————————————————————————————————————————————

// rh 20170208: Sample CALL to logLoggedIn

//—————————————————————————————————————————————————

IlogLoggedInDto iliUser **=** **new** IlogLoggedInDto **();**

// (long sysperole, int sessionTimeoutMin, long appId, string hostID, string userSource

iliUser**.**setSysperole **(**ouser**.**getUserData**().**getSysperole**());** // Apply current Sysperole

iliUser**.**setTimeOutMin **((**int**)** sessionTimeout**/**1000**/**60**);** // SET TimeOut in Minutes

iliUser**.**setAppId **((**long**)** 717171**);** // Apply AppID

iliUser**.**setHostID **(**"XY-HostID "**);** // Apply HostID

iliUser**.**setUserSource **(**"SIRC:TheOne"**);** // Apply UserSource

getService**().**getappManagementService**().**logLoggedIn**(**iliUser**);**

### Sample CALL to listLoggedIn

//—————————————————————————————————————————————————————————————

// rh 20170208: Sample CALL to listLoggedIn (= search CICLOG)

//—————————————————————————————————————————————————————————————

// SET search pattern data

CicLogDto searchpattern **=** **new** CicLogDto**();**

searchpattern**.**setCicbenutzer**(**"07333377"**);** // User 07333377

searchpattern**.**setLogindate **(new** Date **(**2017 **-** 1900**,** 01**,** 02**));** // LogIns seit 2017.02.02

// SET search pattern

ILoggedInListDto iliUserList **=** **new** ILoggedInListDto **();**

iliUserList**.**setLoggedInSearchPattern **(**searchpattern**);**

// List logged-in users according to searchpattern

OLoggedInListDto oliList **=** getService**().**getappManagementService**().**listLoggedIn **(**iliUserList**);**

### Sample GET data (Daten aus dem Log holen)

//——————————————————————————————————————

// GET data

//——————————————————————————————————————

CicLogDto searchpaternTEST **=** iliUserList**.**getLoggedInSearchPattern**();**

searchpaternTEST**.**setCicbenutzer **(**ouser**.**getUserData**().**getCode**());**

List**<**CicLogDto**>** loglist**;**

loglist **=** oliList**.**getLoggedInUsers**().**getCicLogDtos**();**

Iterator**<**CicLogDto**>** iter **=** loglist**.**iterator**();**

**while** **(**iter**.**hasNext**())**

**{**

CicLogDto logrow **=** iter**.**next**();**

String s **=** logrow**.**getCicbenutzer**();** // TEST: GET every Cicbenutzer in the list

Date x **=** logrow**.**getLogindate**();** // TEST every Logindate in the list

**}**

## Questions + Answers

* Wie lange darf ein LogInDate zurückliegen, dass es noch als ON akzeptiert wird?
  + WENN LoginDate + TimeOut > Now

## Known Issues

* Beim Verweilen innerhalb einer Maske wird das Logging beim nächsten Makenwechsel einen neuen LogIn-Eintrag anlegen und nicht den „alten“ LogIn-Eintrag verlängern/erneuern.
* Das direkte LogOut kann in der aktuellen Version nicht direkt als LogOut geloggt werden, jedoch erfolgt beim LogOut sicher noch ein Update des wahrscheinlichen“ LogOut-Zeitpunktes.
* SessionTimeOut in SvcInfo-File
  + <property path="/reliableSession/inactivityTimeout" isComplexType="false" isExplicitlyDefined="false" clrType="System.TimeSpan, mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"> <serializedValue>00:10:00</serializedValue>

The Timeout property can be set in the Web.config file for an application using the **timeout** attribute of the configuration element, or you can set the Timeout property value directly using application code.

The Timeout property cannot be set to a value greater than 525,600 minutes (1 year). The default value is 20 minutes.

ATTENTION!:

But it will be null unless you also specify EnableSession=true:

[System.Web.Services.WebMethod(EnableSession = true)]

* Wenn ein User innerhalb der TimeOut-Zeitspanne nochmal ausloggt und nochmal einloggt, dann wird er im Logging als „durchgehend“ eingeloggt eingetragen.
* Wenn die Properties einmal vom Frontend gesetzt wurden, dann bleiben sie solange im WebService eingetragen bis die Properties geändert werden oder der WebService neu gestartet wird.
* Wenn die Properties vom FrontEnd NICHT GESETZT werden, dann werden (nur) folgende Werte mitprotokolliert:

|  |  |
| --- | --- |
| **Name** | **Wert** |
| SYSCICLOG | --- (Any) |
| LOGINDATE | DATE |
| LOGOUTDATE | DATE + TimeOut |
| ORABENUTZER | NULL |
| CICBENUTZER | BenutzerName (=RICHTIGER WERT) |
| MASCHINE | NULL |
| ID | -1 |
| SOURCE | NULL |