Skillbox

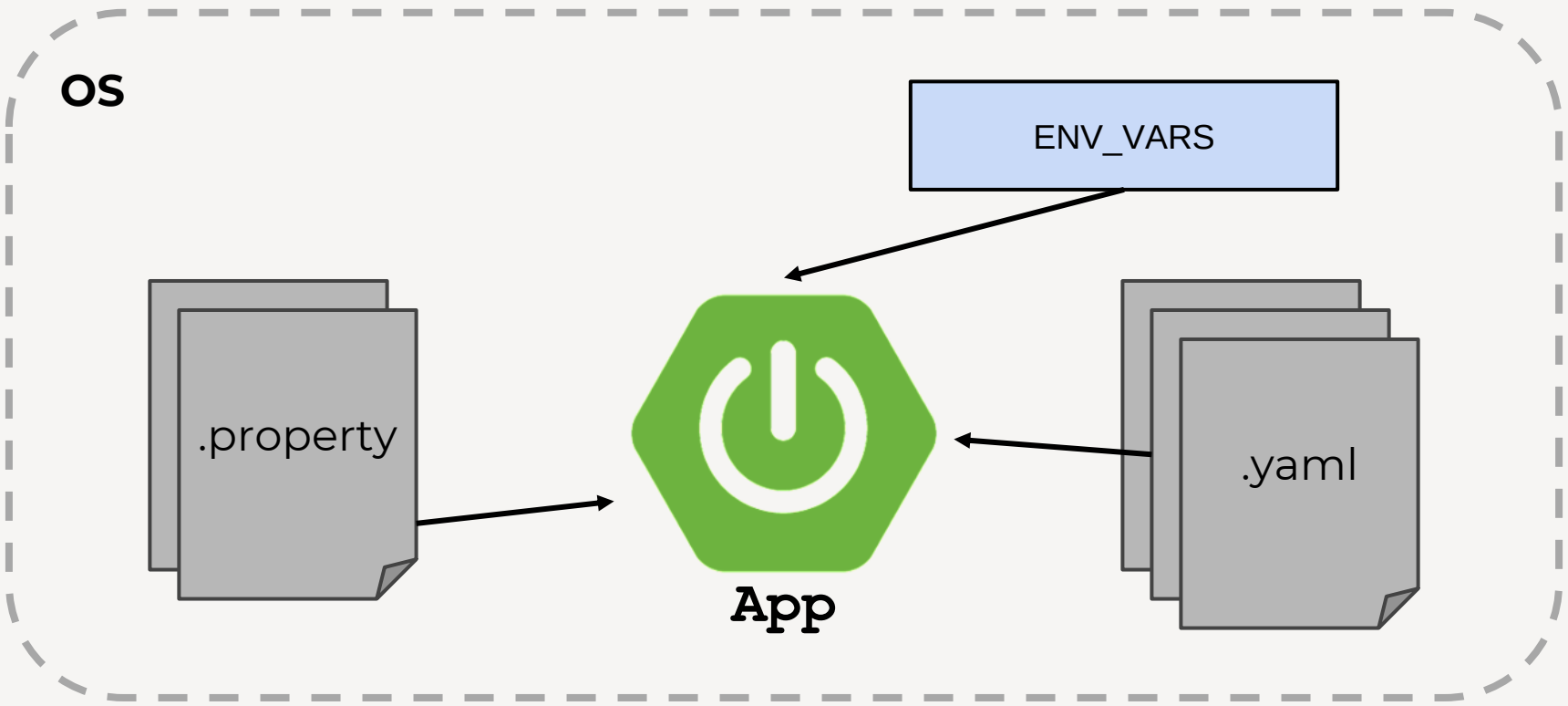
# Application properties Профили Формат YAML

**Кирилл Кошаев**

Java-разработчик в «Газпром информ»

# Окружение

Окружение — это среда, в которой выполняется программа или приложение. Окружение представляет собой аппаратную и программную инфраструктуру, которая поддерживает работу конкретной кодовой базы приложения в реальном времени



Skillbox

# Application Properties

Свойства — это обычные пары «ключ-значение», использующиеся в Java-приложениях в качестве конфигурационных параметров

**Источники Application Properties:**

- файлы свойств,
- системные свойств JV,
- переменные системной среды операционной системы,
- параметры контекста сервлета,
- специальные объекты свойств,
- структуры «ключ-значение».

# Old times — early Spring

Изначально в Spring можно было указывать свойства для бинов в составе xml конфигураций контекста

```
<bean
class="org.springframework.context.support.PropertySourcesPlace
holderConfigurer">
 <property name="location">
 <list>
 <value>classpath:some.properties</value>
 </list>
 </property>
 <property name="ignoreUnresolvablePlaceholders" value="true"/>
</bean>
```

```
@Value("${foo.someProperty}"
private String someProperty;
```

# Spring 3.1

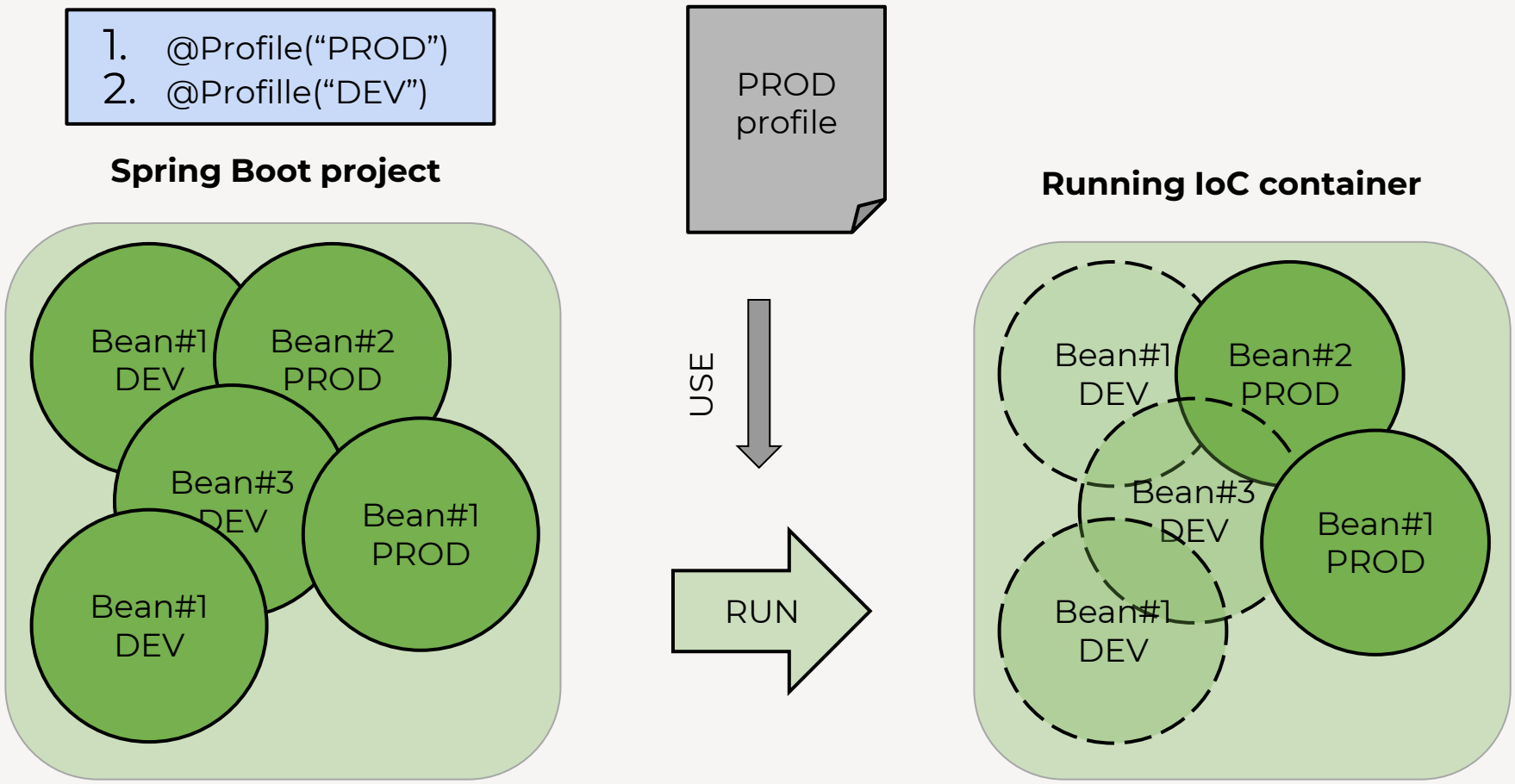
Была представлена аннотация @PropertySource и появилась возможность указывать на property файл прямо в java-классе

```
@Configuration
@PropertySource("/foo.properties")
public class SuperFooConfiguration {

    @Value("${foo.size}")
    private String size;

    @Value("${foo.isBool}")
    private String isBool;

    @Value("${foo.qwertyProp}")
    private String qwertyProp;

}
```

Skillbox

# Профили

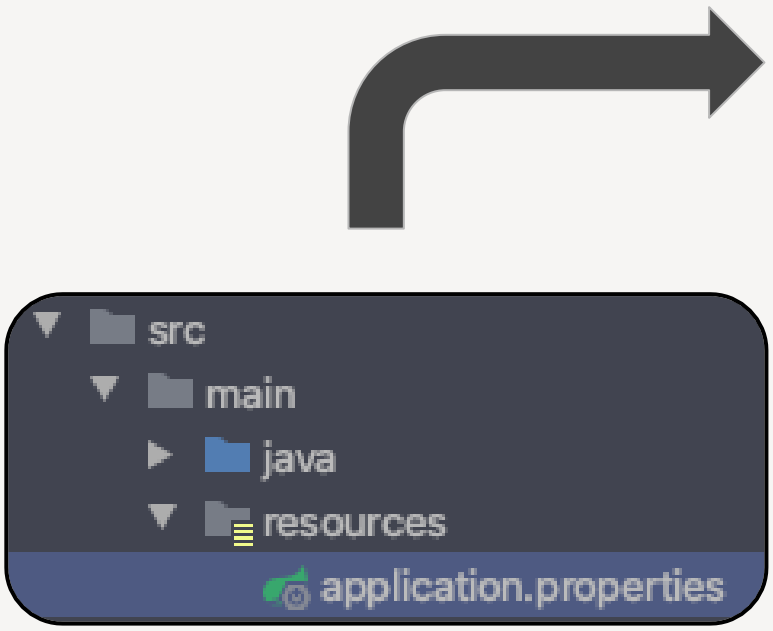
Профили — это ключевая особенность Spring framework, позволяющая создавать конфигурации запуска приложения, сопоставляя бины приложения с разными профилями, например, dev, test или prod



Skillbox

# Spring Boot Application Properties

Достаточно создать в resources-каталоге проекта файл application.properties и начать сразу пользоваться содержащимися там свойствами, не тратя время на конфигурирование.
Более того, application.properties генерируется по умолчанию



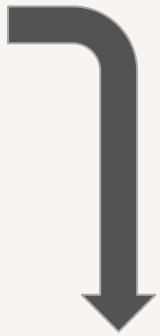

```
@Configuration
public class SuperFooConfiguration {
    @Value("${spring.prop}")
    private String size;

     @Value("${server.isActive}")
    private String isBool;

     @Value("${context.qwertyProp}")
    private String qwertyProp;
}
```

Skillbox

# @ConfigurationProperties

Документация SpringBoot рекомендует удобный механизм, позволяющий определять свои собственные свойства при помощи простых POJO-классов

```
@Configuration //optional
@ConfigurationProperties(prefix = "my")
public class MyConfigProperties {

    private String hostName;
    private int port;
    private String from;

    // standard getters and setters
}
```



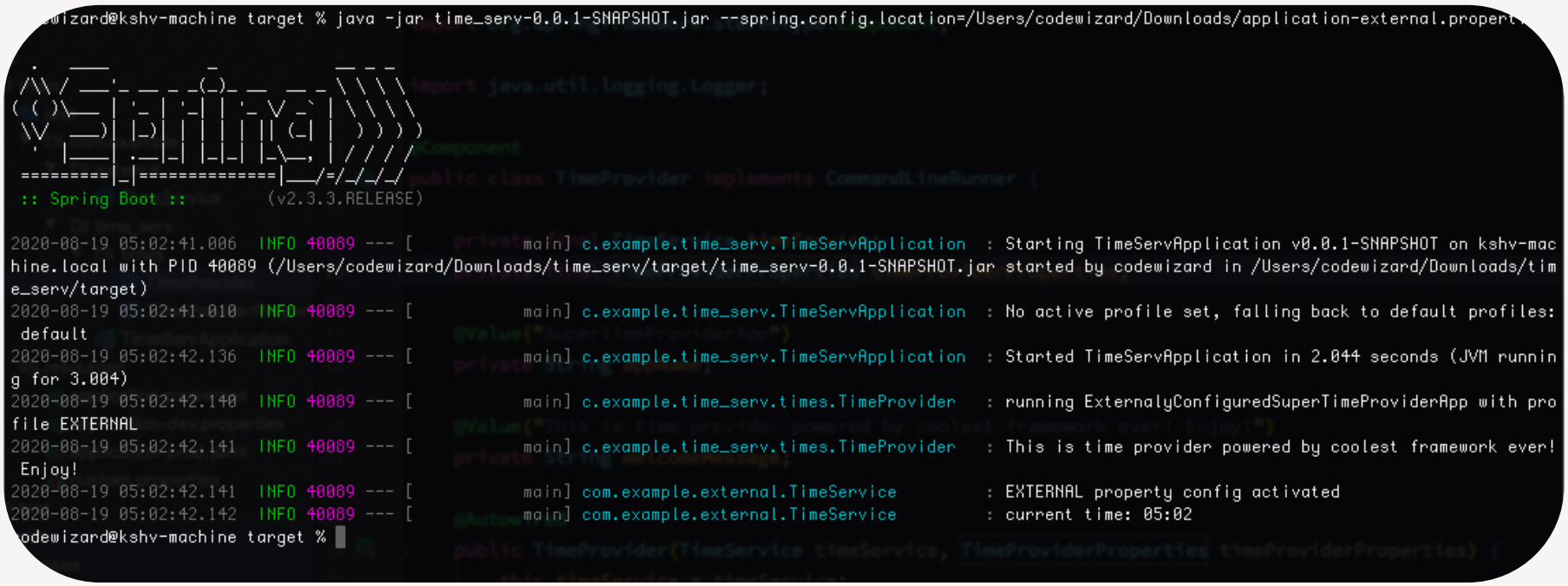
```
@SpringBootApplication
@EnableConfigurationProperties(MyConfigProperties.class)
public class DemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

Skillbox

# Running JAR

При запуске скомпилированного jar-файла проекта мы можем указывать в качестве параметра другой конфигурационный файл свойств, с которым мы хотим провести запуск приложения

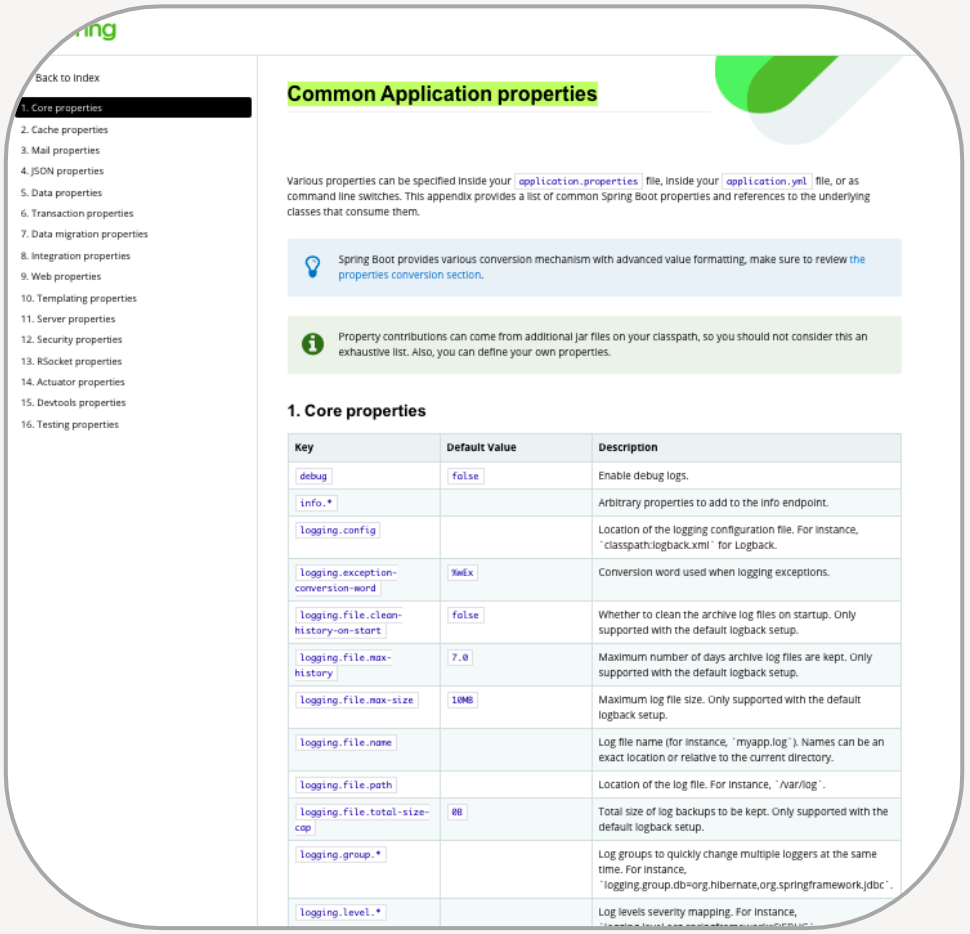
```
java -jar app.jar --spring.config.location=classpath:/another-location.properties
java -jar app.jar --spring.config.location=config/*/
```



Skillbox

# Common Application properties

https://docs.spring.io/spring-boot/docs/current/reference/html/appendix-application-properties.html



ing

Back to Index

1. Core properties
2. Cache properties
3. Mail properties
4. JSON properties
5. Data properties
6. Transaction properties
7. Data migration properties
8. Integration properties
9. Web properties
10. Templating properties
11. Server properties
12. Security properties
13. RSocket properties
14. Actuator properties
15. Devtools properties
16. Testing properties

## Common Application properties

Various properties can be specified inside your `application.properties` file, inside your `application.yml` file, or as command line switches. This appendix provides a list of common Spring Boot properties and references to the underlying classes that consume them.

Spring Boot provides various conversion mechanism with advanced value formatting, make sure to review the properties conversion section.

Property contributions can come from additional jar files on your classpath, so you should not consider this an exhaustive list. Also, you can define your own properties.

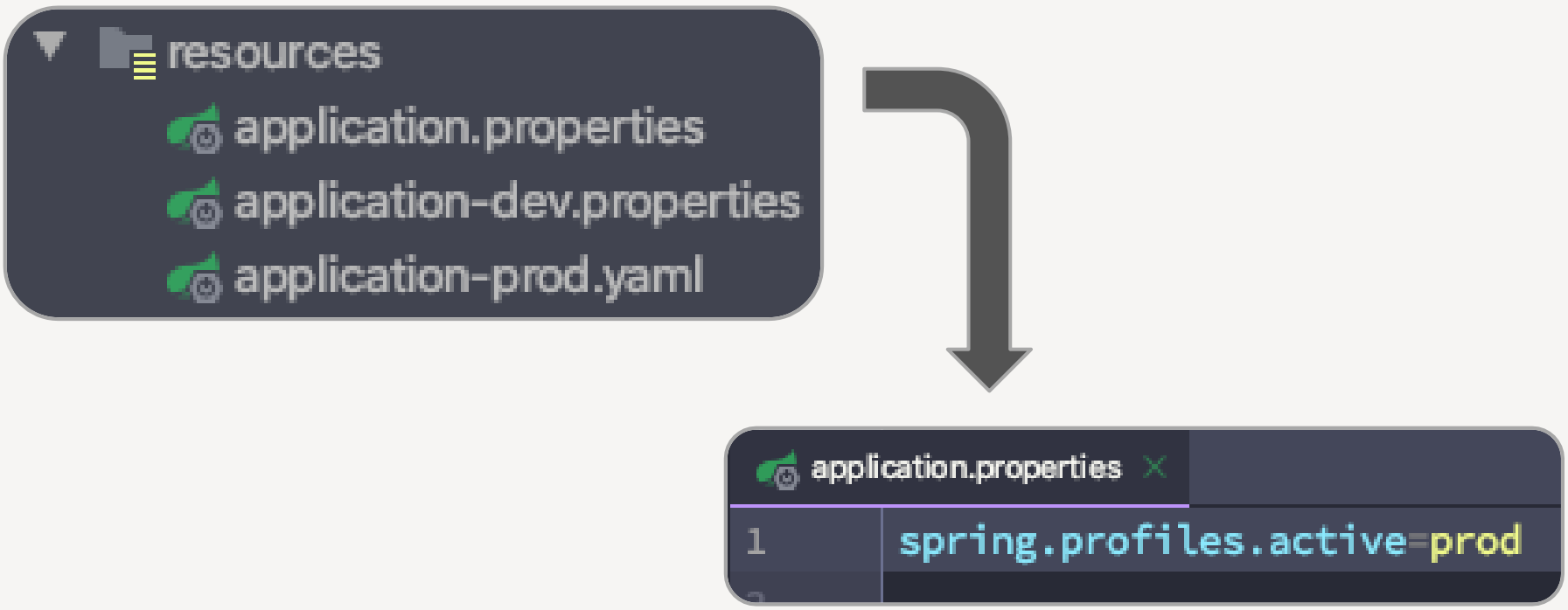
### 1. Core properties

| Key | Default Value | Description |
|---|---|---|
| `debug` | `false` | Enable debug logs. |
| `info.*` | | Arbitrary properties to add to the info endpoint. |
| `logging.config` | | Location of the logging configuration file. For instance, `classpath:logback.xml` for Logback. |
| `logging.exception-conversion-word` | `%wEx` | Conversion word used when logging exceptions. |
| `logging.file.clean-history-on-start` | `false` | Whether to clean the archive log files on startup. Only supported with the default logback setup. |
| `logging.file.max-history` | `7.0` | Maximum number of days archive log files are kept. Only supported with the default logback setup. |
| `logging.file.max-size` | `10MB` | Maximum log file size. Only supported with the default logback setup. |
| `logging.file.name` | | Log file name (for instance, `myapp.log`). Names can be an exact location or relative to the current directory. |
| `logging.file.path` | | Location of the log file. For instance, `/var/log`. |
| `logging.file.total-size-cap` | `0B` | Total size of log backups to be kept. Only supported with the default logback setup. |
| `logging.group.*` | | Log groups to quickly change multiple loggers at the same time. For instance, `logging.group.db=org.hibernate,org.springframework.jdbc`. |
| `logging.level.*` | | Log levels severity mapping. For instance, |

Skillbox

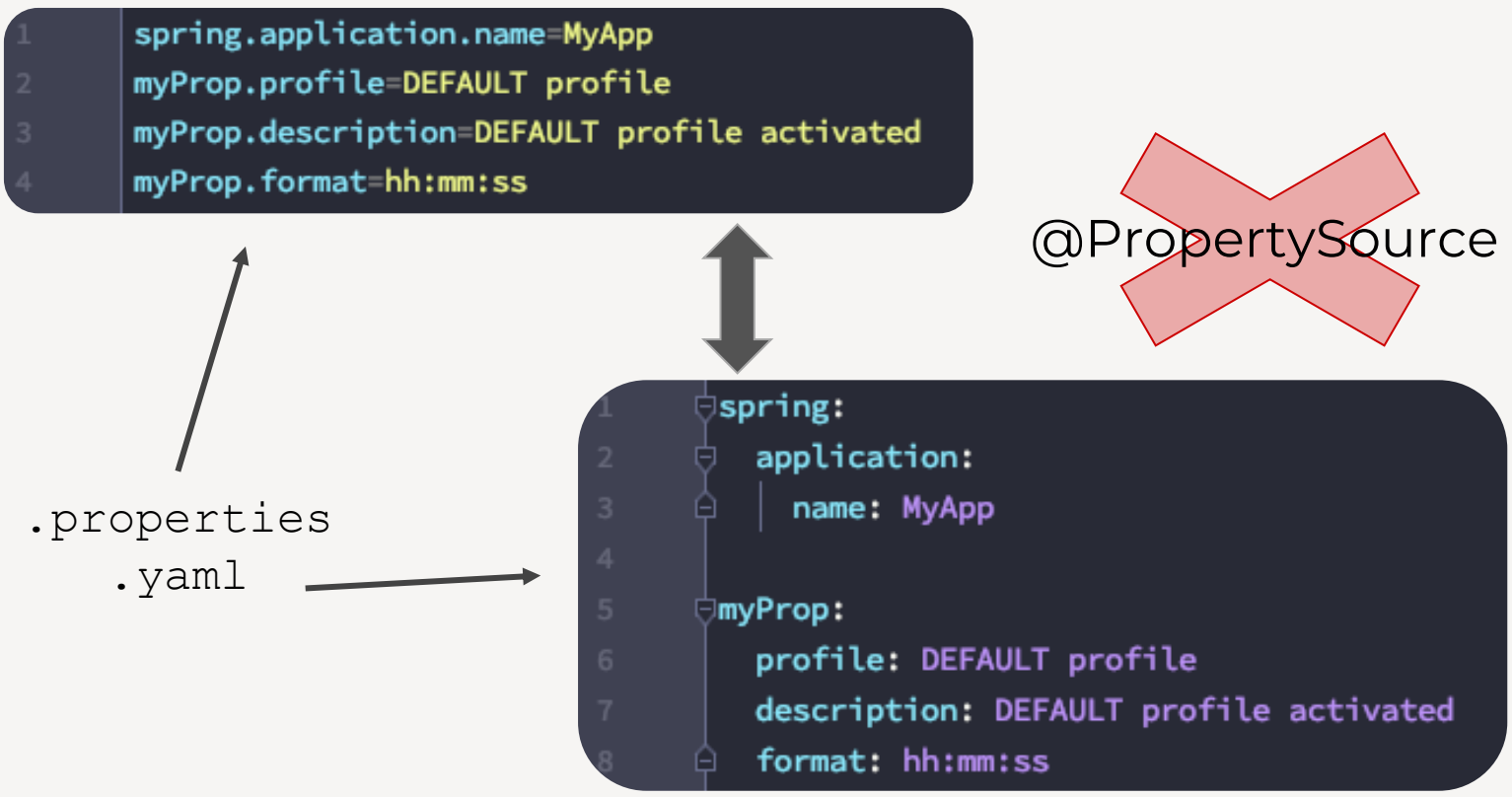
# Spring Boot
# работа с профилями

Можно определить файл «application-[имя окружения].properties» в каталоге resources проекта, а затем при запуске установить необходимый профиль, обратившись к нужному по имени окружения из файла application.properties



# YAML

YAML — «дружественный» формат сериализации данных, концептуально близкий к языкам разметки, но ориентированный на удобство ввода-вывода типичных структур данных многих языков программирования



Skillbox

# Спасибо за внимание!