



SPRING FRAMEWORK

INITIALIZE & DESTROY

APRIL, 2015

FileEventLogger

```
- filename: String;  
+ logEvent(...);  // appends to file
```

FileEventLogger

```
- filename: String;  
+ logEvent(...); // appends to file
```

Use **FileUtils** from **commons-io**:

- `<groupId>commons-io</groupId>`
`<artifactId>commons-io</artifactId>`
`<version>2.4</version>`
- `FileUtils.writeStringToFile(File file,
String str, boolean append)`

INIT METHOD

- Actions for initializing bean's state after creation
 - For example, add check for file write access and create file
- Initialize beans in constructor is not always possible
 - Bean can get its dependencies after creation

- FileEventLogger:

```
public void init() throws IOException {  
    this.file = new File(fileName);  
    // check file write access  
}
```

- FileEventLogger:

```
public void init() throws IOException {  
    this.file = new File(fileName);  
    // check file write access  
}
```

- spring.xml

```
<bean id="..." class="..."  
        init-method="init">
```

- no arguments
- any access modifier
- can return something or throw Exception

CacheFileEventLogger

```
- cacheSize: int;  
- cache: List<Event>;  
+ logEvent(...); // cache or write
```



FileEventLogger

CacheFileEventLogger

- cacheSize: int;
- cache: List<Event>;
- + logEvent(...); *// cache or write*

```
public void logEvent(Event event) {  
    cache.add(event);  
  
    if (cache.size() == cacheSize) {  
        writeEventsFromCache();  
        cache.clear();  
    }  
}
```


- Being called when context is closing

- CacheFileEventListener:

```
public void destroy() {  
    if ( !cache.isEmpty() )  
        writeEventsFromCache();  
}
```

- spring.xml

```
<bean id="..." class="..."  
    destroy-method="destroy">
```

- App class

```
ConfigurableApplicationContext ctx = ...  
ctx.close();
```



THANK YOU

YURIY_TKACH@EPAM.COM

APRIL, 2015