





APACHE SLING & FRIENDS TECH MEETUP
10-12 SEPTEMBER 2018

System Ready framework

Christian Schneider - Adobe, Apache

Andrei Dulvac - Adobe, Apache

The speakers

- Christian Schneider
 - Adobe Switzerland
 - Apache
 - @schneider_chris 
- Andrei Dulvac
 - Adobe Switzerland
 - Apache
 - @adulvac 

When is an *OSGi application* ready? (1)

- Why do you need to know?
 - When can automated tests start accessing your system
 - When is your application ready to receive traffic? (zero downtime deployments)

When is an *OSGi application* ready? (2)

- What could *ready* mean?
 - Particular OSGi services are active
 - Oak initial indexing is done/ search is available
 - Critical async startup jobs are finished (e.g. LDAP users migration)
 - For complex applications – need **app/deployment -specific checks**

Use cases

- Kubernetes
 - Provides HTTP readiness and liveness checks
- PaxExam
 - Waiting for a system in integration tests
 - Root-cause analysis – why did a system not start?
- End-to-end automated tests
 - Waiting for the application to be ready before starting the tests

Felix systemready

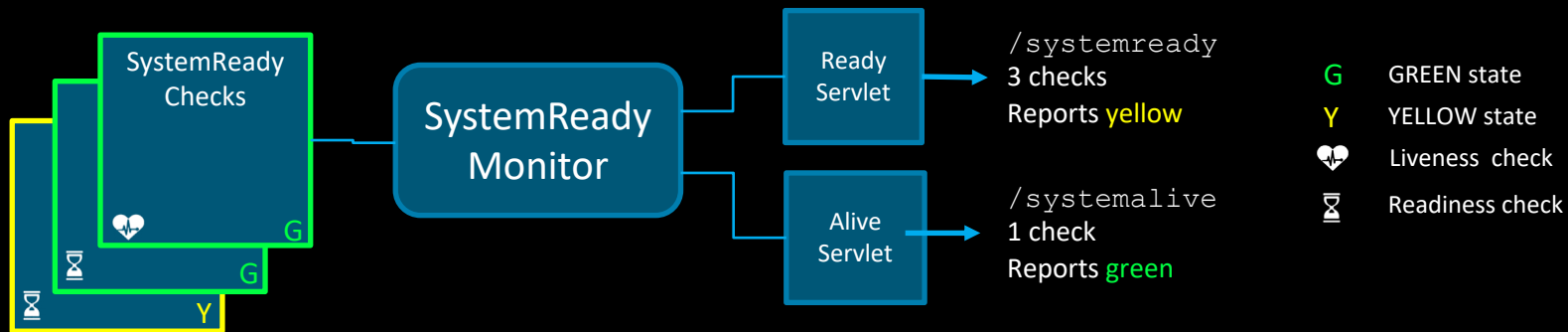
- 49KB OSGi bundle in Felix, framework-agnostic, very little dependencies
- Ready checks
 - Framework started
 - List of services present
 - List of DS components present
 - Extensible by application-specific checks (SPI)
- Aggregated system status for liveness and readiness - GREEN/ YELLOW/ RED
- Built-in HTTP servlets – 200 for GREEN, 503 otherwise
- Root-cause analysis for missing components

How the default servlets output looks like

```
HTTP RESPONSE CODE: 503
```

```
{"systemStatus": "YELLOW",  
"checks": [  
  {  
    "check": "Framework Start Ready Check",  
    "status": "GREEN",  
    "details": "Framework started. Start level: 30"  
  },  
  {  
    "check": "My custom check",  
    "status": "YELLOW",  
    "details": "I still have 2/5 startup whatchamacallits to do"  
  }  
]  
}
```

Felix systemready architecture



SHOW ME!

Demo layout

- AEM jar deployment
- Kubernetes – OSGi web microservice example
- Kubernetes – AEM deployment
- Q & A