adaptTo()

APACHE SLING & FRIENDS TECH MEETUP

2 - 4 SEPTEMBER 2019

~~Sling~~ Felix Health Checks

Georg Henzler, netcentric

A Cognizant Digital Business

adaptTo()

Health Checks - Recap

How can you trust your current (fragile?) combination of OSGi bundles and configs?

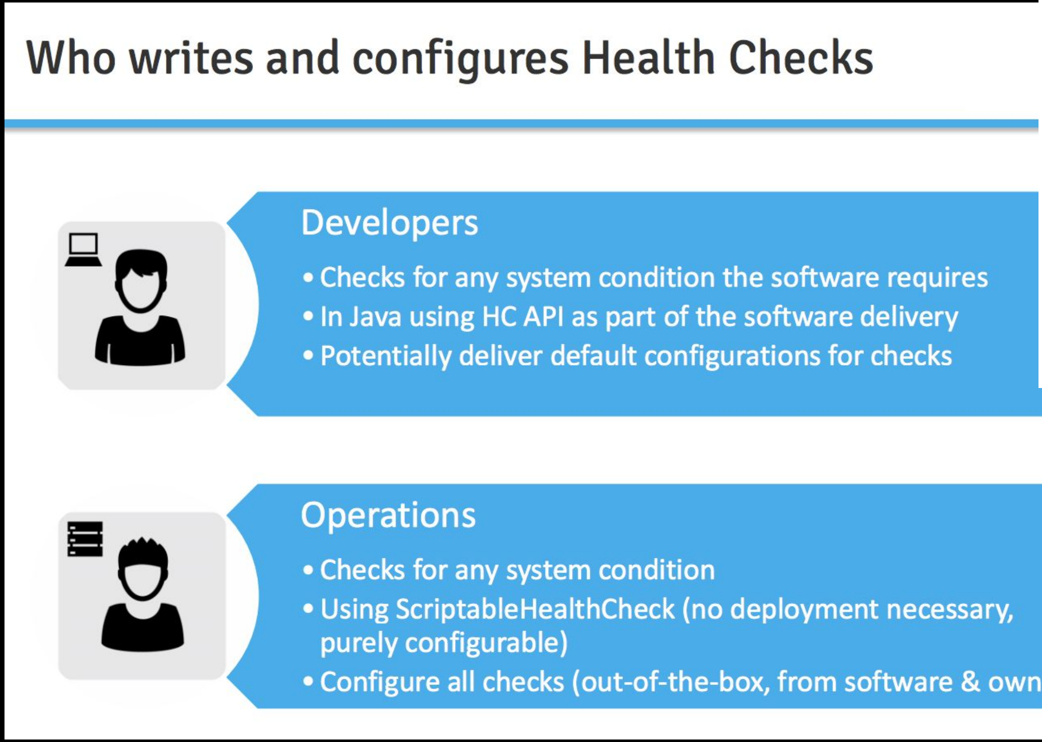
It needs live TESTING.

HealthCheck APIs

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

public class Result {
   public boolean isOk() ...
   public Status getStatus() ...
   public Iterator<ResultLog.Entry> iterator() ...
}
```

HealthCheck OSGi services

Service props: name, MBean, tags



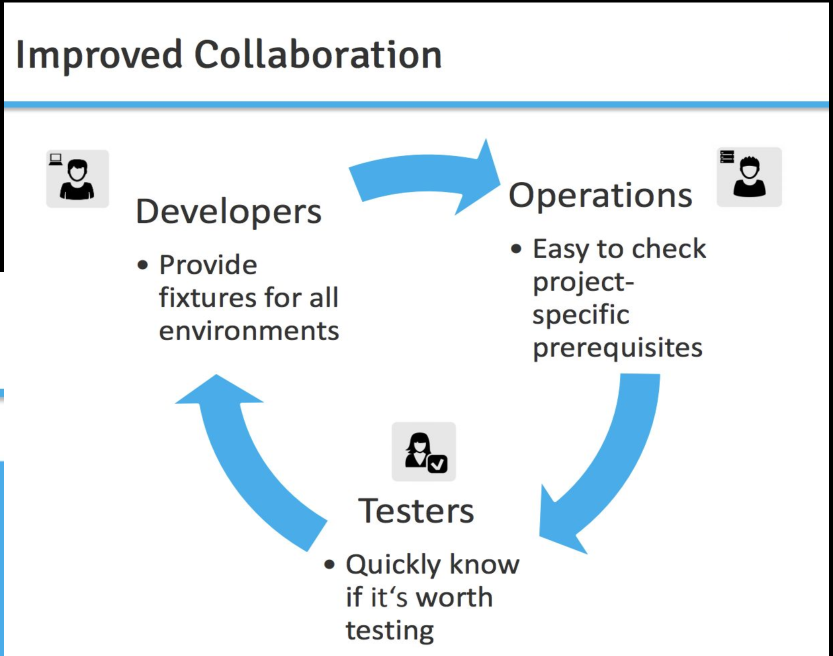
Who writes and configures Health Checks

Developers

- Checks for any system condition the software requires
- In Java using HC API as part of the software delivery
- Potentially deliver default configurations for checks

Operations

- Checks for any system condition
- Using ScriptableHealthCheck (no deployment necessary, purely configurable)
- Configure all checks (out-of-the-box, from software & own)



Improved Collaboration

Developers

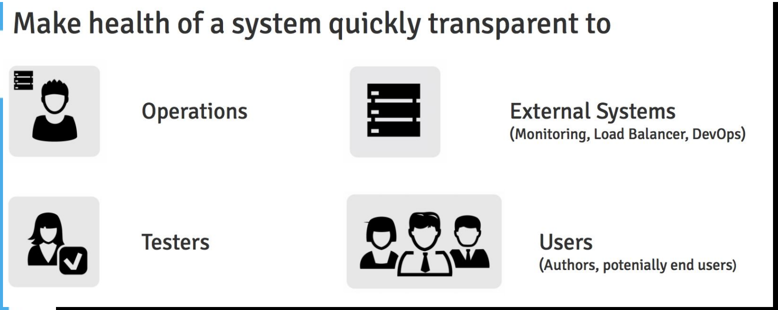
- Provide fixtures for all environments

Operations

- Easy to check project-specific prerequisites

Testers

- Quickly know if it‘s worth testing



Make health of a system quickly transparent to

Operations	External Systems (Monitoring, Load Balancer, DevOps)
Testers	Users (Authors, potenially end users)

Health Check Executor

… a container for Health Checks

VARNISH SOFTWARE

NGINX

HAPROXY COMMUNITY EDITION

Nagios®

HCs are widely configurable today

Consul

EXPERIENCE MANAGER DISPATCHER

adaptTo()

Why move?

If you start the heavy lifting...

Photo by Théo Dorp on Unsplash

(often) **Years later…**

Photo by chuttersnap on Unsplash

Why move to Felix? (anyway)

- Sling HCs with many dependencies to Sling
  - mostly unnecessary
- Make Health Checks available to a broader audience
- Avoid duplication with SystemReady
- Be able to improve Sling startup/shutdown behaviour

Also with every move: Potential for cleanup!

Photo by Josh Howard on Unsplash

Cleanup

Status Values

Status	System is usable	Meaning
OK	yes	Everything is ok.
WARN	yes	Tendency to CRITICAL - Take action!
TEMPORARILY_UNAVAILABLE	no	Tendency to OK - Wait!
CRITICAL	no	System is not functional and must not be used
HEALTH_CHECK_ERROR	no	Actual status unknown

… more cleanup

- Removed status values DEBUG & INFO
- OSGi R7 component property type
    - @HealthCheckService
- Use tag "default" if no tag is provided
- Better name for JMX Bean AdjustableStatusHealthCheck
- Migrated all systemready checks to Felix Health Checks Runtime

adaptTo()

New Features

New Features

- General Checks
  - Bundles active, disk space, memory, CPU, threads
  - JMX Attributes, Http Requests, Scripted Check
- Auto Logging with category prefix "healthcheck."
- Filters
  - Service Unavailable Filter
  - AdhocResultDuringRequestProcessingFilter

How to migrate

Migrate a Runtime

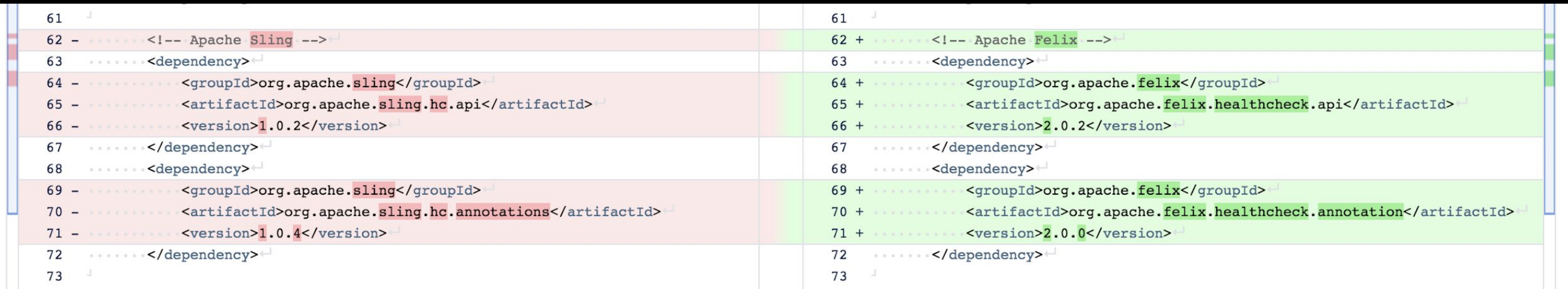
- New runtimes, e.g. Sling Starter
  - Use Felix HC bundles (api/core/webconsoleplugin/generalchecks)
  - Optional: Add Sling HC API if bundles with that dependency exist (core/webconsole not needed, keep hc.support)
- Existing runtimes, e.g. AEM 6.4/6.5
  - Use migration kit (see References at the end)
  - Sling HC API and some CA Configs for AEM UI are needed
- ⚠️ JMX Paths have changed

Migrate your Checks - Step by Step

- Adjust Maven dependencies
- "Organise Imports" in your IDE
- Replace @SlingHealthCheck with @HealthCheckService
  - … and maybe @Async, @ResultTTL, @HealthCheckMBean, @Sticky
- Using cron expressions requires quartz bundle
- Sticky checks specify stickiness in seconds now
- Replace potential usages of Status.DEBUG

Migrate your Checks

- Maven Dependencies:

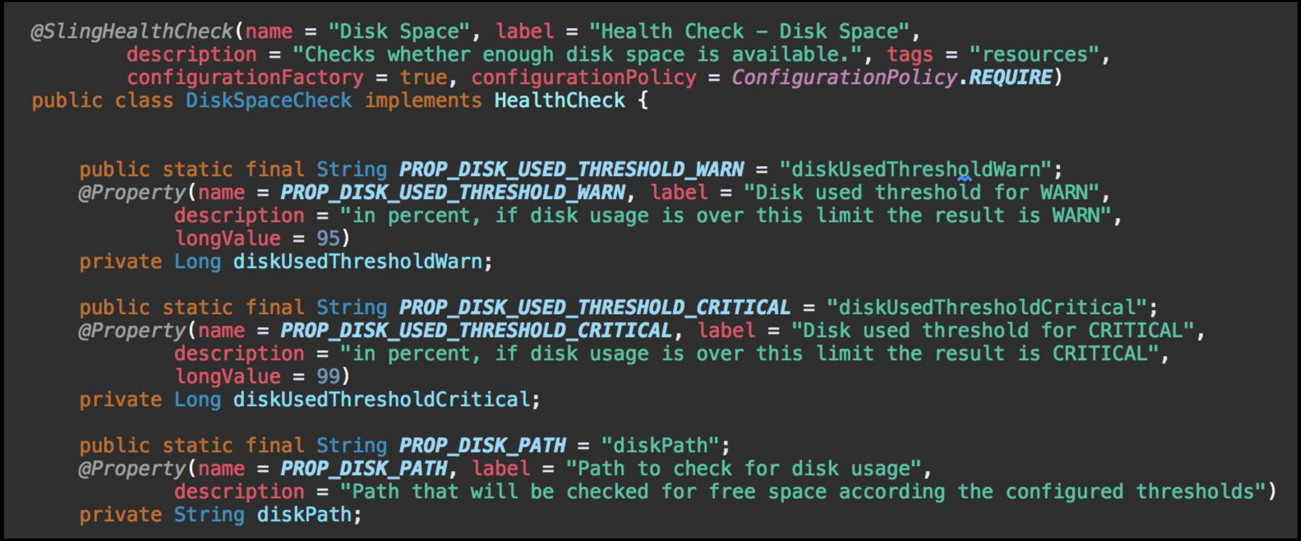


```
 
-    <!-- Apache Sling -->
     <dependency>
-        <groupId>org.apache.sling</groupId>
-        <artifactId>org.apache.sling.hc.api</artifactId>
-        <version>1.0.2</version>
     </dependency>
     <dependency>
-        <groupId>org.apache.sling</groupId>
-        <artifactId>org.apache.sling.hc.annotations</artifactId>
-        <version>1.0.4</version>
     </dependency>
 
```

```
 
+    <!-- Apache Felix -->
     <dependency>
+        <groupId>org.apache.felix</groupId>
+        <artifactId>org.apache.felix.healthcheck.api</artifactId>
+        <version>2.0.2</version>
     </dependency>
     <dependency>
+        <groupId>org.apache.felix</groupId>
+        <artifactId>org.apache.felix.healthcheck.annotation</artifactId>
+        <version>2.0.0</version>
     </dependency>
 
```

Sling Version (Old School)

- Using @SlingHealthCheck and Service Interface org.apache.sling.hc.api.HealthCheck

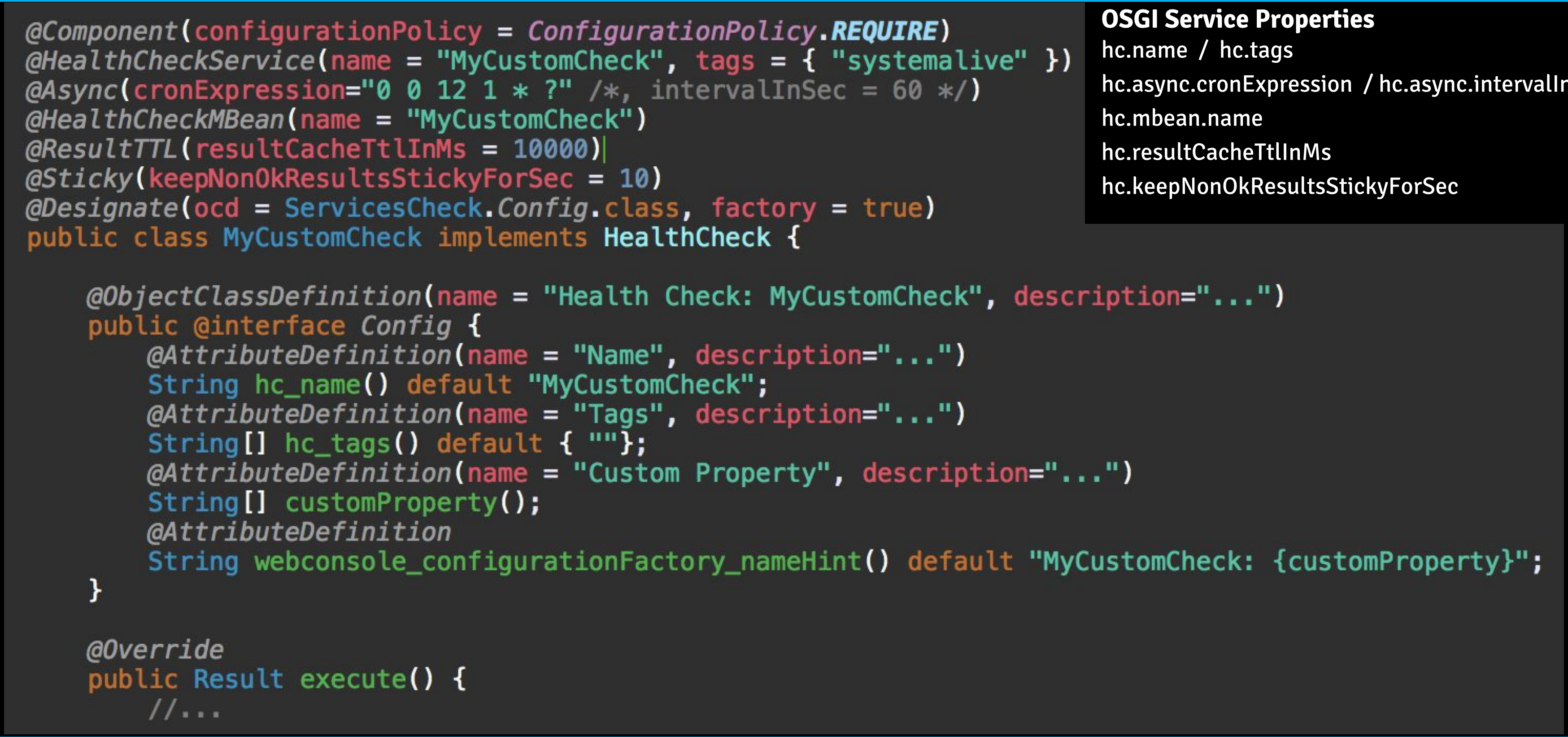


```
@SlingHealthCheck(name = "Disk Space", label = "Health Check - Disk Space",
        description = "Checks whether enough disk space is available.", tags = "resources",
        configurationFactory = true, configurationPolicy = ConfigurationPolicy.REQUIRE)
public class DiskSpaceCheck implements HealthCheck {


    public static final String PROP_DISK_USED_THRESHOLD_WARN = "diskUsedThresholdWarn";
    @Property(name = PROP_DISK_USED_THRESHOLD_WARN, label = "Disk used threshold for WARN",
            description = "in percent, if disk usage is over this limit the result is WARN",
            longValue = 95)
    private Long diskUsedThresholdWarn;

    public static final String PROP_DISK_USED_THRESHOLD_CRITICAL = "diskUsedThresholdCritical";
    @Property(name = PROP_DISK_USED_THRESHOLD_CRITICAL, label = "Disk used threshold for CRITICAL",
            description = "in percent, if disk usage is over this limit the result is CRITICAL",
            longValue = 99)
    private Long diskUsedThresholdCritical;

    public static final String PROP_DISK_PATH = "diskPath";
    @Property(name = PROP_DISK_PATH, label = "Path to check for disk usage",
            description = "Path that will be checked for free space according the configured thresholds")
    private String diskPath;
```

adaptTo()

Felix Health Check (The New Way™)

netcentric
A Cognizant Digital Business



```
@Component(configurationPolicy = ConfigurationPolicy.REQUIRE)
@HealthCheckService(name = "MyCustomCheck", tags = { "systemalive" })
@Async(cronExpression="0 0 12 1 * ?" /*, intervalInSec = 60 */)
@HealthCheckMBean(name = "MyCustomCheck")
@ResultTTL(resultCacheTtlInMs = 10000)
@Sticky(keepNonOkResultsStickyForSec = 10)
@Designate(ocd = ServicesCheck.Config.class, factory = true)
public class MyCustomCheck implements HealthCheck {

    @ObjectClassDefinition(name = "Health Check: MyCustomCheck", description="...")
    public @interface Config {
        @AttributeDefinition(name = "Name", description="...")
        String hc_name() default "MyCustomCheck";
        @AttributeDefinition(name = "Tags", description="...")
        String[] hc_tags() default { ""};
        @AttributeDefinition(name = "Custom Property", description="...")
        String[] customProperty();
        @AttributeDefinition
        String webconsole_configurationFactory_nameHint() default "MyCustomCheck: {customProperty}";
    }

    @Override
    public Result execute() {
        //...
```

OSGI Service Properties

hc.name / hc.tags

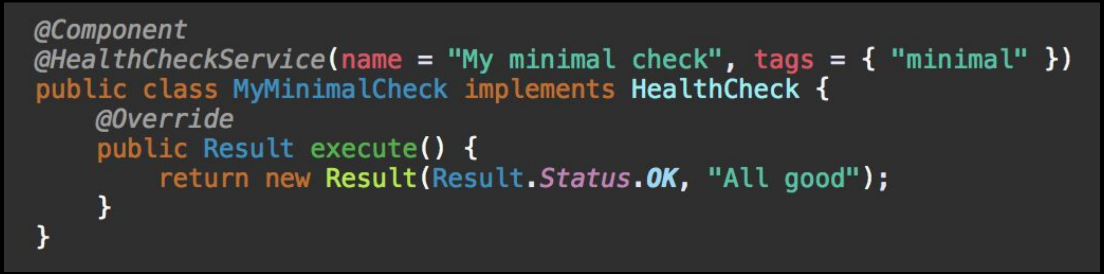
hc.async.cronExpression / hc.async.intervalInSec

hc.mbean.name

hc.resultCacheTtlInMs

hc.keepNonOkResultsStickyForSec

Felix Health Check (The Minimal Way™)



```
@Component
@HealthCheckService(name = "My minimal check", tags = { "minimal" })
public class MyMinimalCheck implements HealthCheck {
    @Override
    public Result execute() {
        return new Result(Result.Status.OK, "All good");
    }
}
```

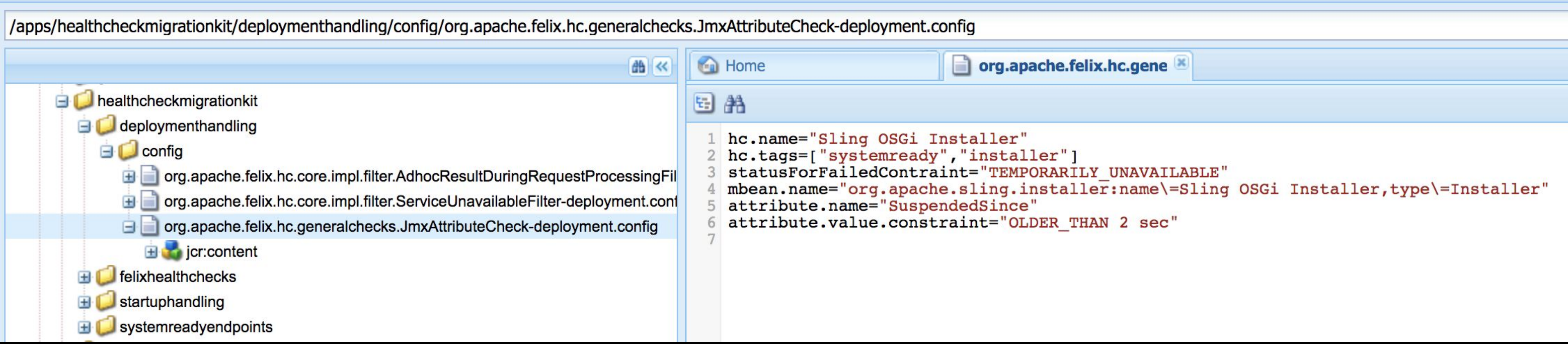
adaptTo()

OOTB Configurable Checks

OOTB Configurable Checks

- In bundle o.a.felix.healthchecks.generalchecks:
  - JMX Attributes, Http Requests, Services, Components, etc.
  - Scripted Check
    - "JCR-enabled version" in o.a.sling.hc.support
- SimpleConstraintChecker
  - See JUnit test for full reference
- Flexibility for Operations!

JMX Health Check

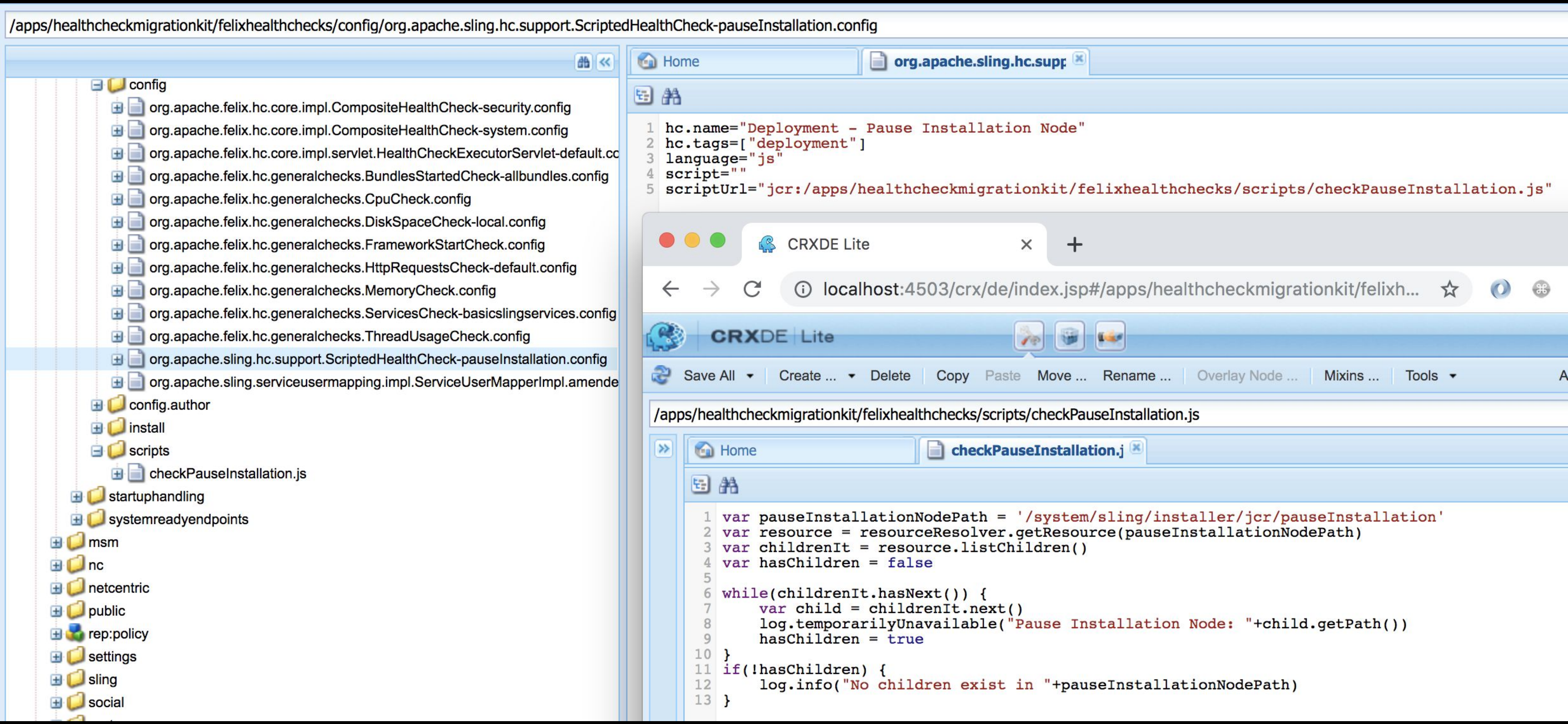


/apps/healthcheckmigrationkit/deploymenthandling/config/org.apache.felix.hc.generalchecks.JmxAttributeCheck-deployment.config

healthcheckmigrationkit
deploymenthandling
config
org.apache.felix.hc.core.impl.filter.AdhocResultDuringRequestProcessingFil
org.apache.felix.hc.core.impl.filter.ServiceUnavailableFilter-deployment.conf
org.apache.felix.hc.generalchecks.JmxAttributeCheck-deployment.config
jcr:content
felixhealthchecks
startuphandling
systemreadyendpoints
Home
org.apache.felix.hc.gene

```
hc.name="Sling OSGi Installer"
hc.tags=["systemready","installer"]
statusForFailedContraint="TEMPORARILY_UNAVAILABLE"
mbean.name="org.apache.sling.installer:name\=Sling OSGi Installer,type\=Installer"
attribute.name="SuspendedSince"
attribute.value.constraint="OLDER_THAN 2 sec"
```

Scripted Health Check



/apps/healthcheckmigrationkit/felixhealthchecks/config/org.apache.sling.hc.support.ScriptedHealthCheck-pauseInstallation.config

```
config
    org.apache.felix.hc.core.impl.CompositeHealthCheck-security.config
    org.apache.felix.hc.core.impl.CompositeHealthCheck-system.config
    org.apache.felix.hc.core.impl.servlet.HealthCheckExecutorServlet-default.cc
    org.apache.felix.hc.generalchecks.BundlesStartedCheck-allbundles.config
    org.apache.felix.hc.generalchecks.CpuCheck.config
    org.apache.felix.hc.generalchecks.DiskSpaceCheck-local.config
    org.apache.felix.hc.generalchecks.FrameworkStartCheck.config
    org.apache.felix.hc.generalchecks.HttpRequestsCheck-default.config
    org.apache.felix.hc.generalchecks.MemoryCheck.config
    org.apache.felix.hc.generalchecks.ServicesCheck-basicslingservices.config
    org.apache.felix.hc.generalchecks.ThreadUsageCheck.config
    org.apache.sling.hc.support.ScriptedHealthCheck-pauseInstallation.config
    org.apache.sling.serviceusermapping.impl.ServiceUserMapperImpl.amende
config.author
install
scripts
    checkPauseInstallation.js
startuphandling
systemreadyendpoints
msm
nc
netcentric
public
rep:policy
settings
sling
social
```

```
hc.name="Deployment - Pause Installation Node"
hc.tags=["deployment"]
language="js"
script=""
scriptUrl="jcr:/apps/healthcheckmigrationkit/felixhealthchecks/scripts/checkPauseInstallation.js"
```

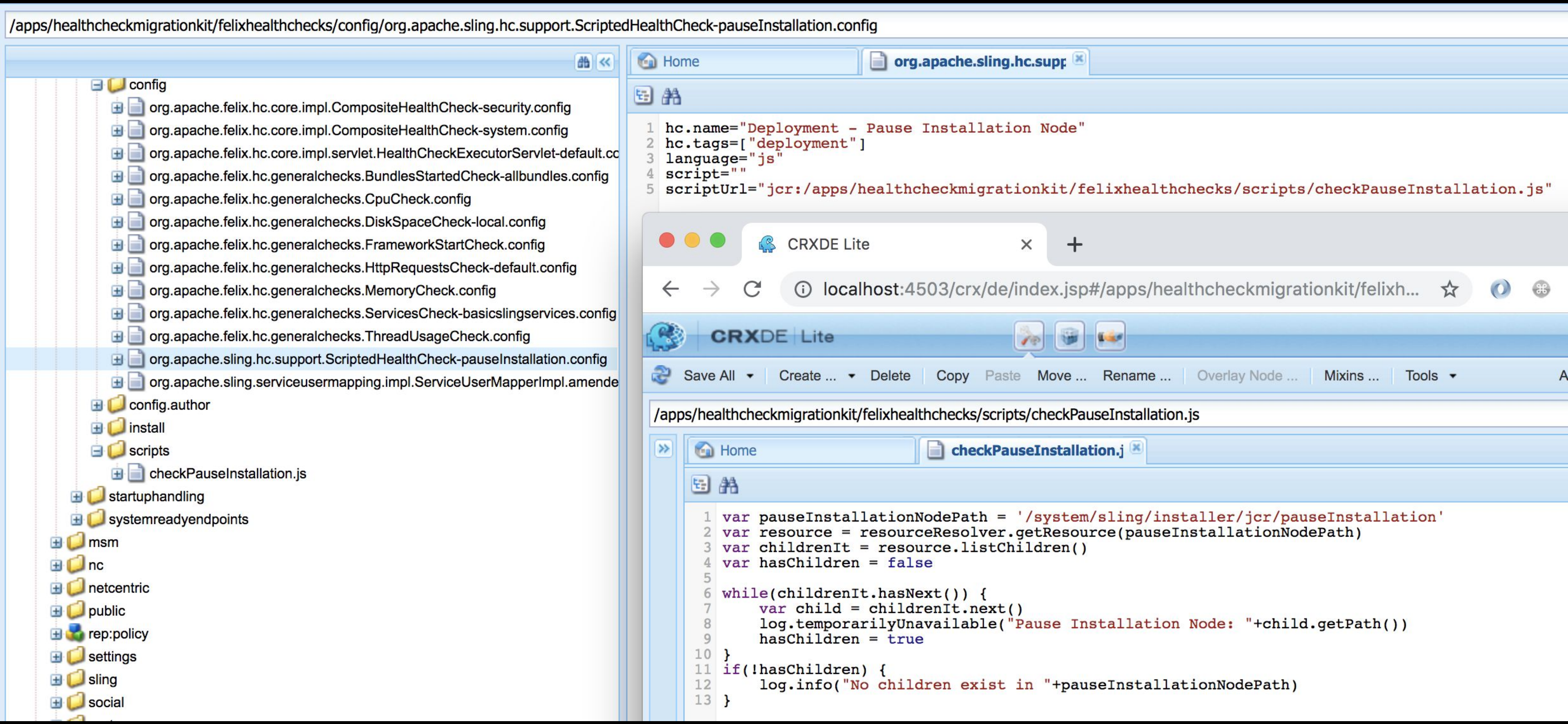
/apps/healthcheckmigrationkit/felixhealthchecks/scripts/checkPauseInstallation.js

```
var pauseInstallationNodePath = '/system/sling/installer/jcr/pauseInstallation'
var resource = resourceResolver.getResource(pauseInstallationNodePath)
var childrenIt = resource.listChildren()
var hasChildren = false

while(childrenIt.hasNext()) {
    var child = childrenIt.next()
    log.temporarilyUnavailable("Pause Installation Node: "+child.getPath())
    hasChildren = true
}
if(!hasChildren) {
    log.info("No children exist in "+pauseInstallationNodePath)
}
```

adaptTo()

Scripted Health Check

netcentric
A Cognizant Digital Business



/apps/healthcheckmigrationkit/felixhealthchecks/config/org.apache.sling.hc.support.ScriptedHealthCheck-pauseInstallation.config

- config
  - org.apache.felix.hc.core.impl.CompositeHealthCheck-security.config
  - org.apache.felix.hc.core.impl.CompositeHealthCheck-system.config
  - org.apache.felix.hc.core.impl.servlet.HealthCheckExecutorServlet-default.cc
  - org.apache.felix.hc.generalchecks.BundlesStartedCheck-allbundles.config
  - org.apache.felix.hc.generalchecks.CpuCheck.config
  - org.apache.felix.hc.generalchecks.DiskSpaceCheck-local.config
  - org.apache.felix.hc.generalchecks.FrameworkStartCheck.config
  - org.apache.felix.hc.generalchecks.HttpRequestsCheck-default.config
  - org.apache.felix.hc.generalchecks.MemoryCheck.config
  - org.apache.felix.hc.generalchecks.ServicesCheck-basicslingservices.config
  - org.apache.felix.hc.generalchecks.ThreadUsageCheck.config
  - org.apache.sling.hc.support.ScriptedHealthCheck-pauseInstallation.config
  - org.apache.sling.serviceusermapping.impl.ServiceUserMapperImpl.amende
- config.author
- install
- scripts
  - checkPauseInstallation.js
- startuphandling
- systemreadyendpoints
- msm
- nc
- netcentric
- public
- rep:policy
- settings
- sling
- social

Home | org.apache.sling.hc.supp

```
hc.name="Deployment - Pause Installation Node"
hc.tags=["deployment"]
language="js"
script=""
scriptUrl="jcr:/apps/healthcheckmigrationkit/felixhealthchecks/scripts/checkPauseInstallation.js"
```

CRXDE Lite

localhost:4503/crx/de/index.jsp#/apps/healthcheckmigrationkit/felixh...

/apps/healthcheckmigrationkit/felixhealthchecks/scripts/checkPauseInstallation.js

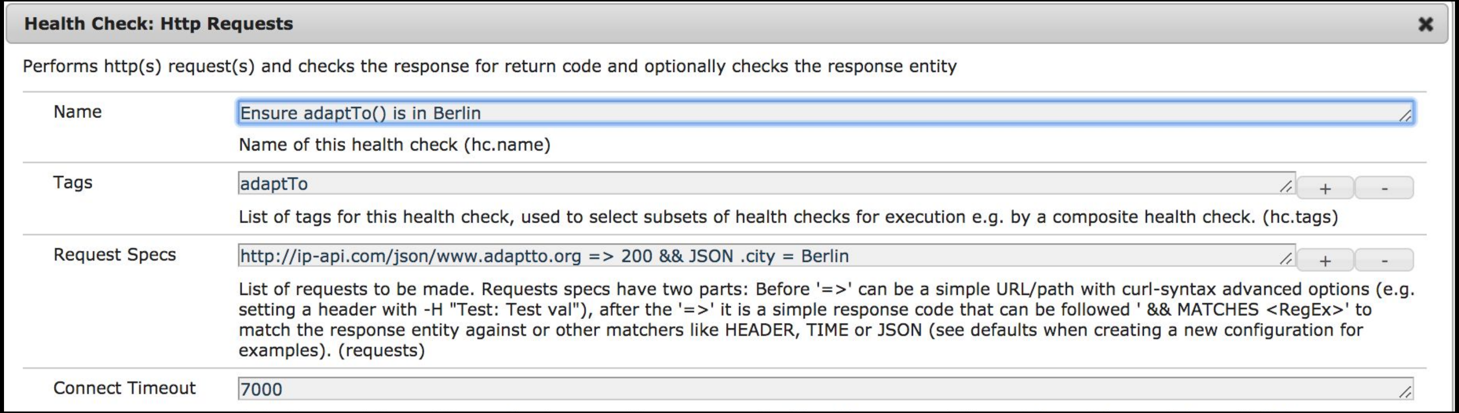
Home | checkPauseInstallation.j

```
var pauseInstallationNodePath = '/system/sling/installer/jcr/pauseInstallation'
var resource = resourceResolver.getResource(pauseInstallationNodePath)
var childrenIt = resource.listChildren()
var hasChildren = false

while(childrenIt.hasNext()) {
    var child = childrenIt.next()
    log.temporarilyUnavailable("Pause Installation Node: "+child.getPath())
    hasChildren = true
}
if(!hasChildren) {
    log.info("No children exist in "+pauseInstallationNodePath)
}
```

adaptTo()

Http Requests Health Check

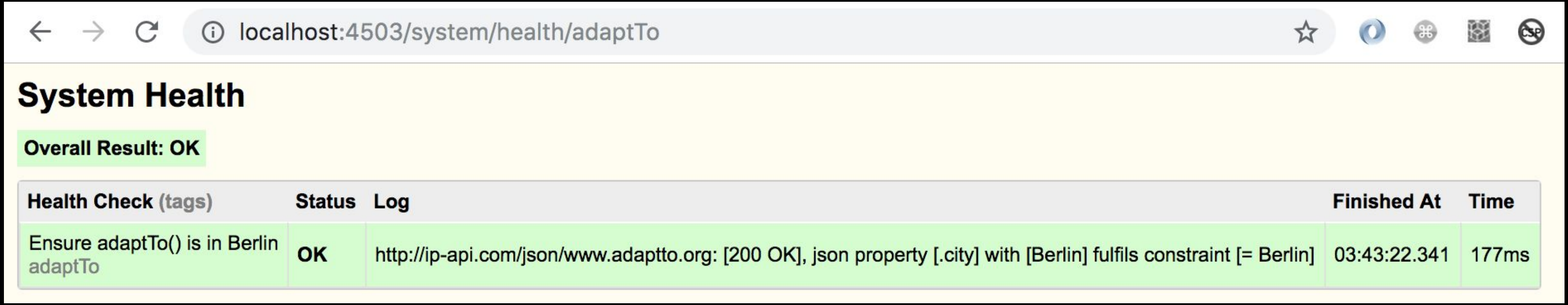
netcentric
A Cognizant Digital Business



Health Check: Http Requests ✖

Performs http(s) request(s) and checks the response for return code and optionally checks the response entity

Name	Ensure adaptTo() is in Berlin Name of this health check (hc.name)
Tags	adaptTo [+] [-] List of tags for this health check, used to select subsets of health checks for execution e.g. by a composite health check. (hc.tags)
Request Specs	http://ip-api.com/json/www.adaptto.org => 200 && JSON .city = Berlin [+] [-] List of requests to be made. Requests specs have two parts: Before '=>' can be a simple URL/path with curl-syntax advanced options (e.g. setting a header with -H "Test: Test val"), after the '=>' it is a simple response code that can be followed ' && MATCHES <RegEx>' to match the response entity against or other matchers like HEADER, TIME or JSON (see defaults when creating a new configuration for examples). (requests)
Connect Timeout	7000



localhost:4503/system/health/adaptTo

System Health

Overall Result: OK

Health Check (tags)	Status	Log	Finished At	Time
Ensure adaptTo() is in Berlin adaptTo	OK	http://ip-api.com/json/www.adaptto.org: [200 OK], json property [.city] with [Berlin] fulfils constraint [= Berlin]	03:43:22.341	177ms

Startup/Shutdown

Status 2018

- Responses to arbitrary requests
  - Sling: 503 via sling-starter-startup (only startup)
  - AEM: 503 via sling-startupfilter-disabler (only startup)
- Querying system status during startup/shutdown
  - Not possible with Sling HCs (only start at start level 20)
  - System Ready to close that gap
- Inconsistent behaviour

Startup/Shutdown with Felix HCs

- Responses to arbitrary requests
  - 503 via SystemUnavailableFilter (startup & shutdown)
- Querying system status during startup/shutdown
  - Via Felix HCs (active from start level 5 along with servlet engine)
- Advantages
  - Consistent behaviour
  - Extensible & Configurable

adaptTo()

Filters

(more use cases)

Deployment

- **ServiceUnavailableFilter**
  - "Returns a 503 Service Unavailable Page if certain tags are in non-ok result"
  - For sling context of http whiteboard and if tag "deployment" is in TEMPORARILY_UNAVAILABLE
- **AdhocResultDuringRequestProcessingFilter**
  - "Dynamically adds a HC result for configured requests"
  - For requests to package manager, add a tag "deployment"

References

References

- Felix Health Checks Documentation
  - https://github.com/apache/felix/blob/trunk/healthcheck/README.md
- Migration from Sling
  - https://sling.apache.org/documentation/bundles/sling-health-check-tool.html#migrate-custom-checks
  - https://sling.apache.org/documentation/bundles/sling-health-check-tool.html#migrate-a-runtime
- Migration Kit for AEM
  - https://github.com/Netcentric/healthcheck-migration-kit

adaptTo()

Q&A