

Prior to resolve any AEM application issue we should check that:

- Required bundles are up and running;
- Related web services endpoint are available;
- Required resources and appropriate content structure exists;
- And so on ...

Of course all above steps can be done manually from time to time but we need to run these validations automatically which help to monitor AEM application at a glance.

To not reinvent the wheel we can use OOTB Apache Sling Health Checks (below simply HC).

What is this?

HC is an OSGI service that implements **org.apache.sling.hc.api.HealthCheck** and returns **Result** according to validate conditions:

```
public interface HealthCheck {
    public Result execute();
}
```

Result is a simple immutable class that provides a Status (**OK**, **WARN**, **CRITICAL** and etc.) and one or more log-like messages for additional info.

Note: if you set any Result log message it will be identified within AEM as WARN. Therefore we should define as :

```
new Result(Result.Status.OK, "Some Message")
```

How to configure global "HC Executor" options ?

AEM is a modular system therefore we have a couple of configuration places for every piece of the system, one of this - **HC Executor** service which can be configured from **"/system/console/configMgr/org.apache.sling.hc.core.impl.executor.HealthCheckExecutorImpl"**:

The screenshot shows the configuration page for the Apache Sling Health Check Executor. The page title is "/system/console/configMgr/org.apache.sling.hc.core.impl.executor.HealthCheckExecutorImpl". The configuration is for the "Apache Sling Health Check Executor" service, which runs health checks for a given list of tags in parallel. The configuration includes the following fields:

- Timeout:** 2000 (Timeout in ms until a check is marked as timed out (timeoutInMs))
- Timeout threshold for CRITICAL:** 300000 (Threshold in ms until a check is marked as 'exceedingly' timed out and will be marked CRITICAL instead of WARN only (longRunningFutureThresholdForCriticalMs))
- Results Cache TTL in Ms:** 2000 (Result Cache time to live - results will be cached for the given time (resultCacheTtlInMs))

Configuration Information:

- Persistent Identity (PID):** org.apache.sling.hc.core.impl.executor.HealthCheckExecutorImpl
- Configuration Binding:** Unbound or new configuration

Buttons: Cancel, Reset, Delete, Unbind, Save

Note: every HC executes by HC Executor within **Sling Thread pool** which guarantees that we have only one single running instance at a time.

How to implement custom HC?

For implementation individual HC initially we should implements `org.apache.sling hc.api.HealthCheck` and specify options for service like below:

```
@Component(metatype = true)
@Properties({
    @Property(name = HealthCheck.NAME,value = "HCName"),
    @Property(name = HealthCheck.TAGS,value = {"meetup"}),
    @Property(name = HealthCheck.MBEAN_NAME,value = "HCName")
})
@Service(value = {HealthCheck.class})
public class IncorrectLocalhostHC implements HealthCheck {
```

If HC need to be executed by scheduler (once a day for example) we can specify schedule intervale within properties:

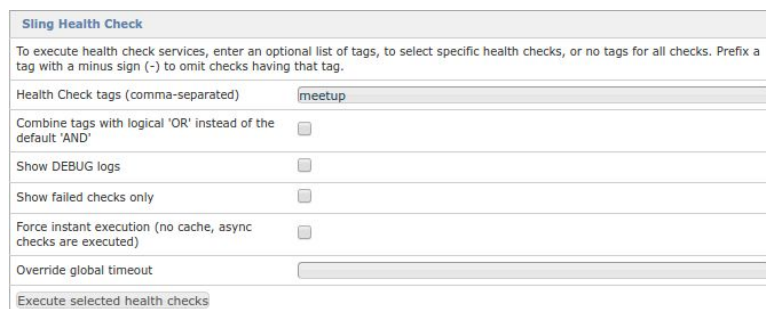
```
@Property(name = HealthCheck.ASYNC_CRON_EXPRESSION, value = "0 0 12 1/1 * ? *")
```

Also from version of Sling HC core 1.2.6 there will be a new property `hc.resultCacheTtlInMs` which overrides the global default TTL as configured in HC executor for HC responses.

HC can be also configured with annotation `@SlingHealthCheck` but this not working OOTB in AEM 6.1 :

```
@SlingHealthCheck(
    name="Health Check Name For Felix Console",
    mbeanName="JMX Name",
    description="Health Check Description",
    tags={"meetup"}
)
```

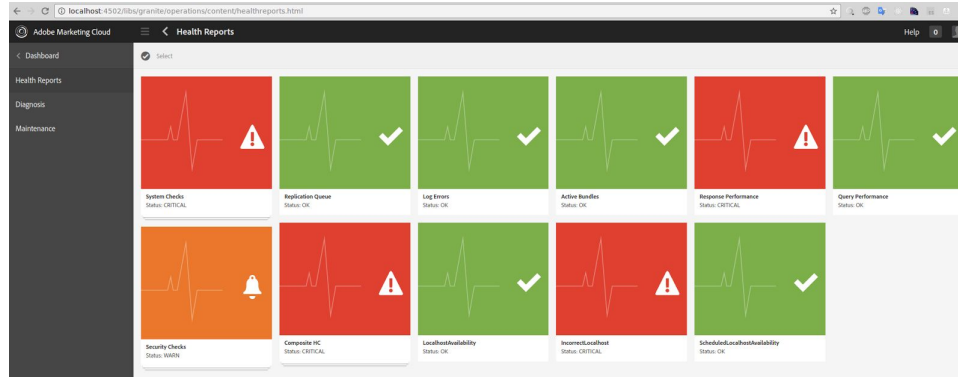
So this simple steps allow to implement HC which can be executed from Felix console over the path `/system/console/healthcheck`:



The screenshot shows the 'Sling Health Check' web interface. It includes a title bar, a descriptive paragraph, a text input field for 'Health Check tags (comma-separated)' containing 'meetup', several checkboxes for options like 'Combine tags with logical OR', 'Show DEBUG logs', 'Show failed checks only', and 'Force instant execution', and a text input field for 'Override global timeout'. At the bottom is a button labeled 'Execute selected health checks'.

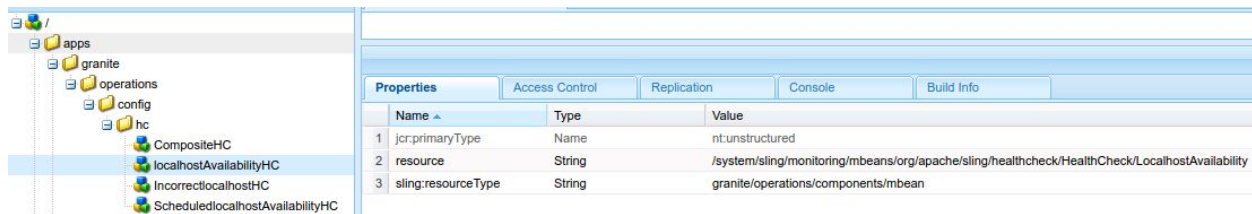
How to display HC on “Operations Dashboard” ?

If we will open **Tools -> Operations -> Dashboard -> Console -> Health Reports** (or from path `/libs/granite/operations/content/healthreports.html`) we will see cards with HC.



So to add a custom HC as a card on this dashboard we need to create node under the path [/apps/granite/operations/config/hc](#) with properties:

- **resource**{String} - `/system/sling/monitoring/mbeans/org/apache/sling/healthcheck/HealthCheck/[hc.name]`
- **sling:resourceType**{String} - `granite/operations/components/mbean`



Operations dashboard also allow to merge HC cards into the groups (as it's already done for "System Checks" and "Security Checks"). All composite HC configurations locates within **org.apache.sling.hc.core.impl.CompositeHealthCheck** factory, so to add our custom HC here we should add new configuration to this factory from Felix console or from configuration file. In case of configuration from Felix console we need to specify "Name" for HC group and "Filter Tags" so all HC with this tags will be available under this composite HC card on Operations dashboard

