eCoRo Web

Easy Control Room – Manuale Installazione e Configurazione

Requisiti

Sistema Operativo	Fortemente consigliato Linux, ma opportunamente configurato può essere utilizzato anche Windows
RAM	1GB
Spazio disco	100 MB
Web server	Apache Web server 2.4 Con RewriteEngine attivo
Database	MySql 5.5, 5.7 o 8.0
Scripting	PHP 7.1 con attive le estensioni: mbstring PDO_mysql Json PHP 7.4 – supportato, mentre PHP 8 no.

Installazione database

Per creare il database necessario all'applicazione eCoRo è sufficiente eseguire lo script: db/ecoro_initial_database.sql che si trova del file eCoRo_<versione>.zip

È possibile personalizzare le prima righe dello script per indicare il nome del database e dell'utente che si preferisce, ma soprattutto per modificare la password dell'utenza MySQL che verrà utilizzata dall'applicazione.

MySql 5.7

Per installazione su MySQl 5.7 abilitare l'inserimento di date vuote impostando

sql-mode=NO_ZERO_DATE,NO_ZERO_IN_DATE

nel mysql.cnf (Linux) mysql.ini (windows)

MySql 8

Per versioni PHP precedenti la 7.1.16 o 7.2.4, è necessario configurare nella sezione [mysqld] in /etc/mysql/mysql.conf.d/mysqld.cnf:

default_authentication_plugin=mysql_native_password

Abilitare l'utente ecoroUser (oppure quello configurato in config_app.inc.php per accedere al database) ad avere i privilegi di SYSTEM_VALIARBLES_ADMIN con l'istruzione sql:

GRANT SYSTEM_VARIABLES_ADMIN ON *.* TO ecoroUser;

Installazione console web

L'installazione consiste nello scompattare i file eCoRo_<versione>.zip (in eCoRoWeb oppure direttamente nella root) e AldeaInclude.zip (solitamente /AldeaInclude) in directory del web server e personalizzare i file:

Per configurare eCoRo Web è necessario personalizzare le chiavi nei seguenti file:

File config/ecoroConfig.inc.php

// Per solo Test

// ini_set('display_errors', 1);

//error_reporting(E_ALL);

// Per produzione

ini_set('display_errors', 0);

error_reporting(0);

$configRepository = "C:/eCoRo/repository/config"; // Percorso dove sono memorizzati i file di configurazione da inviare ai dispositivi remoti.

$getConfigAuth = false; // se impostato a true richiede l'autenticazione per accedere al servizio GetConfig, solitamente disabilitato

$configAutoAddNewDevice = true; // Se true aggiunge alla tabella ecoro_device i dispositivi che non sono ancora stati censiti.

// Attenzione ad attivarlo potrebbe essere motivo di popolamento database indiscriminato se accessi non autenticati.

$allowUserPublication=true; // Se attivo permette la pubblicazione della configurazione direttamente da parte degli utenti. Per esempio da App

$showAlarms = true; // Se attivo mostra l'opzione per visualizzazione Report Allarmi

$MQTT_SERVER = ""; // Indirizzo del MQTT server comprensivo di porta es. tcp://192.168.1.10:1883/

$ecoroDeviceIds = "eCoRo1;"; // Lista dei dispositivi che utilizzano eCoro che devono ricevere le notifiche con MQTT.Gli dei dispositivi id devono essere separati da punto e virgola.L'id è impostato nella configurazione dell'App e corrisponde al nome della coda.

$showAlarm_GPS=true; // Visualizza in allarmi le colonne della posizione GPS

$showAlarm_WiFi=true; // Visualizza in allarmi le colonne della posizione WiFi

$showAlarm_IPS=true; // Visualizza in allarmi le colonne della posizione IPS

$alarmComment=true; // Se attivo permette di commentare un allarme

// L'eCoRo WebSocket server è necessario se si vuole utilizzare la funzione Monitor

// che viene aggiornata ad ogni allarme

$webSocketServer=localhost; // Server dove è installato l'eCoRo webSocket

$webSocketPort=10200; // Server dove è installato l'eCoRo webSocket

$aldeaIncludePath = $_SERVER['DOCUMENT_ROOT']; // Percorso dove è installato il modulo AldeaInclude

$patrolAminEmail = ""; // Indirizzo email a cui inviare le notifiche per il modulo Patrol

$firstAckSyncOtherMonitor=true; // Se impostato a true, alla ricezione di un ACK viene inviato un messaggio di aggiornamento a tutti i monitor/eCoRoApp

$appUrl="http://192.168.1.10/eCoRoWeb"; // Url dell'App. Indispensabile per il eCoRoWebDeamon

$deviceStatusHistory = false; // Definisce se gli stati dei dispositivi devono essere conservati tutti. Nel caso sia false viene conservato solo l'ultimo stato comunicato.

$deviceChargingtime=false; // Se attivo permette di avere nella lista dispositivi anche la colonna sulla durata dell'ultima ricarica e nel dettaglio status l'intervallo orario della ricarica.

$eCoRoAdminEmail="info@dominio.it";

$passwordRules = ["minPasswordLenght" => 8, "minLowerCase" => 2, "minUpperCase" => 2, "minDigit" => 2, "minSpecialChars" => 2]; // Definisce le regole per le nuove password

$passwordValidity=180; // Numero di giorni di validità della password per gli utenti che hanno la gestione della scadenza password attiva

$passwordHashAlgorithm="SHA1"; // Possibili valori SHA1 o SHA256. Se non specificato viene preso SHA1

$acceptedErrorLogin = 5; // Numero di login in errori accettati nell'intervallo specificato in $elapsErrorLogin prima di bloccare l'utente

$elapsErrorLogin=5; // Intervallo temporale espresso in minuti per i controlli dei login in errori accettati.

$authMode = "DB"; // DB=Database; LA=LDAP Authentication Only; LF= LDAP Authentication adn profile

$ldapHost = "ldap://192.168.1.43:389"; // Preso in considerazione solo per authMode=LA o LF

$ldapDnBase= "ou=People,dc=aldea,dc=it"; // Preso in considerazione solo per authMode=LA o LF

$licUsername ="User"; // Credenziali per verificare aggiornamenti da Cloud

$licPassword ="xxxx";

// Se impostate le seguenti chiavi, vengono utilizzare per controllare in checkHealth che ldap permetta l'autenticazione.

$checkHealth_ldap_user=" ";

$checkHealth_ldap_password="";

// secureInstallation : attivarlo permette di avere alcuni accorgimenti in più per le installazioni che richiedono https

$secureInstallation = false;

File config/config_app.inc.php

$db_name = 'ecoro'; // Nome database mysql

$db_user = 'ecoroUser'; // Utenza per accedere a mysql

$db_password = 'ecoro4578'; //Password per l'utenza mysql

$db_host = "localhost"; // Nome del server mysql

```
$db_port=3306; // Porta TCP di comunicazione su cui ascolta mysql

$db_persistant = true;  // Per velocizzare l'accesso al database
```

Servizio WebSocket

Per l'aggiornamento automatico della funzione Monitor di eCoRoWeb è necessario che sul server venga avviato il servizio che permette le comunicazioni websocket dal server verso i web browser collegati.

Il servizio (disponibile solo per eCoRo Web e non nella versione Lite) si trova nella directory websocket/bin e va attivato tramite l'istruzione seguente come task pianificato all'avvio del sistema.

```
C:\Apache24/htdocs/eCoRoWeb\websocket> startEcoroWebSocketServer.bat
```

Per pianificare l'avvio del servizio al boot di sistema:

Per Linux:

```
sudo cp /var/www/html/websocket/startEcoroWebSocketServer.sh /etc/init.d

sudo chmod 755 /etc/init.d/startEcoroWebSocketServer.sh
sudo update-rc.d startEcoroWebSocketServer.sh defaults
```

Per Windows:

Pianificare l'esecuzione all'avvio dello script:

```
c:\Apache24\htdocs\eCoRoWeb\websocket\startEcoroWebSocketServer.bat
```

Oppure è possibile configurare un servizio per l'esecuzione dello script all'avvio del sistema:

/etc/systemd/system/ecoro_websocket.service

```
[Unit]
Description=eCoRo WebSocket service

Wants=network-online.target
After=syslog.target network-online.target

[Service]
Type=simple
User=root
Group=root
TimeoutStartSec=0
Restart=on-failure
RestartSec=30s
#ExecStartPre=
KillMode=process
WorkingDirectory=/var/www/html/websocket/
```

```
ExecStart=/var/www/html/websocket/startEcoroWebSocketServer.sh
#ExecStop=

[Install]
WantedBy=multi-user.target
```

Eseguire i comandi

```
sudo systemctl enable ecoro_websocket

sudo systemctl deamon-reload

sudo systemctl start ecoro_websocket
```

Demone

Nel caso gli smartphone con VerticalMan siano sotto copertura Wi-Fi e possano sempre raggiungere il server con eCoRoWeb è possibile attivare la verifica della raggiungibilità dei dispositivi. In pratica VerticalMan invia periodicamente lo stato del dispositivo ad eCoRoWeb e un demone applicativo, sempre in esecuzione, verifica se queste comunicazioni siano sempre aggiornate, caso contrario solleva un allarme DNR (Dispositivo Non Raggiungibile). Per attivare questa funzione è necessario:

1. Attivare in VerticalMan->Impostazioni->Notifiche allarme->Notifica via Web->Invio stato sistema via Web
   Con una periodicità di 30 minuti.
2. Configurare in ecoroConfig.inc.php la chiave $appUrl per indicare l'url dell'applicazione eCoRoWeb
3. Avviare lo script startEcoroDeamon.sh/bat in via permanente con le pianificazioni di sistema come per il servizio WebSocket.
4. Da eCoRoWeb impostare per ogni dispositivo se si vuole attivare la verifica della raggiungibilità

Per personalizzare i temporizzatori è possibile modificare le chiavi di configurazione in demon/demon.php

Queste le configurazioni di default per avere segnalazioni di dispositivo non raggiungibile dopo 2 tentativi di comunicazione dello stato previsto ogni 30 minuti dall'App remota e non arrivato dopo 65 minuti.

$timeoutReachable = 65; // After how many minutes a device must be considered not reachable

$intervalCheckReachable = 60; // How many seconds check device reachable

$noNewAlarmWithin = 30; // No alarm if device is not reachable within this parameters in Minutes

Utenza root accesso ad eCoRoWeb

Esecuzione dello script SQL di inizializzazione del database crea l'utenza di accesso ad eCoRo con i più alti privilegi:

Utenza	admin@ecorolite.it
Password	df45HYQ1!

È raccomandato di cambiare la password prima del passaggio in produzione.

Accesso diretto da URL

È possibile utilizzare la chiamata alla pagina auth passando utenza e password e la pagina da visualizzare per automatizzare l'ingresso in eCoRo Web direttamente con un url.

Sintassi url:
http://<NomeServer>/eCoRoWeb/auth?username=<utenza>&password=<password>&firstContent=<content>

Il parametro firstContent permette di definire la pagina di eCoRo Web da aprire. Se non viene passato si aprirà la Home. Per esempio passando firstContent=monitor si aprirà la pagina di monitor

E anche possibile richiedere il login per poi aprire una pagina specifica con il link:

http://<NomeServer>/eCoRoWeb/login?firstContent=<content>

es. http://<NomeServer>/eCoRoWeb/login?firstContent=monitor

Sicurezza del Sistema

E' possibile installare un certificato SSL in Apache httpd per utilizzare il protocollo https.

Per ottimizzare la sicurezza delle comunicazioni https è suggerito di:

1. Attivare il parametro $secureInstallation che invia il parametro header:
   Strict-Transport-Security: max-age=31536000
2. Impostare il flag Secure i cookie tramite la documentazione di PHP : https://www.php.net/manual/en/session.security.ini.php
3. Impostare gli header "X-XSS-Protection: 1; mode=block" tramite la documentazione Apache httpd: https://www.keycdn.com/blog/x-xss-protection
4. Impostare il flag httpOnly i cookie tramite la documentazione di PHP: https://www.php.net/manual/en/session.security.ini.php
5. Dopo un login riuscito, la funzionalità di login deve sempre creare (e utilizzare) un nuovo ID di sessione.Questo è possibile impostando i parametri di PHP: https://www.php.net/manual/en/session.configuration.php

6. Per limitare Information Leakage impostare in php.ini il parametro expose_php e in eCoRoConfig definire:
   ini_set('display_errors', 0);
   error_reporting(0);

Per associare un cifratura TLS/SSL al WebSocket Server è sufficiente seguire le indicazioni riportate in questo articolo che sostanzialmente creerebbe una connessione wss tramite il tool stunnel.Per permette di utilizzare il protocollo wss per il WebSocket impostare la chiave **$webSocketServerProtocol**

L'autenticazione utente è possibile effettuarla (**$authMode**) tramite il database ecoro oppure ldap server.

Raccomandazioni nel caso di autenticazione con LDAP Server

L'utenza tecnica utilizzata per notificare gli allarmi è importante che non abbia scadenza automatica della password per evitare che al momento della segnalazione di emergenza questa venga rifiutata perché la password è scaduta. Si invita a cambiare periodicamente la password di queste utenze manualmente tenendo cura di aggiornare anche i sistemi che la utilizzano, in primis l'App VerticalMan.

Gestione sicurezza password

Se la gestione dell'autenticazione è impostata con il database ecoro (**$authMode** = "DB") le seguenti chiavi permettono di dettagliare le regole per la sicurezza.

Con la personalizzazione della chiave **$passwordRules** si possono definire i criteri minimi delle nuove password impostate dagli utenti.

Mentre con **$passwordValidity** definisce per quanti giorni deve valere la nuova password per gli utenti che hanno attiva la gestione della scadenza password. Nell'anagrafica utente è possibile definire se deve essere attiva questa gestione. E' importante non attivare la gestione della scadenza password per le utenze tecniche utilizzate per l'accesso dalle App remote perché potrebbe rendere il sistema non sicuro in caso di emergenze.

Nel caso si voglia cambiare password alla utente tecniche è necessario gestirlo manualmente così da comprovarne il successo nella procedura di aggiornamento con anche un test per la simulazione di un allarme.

Per limitare le possibilità di login errati si devono utilizzare le seguenti chiavi.

$acceptedErrorLogin = 5; // Numero di login in errori accettati nell'intervallo specificato in $elapsErrorLogin prima di bloccare l'utente

$elapsErrorLogin=5; // Intervallo temporale espresso in minuti per i controlli dei login in errori accettati.

In ogni caso ad ogni login errato viene inviata una mail alla casella configurata in $eCoRoAdminEmail.

Se un'utenza viene bloccata per eccesso di tentativi di login errati, l'amministratore dovrà manualmente riattivarla dalla gestione utenti.

E' possibile definire l'algoritmo con cui la password è registrata sul database con la chiave **$passwordHashAlgorithm**, per un buon livello di sicurezza si suggerisce di utilizzare SHA256.

Per compatibilità con le precedenti installazioni l'algoritmo di default è SHA1.

Attivazioni esterne

eCoRoWeb può eseguire delle attivazioni alla ricezione di un allarme per notificarlo ad un sistema esterno.

Questa funzione è utile per:

Funzione	Tramite
attivare una sirena esterna	Controllore web
allarmare un combinatore telefonico	Controllore web
Inviare SMS	Servizio su Internet
Inviare Email	Server di posta elettronica

Attualmente è possibile attivare solamente sistemi che richiedono chiamate http.

La configurazione delle attivazioni esterne è da effettuarsi direttamente sulla tabella ecoro_externalActivation inserendo almeno un record per ogni signal_id.

Campo	Descrizione
company_id	ID della compagnia
signal_id	ID del segnale/allarme
channel	W=Web
url	URL da richiamare
headers	Eventuali header da indicare alla chiamata dell'url espressi in oggetto json
method	Metodo http da utilizzare per la chiamata. GET o PUT
body	Informazioni da inviare nella chiamata.

Esempio di configurazione per controllori Moxa ioLogik E1200 series

signal_id	channel	url	headers	method	body
ACK	W	http://192.168.1.97/api/slot/0/io/relay/0/relayStatus	{"Accept":"vdn.dac.v1","Content-Type":"application/json"}	PUT	{"slot":0,"io":{"relay":{"0":{"relayStatus":0}}}}
ACK	W	http://192.168.1.97/api/slot/0/io/relay/1/relayStatus	{"Accept":"vdn.dac.v1","Content-Type":"application/json"}	PUT	{"slot":0,"io":{"relay":{"1":{"relayStatus":0}}}}
MD	W	http://192.168.1.97/api/slot/0/io/relay/1/relayStatus	{"Accept":"vdn.dac.v1","Content-Type":"application/json"}	PUT	{"slot":0,"io":{"relay":{"1":{"relayStatus":1}}}}
SOS	W	http://192.168.1.97/api/slot/0/io/relay/0/relayStatus	{"Accept":"vdn.dac.v1","Content-Type":"application/json"}	PUT	{"slot":0,"io":{"relay":{"0":{"relayStatus":1}}}}

Alla ricezione dell'allarme SOS viene attivato il primo relè, mentre per l'Uomo a Terra (MD) il secondo relè. Ovviamente è possibile configurare che alla ricezione di qualunque allarme venga attivato sempre lo stesso relè.Al riconoscimento di qualunque allarme (ACK) vengono disattivati i primi due relè.

Esempio configurazione per controllore WebRelay

Viene impostata anche l'autenticazione con password

company_id	signal_id	channel	url	headers	method	body	username	password
1	ACK	W	http://192.168.1.2/stateFull.xml?relayState=0		GET		none	webrelay
1	FALL	W	http://192.168.1.2/stateFull.xml?relayState=1		GET		none	webrelay
1	IMP	W	http://192.168.1.2/stateFull.xml?relayState=1		GET		none	webrelay
1	MD	W	http://192.168.1.2/stateFull.xml?relayState=1		GET		none	webrelay
1	MM	W	http://192.168.1.2/stateFull.xml?relayState=1		GET		none	webrelay
1	PB	W	http://192.168.1.2/stateFull.xml?relayState=1		GET		none	webrelay
1	SOS	W	http://192.168.1.2/stateFull.xml?relayState=1		GET		none	webrelay
1	VIMP	W	http://192.168.1.2/stateFull.xml?relayState=1		GET		none	webrelay

Esempio configurazione invio SMS con servizio MessageNet

Per invio SMS per allarme SOS

signal_id	channel	url	headers
SOS	W	https://api.messagenet.com/api/send_sms	{"Content-Type":"application/x-www-form-urlencoded"}

method	body	username	password
POST	auth_userid=#username#&auth_password=#password#&destination=+39335373944 &format=json&sender=eCoRoWeb&text=ALARM #AlarmStatus# #AlarmName# #DeviceId# #GoogleMap# (#GPS_date#) with instance Id #AlarmInstanceId#	<UserId>	<Password>

Esempio configurazione invio SMS con SMS Gateway

Per invio SMS per allarme SOS

signal_id	channel	url	headers	method	username	password
SOS	W	http://192.168.0.66/service.xml		POST	<UserId>	<Password>

body

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:pos="poseidonService.xsd"><soapenv:Header/><soapenv:Body><pos:QueueAdd><Queue>GsmOut</Queue><Gsm><Cmd>SMS</Cmd><Nmr>+3955555555</Nmr><Text>ALARM #AlarmStatus# #AlarmName# #DeviceId# #GoogleMap# (#GPS_date#) with instance Id #AlarmInstanceId#</Text></Gsm></pos:QueueAdd></soapenv:Body></soapenv:Envelope>
```

Esempio configurazione invio email

Per invio SMS per allarme SOS.

In url vanno indicati tutti i destinatari separati da virgola, mentre in headers va riportato l'oggetto dell'email. Nell'oggetto e nel corpo del messaggio possono essere utilizzati i campi variabili documentati nel manuale delle configurazioni.

signal_id	channel	url	headers
SOS	E	mailto:helpdesk@aldea.it,info@aldea.it	Alarm #AlarmName# #AlarmStatus# for device #DeviceId#

method	body	username	password
MAIL	ALARM #AlarmStatus# #AlarmName# #DeviceId# #GoogleMap# (#GPS_date#) with instance Id #AlarmInstanceId#		

Per poter inviare email deve essere correttamente configurato il PHP nella sezione [mail function]

Installazione su Linux

1. Update all system:
   sudo apt update
2. Install Apache httpd:
   sudo apt install apache2
3. Install PHP
   sudo apt install php libapache2-mod-php
4. Install PHP extensions:
   sudo apt install php-mbstring
   sudo sudo phpenmod mbstring
   sudo sudo phpenmod pdo_mysql
   sudo sudo phpenmod json
5. Restart Apache Httpd:
   sudo systemctl restart apache2
6. Install MySql:
   sudo apt install mysql-server
7. Configure MySql security:
   sudo mysql_secure_installation

If you are problem with first root login, reset root password :
https://devanswers.co/how-to-reset-mysql-root-password-ubuntu/

Installazione su Windows

1. Da httpd-2.4.41-win64-VS16.zip copiare la directory \Apache24 in C:\
2. Copiare la directory \Apache24 della cartella setup
3. Da una finestra DOS avviata come Amministratore eseguire i comandi per rendere Apache Http un servizio

```
cd \Apache24\bin

httpd.exe -k install -n "Apache HTTP Server"
```

4. Installare VC_Redist.exe e vcredist_x64.exe
5. Installare MySQL 5.5
6. Installare MySQL Workbench
7. Copiare la directory PHP_7_1_10 in C:\php (quest'ultima dir da creare)
8. Copiare il contenuto di eCoRo_vX.XX.zip in c:\Apache24\htdocs\eCoRoWeb (quest'ultima dir da creare)
9. Creare una directory che sarà il repository dei file di configurazione es. c:\eCoRoWeb\Repository\1 l'uno finale indica la company 1. Impostare questo percorso nella $configRepository del file c:\Apache24\htdocs\eCoRoWeb\config\ecoroConfig.inc.php
10. Copiare la directory AldeaInclude di AldeaInclude.zip in c:\Apache24\htdocs
11. Aggiungere nel PATH di sistema il percorso C:\php\PHP_7_1_10 per poter eseguire il php in ogni directory. Indispensabile per avviare il websocket server
12. Pianificare l'esecuzione dell'avvio del sistema del websocket server con il comando:

```
php bin\server.php
```

    impostando come directory di lavoro: C:\Apache24\htdocs\eCoRoWeb\websocket
13. Creare il database ecoro con lo script C:\Apache24\htdocs\eCoRoWeb\setup\ecoro_initial_database.sql
14. Personalizzare con utenza e password di mysql il file :
15. Avviare il servizio Apache HTTP Server
16. Poter notificare gli allarmi da VerticalMan è necessario creare un utenza in eCoRo Web con privilegi User che poi andrà inserita nella configurazione delle notifiche allarme e configurazione Web .
    E' sufficiente una unica utenza per tutti i dispositivi, ma la scelta è dell'amministratore, nel caso voglia creare tante utenze quanti sono i VerticalMan

Aggiornamento versione

Per aggiornare eCoRo Web è necessario copiare tutti i file contenuti nello zip della nuova versione ad eccezione della cartella config per evitare di sostituire la configurazione già personalizzata.

Nell'eventualità nella cartella db dello zip siano presenti degli script SQL devono essere eseguiti sequenzialmente come da nome file.

Se sono attivi il WebSocket server o il demon è necessario fermarli prima di eseguire l'aggiornamento.

Aggiornamento definizione chiavi configurazione

Nel caso venga rilasciato da Aldea un nuovo script SQL con la definizione delle nuove chiavi di configurazione, l'amministratore dopo l'esecuzione dello script dovrà eseguire le due procedure "Generate Preference" e "Generate Schema" dalla console Web nella sezione Admin.

Controllo salute sistema

Per controllare lo stato di salute di tutti i componenti dell'applicazione eCoRo Web è possibile richiamare periodicamente la pagina **checkHealth** che ritorna un json con le seguenti chiavi:

Chiave	Descrizione
responseCode	Risultato di tutte le verifiche 0 - Tutto OK 1 - Errore generico (Exception) 2 - Database error 3 - WebSocket Server error 4 - eCoRo Demon non attivo 5 - LDAP error
responseDescription	OK oppure descrizione dell'errore
databaseHealth	json dettaglio salute database con chiavi status, description, relevance e action
webSocketServerHealth	json dettaglio salute WebSocketServer con chiavi status, description, relevance e action. I controlli sul WebSocket Server vengono eseguiti solo se il WebSocket Server è configurato in ecoroConfig.
deamonHealth	json dettaglio salute Deamon con chiavi status, description, relevance e action. I controlli sull'esecuzione dell'eCoRo Deamon vengono effettuati solamente se è presente almeno un dispositivo da controllare come raggiungibile (colonna checkReachable = true in ecoro_device)
ldapHealth	json di dettaglio connessione ldap con chiavi status, description, relevance e action

Descrizione chiavi dei json di dettaglio

Chiave	Descrizione
status	Riporta OK o KO
description	Descrizione dell'errore
relevance	Rilevanza della segnalazione : High,Medium,Low
action	Eventuale prima azione da intraprendere per risolvere l'errore

Esempio di ritorno Json con errore di connessione a ldap server e demone non attivo:


```
{
   "responseCode":"5",
   "responseDescription":"LDAP error",
   "databaseHealth":{
      "relevance":"High",
      "status":"OK",
      "description":"OK"
   },
   "webSocketServerHealth":{
      "relevance":"High",
      "status":"OK",
      "description":"OK"
   },
   "ldapHealth":{
      "relevance":"High",
      "status":"KO",
      "description":"-Error connecting to LDAP: No additional information is available.",
      "action":"Check if eCoRo Web server can reach ldap://192.168.1.43:389 or if user prova have the correct password in ecoroConfig.inc.php"
   },
   "deamonHealth":{
      "relevance":"Medium",
      "status":"KO",
      "description":"eCoRo Deamon not running",
      "action":"Check if the eCoRo Deamon is started"
   }

}
```