**Min hälsoplan**

**Auditloggning**

**Revisionshistorik**

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Beskrivning** | **Författare** |
| 2014-02-11 | PA1 | Första utkastet | JV |

**Innehållsförteckning**

1 Inledning 5

2 Syfte 5

3 Loggning 5

3.1 Teknisk loggning 5

3.1.1 Loggning till systemloggen 5

3.2 PDL-loggning 5

3.2.1 Vilken information som loggas 5

3.2.2 Lagring av loggen 5

3.2.3 När loggningen utförs 6

3.2.4 Komprimering av loggen 6

3.2.5 Åtkomst till loggen 6

3.2.6 Framtida utökning 7

**Definitioner och Referenser**

**Definitioner**

|  |  |
| --- | --- |
| **Term** | **Beskrivning** |
| HSA-ID | En identitet för en person, funktion eller enhet i [vårdens elektroniska katalog](http://www.inera.se/TJANSTER--PROJEKT/HSA/) (HSA). |
| PDL | [Patientdatalagen](http://www.datainspektionen.se/fragor-och-svar/faq-patientdatalagen).  http://www.datainspektionen.se/fragor-och-svar/faq-patientdatalagen |

**Referenser**

|  |  |
| --- | --- |
| **Dokument** | **Länk** |
| Min hälsoplan webb | <http://www.minhalsoplan.se> |
| Min hälsoplan programvara | <https://github.com/MinHalsoplan/netcare-healthplan> |
| Min hälsoplan demo | <http://demo.minhalsoplan.se/v2> |

# Inledning

Min hälsoplan är en e-hälsotjänst i Mina vårdkontakter där vårdpersonal och patienter tillsammans skapar en hälsoplan. Hälsoplanen innehåller en eller flera aktiviteter som patienten får påminnelse att göra, samt kan han eller hon rapportera in resultat från aktiviteten direkt till personalen. Syftet med Min hälsoplan är att stötta både patient och personal under en kort eller lång uppföljning genom att öka tillgänglighet och kvalitet på information samt skapa en ökad kontinuitet i vårdprocessen.

Se även: <http://www.minhalsoplan.se>

# Syfte

Detta dokument beskriver hur loggning sker i systemet.

# Loggning

## Teknisk loggning

### Loggning till systemloggen

Varje gång servern anropas så passerar man genom REST-apiet som implementeras av någon klass som ärver av org.callistasoftware.netcare.api.rest.ApiSupport.

Den klassen innehåller en metod som anropas vid varje publikt metodanrop i subklasserna:

**void** debugLog(**final** String action, **final** String what)

Exempel på hur det kan se ut i loggen:

2014-02-11 12:11:04 [...] INFO AlarmApi - User hsa-cg-2, Action: list->alarms

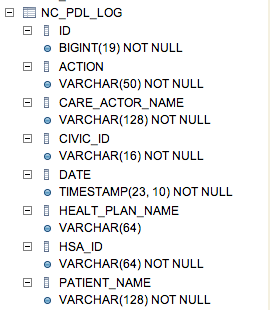
## PDL-loggning

### Vilken information som loggas

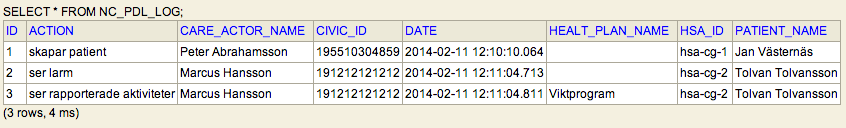
* Vilken vårdperson som är inloggad – HSA-ID och namn
* Vilken patient som avses – personnummer och namn
* Vilken åtgärd som utförts som exempelvis ”ser larm” eller ”skapar patient” (möjliga värden finns i konfigurationsfilen action.properties)
* Vilken tidpunkt det utfördes
* Vilken hälsoplan som avses eller tomt om det inte är någon speciell hälsoplan (tex när man skapar en ny patient)

### Lagring av loggen

I en tabell i databasen som heter NC\_PDL\_LOG med följande utseende



Exempel på loggrader:



### När loggningen utförs

Varje gång information skall hämtas från servern så passerar man genom REST API:et vars implementationsklass ärver från org.callistasoftware.netcare.api.rest.ApiSupport, som

innehåller en metod som anropas för varje publikt metodanrop i subklasserna:

void pdlLog(String,String,PatientBaseView,String,HttpServletRequest)

### Komprimering av loggen

2 gånger per dygn körs ett bakgrundsjobb som tar bort rader som representerar dubbletter med avseende på vårdperson, patient, hälsoplan, action där de olika accesserna skett inom samma 24-timmarsperiod. Raden med senast datum behålls.

Detta görs för att inte informationen skall blir för omfångsrik. Till exempel visas en lista med larm ganska ofta och för en flitig användare kan det innebära att samma larm kommer upp i listan 10-20 ggr under några minuters användning. Utan komprimering finns det risk att 10 – 100 miljoner rader kommer att skapas under den period som det måste lagras. Med komprimering bedöms antalet rader hålla sig till ett fåtal miljoner rader på 5 år men det beror givetvis på hur många patienter som använder system. Om antalet stiger dramatiskt jämfört med idag måste kanske strategin omprövas.

### Åtkomst till loggen

Idag sker åtkomst endast via en databashanterare där man kan ställa SQL-frågor. För att se vem som har begärt åtkomst för en viss patient ställer man frågan:

SELECT HSA\_ID, CARE\_ACTOR\_NAME, MAX(DATE)

FROM NC\_PDL\_LOG

WHERE CIVIC\_ID = '191212121212'

GROUP BY HSA\_ID, CARE\_ACTOR\_NAME

ORDER BY HSA\_ID, CARE\_ACTOR\_NAME

För att se exakta detaljer om hur många gånger, och vilket typ av information som visats osv så ställer man frågan:

SELECT HSA\_ID, CARE\_ACTOR\_NAME, HEALT\_PLAN\_NAME, DATE,

ACTION FROM NC\_PDL\_LOG

WHERE CIVIC\_ID = '191212121212'

ORDER BY HSA\_ID, DATE

### Framtida utökning

Man skulle kunna skapa ett användargränssnitt som innebär att:

* Användaren matar in ett personnummer på en patient
* Systemet söker ut alla distinkta förekomster angående vårdpersonal som haft tillgång till någon information om vald patient
* Systemet visar dessa i en lista sorterad på senaste åtkomsttidpunkt
* Användaren kan bläddra i listan
* Användaren kan expandera en rad
* Systemet söker du ut alla detaljerade rader från PDL-loggen för kombinationen den vårdperson och patient som raden symboliserar och visar en lista med varje typ av access och datum.

Detta är ett förslag på hur man skulle kunna göra. All information finns lagrad i tabellen NC\_PDL\_LOG.

En annan sak som bör adresseras inom några år är rensning av tabellen vid den tidpunkt då PDL-loggen inte behöver lagras längre. Tills vidare kan det göras manuellt av databasadministratören.